N° d'ordre : 3447

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Petra KRÄMER**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Construction de mosaïques de super-résolution à partir de la vidéo de basse résolution. Application au résumé vidéo et la dissimulation d'erreurs de transmission.

Soutenue le : 15 octobre 2007

Après avis des rapporteurs :

M. Philippe JOLY	MdC HDR, Université Toulouse 3
M. A. Murat TEKALP	Professeur, Koc University, Turquie

Devant la commission d'examen composée de :

Mme Jenny BENOIS-PINEAU	Professeur, Université Bordeaux 1	Directrice de thèse
M. Luc BRUN	Professeur, ENSICAEN	Examinateur
M. Jean-Philippe DOMENGER	Professeur, Université Bordeaux 1	Codirecteur de thèse
M. Ofer HADAR	Senior Lecturer, Ben Gurion University, Israël	Examinateur
M. Philippe JOLY	MdC HDR, Université Toulouse 3	Rapporteur
M. Henri NICOLAS	Professeur, Université Bordeaux 1	Président

– 2007 –

Acknowledgements

First of all, I would like to gratefully acknowledge the supervision of Jenny Benois-Pineau and Jean-Philippe Domenger during this work. I also thank Jenny Benois-Pineau for welcoming me kindly in the Video Analysis and Indexing research group. I am honoured that Murat Tekalp and Philippe Joly agreed to review this PhD thesis. I want to thank Henri Nicolas who agreed to be the president of the dissertation defense committee and for interesting discussions on illumination correction. I am grateful to all members of the dissertation defense committee for attending the defense and judging my work. Furthermore, I thank Luc Brun who agreed to participate in the dissertation defense committee and Ofer Hadar for inviting me for a research visit at Ben Gurion University in Israel. I also wish to thank the students I supervised during this PhD as they also contributed to this PhD thesis. Thanks to Sebastian for translating a part of my source code from Matlab to C++, to Marta for the motion reestimation part and to Guido for the error concealment part. I want to thank Chrit, Pascal and Baudouin for providing MRI image sequences and for interesting discussions on biomedical image processing. I thank the current and former members of the Video Analysis and Indexing research group. Especially thanks to Lionel, Laurent, Nicolas and Francesca for sharing the office during the first two years of my PhD and to Claire, Remy and Chris for supporting me during the writing of this PhD thesis. Finally, I am very grateful to my family and friends for their understanding and encouragement throughout this PhD.

Construction de mosaïques de super-résolution à partir de la vidéo de basse résolution. Application au résumé vidéo et la dissimulation d'erreurs de transmission.

Résumé : La numérisation des vidéos existantes ainsi que le développement explosif des services multimédia par des réseaux comme la diffusion de la télévision numérique ou les communications mobiles ont produit une énorme quantité de vidéos compressées. Ceci nécessite des outils d'indexation et de navigation efficaces, mais une indexation avant l'encodage n'est pas habituelle. L'approche courante est le décodage complet des ces vidéos pour ensuite créer des indexes. Ceci est très coûteux et par conséquent non réalisable en temps réel. De plus, des informations importantes comme le mouvement, perdus lors du décodage, sont reestimées bien que déjà présentes dans le flux comprimé. Notre but dans cette thèse est donc la réutilisation des données déjà présents dans le flux comprimé MPEG pour l'indexation et la navigation rapide. Plus précisément, nous extrayons des coefficients DC et des vecteurs de mouvement.

Dans le cadre de cette thèse, nous nous sommes en particulier intéressés à la construction de mosaïques à partir des images DC extraites des images I. Une mosaïque est construite par recalage et fusion de toutes les images d'une séquence vidéo dans un seul système de coordonnées. Ce dernier est en général aligné avec une des images de la séquence : l'image de référence. Il en résulte une seule image qui donne une vue globale de la séquence. Ainsi, nous proposons dans cette thèse un système complet pour la construction des mosaïques à partir du flux MPEG-1/2 qui tient compte de différentes problèmes apparaissant dans des séquences vidéo réeles, comme par exemple des objets en mouvment ou des changements d'éclairage.

Une tâche essentielle pour la construction d'une mosaïque est l'estimation de mouvement entre chaque image de la séquence et l'image de référence. Notre méthode se base sur une estimation robuste du mouvement global de la caméra à partir des vecteurs de mouvement des images P. Cependant, le mouvement global de la caméra estimé pour une image P peut être incorrect car il dépend fortement de la précision des vecteurs encodés. Nous détectons les images P concernées en tenant compte des coefficients DC de l'erreur encodée associée et proposons deux méthodes pour corriger ces mouvements.

Une mosaïque construite à partir des images DC a une résolution très faible et souffre des effets d'aliasing dus à la nature des images DC. Afin d'augmenter sa résolution et d'améliorer sa qualité visuelle, nous appliquons une méthode de super-résolution basée sur des rétro-projections itératives. Les méthodes de super-résolution sont également basées sur le recalage et la fusion des images d'une séquence vidéo, mais sont accompagnées d'une restauration d'image. Dans ce cadre, nous avons développé une nouvelle méthode d'estimation de flou dû au mouvement de la caméra ainsi qu'une méthode correspondante de restauration spectrale.

La restauration spectrale permet de traiter le flou globalement, mais, dans le cas des ob-

jets ayant un mouvement indépendant du mouvement de la caméra, des flous locaux apparaissent. C'est pourquoi, nous proposons un nouvel algorithme de super-résolution dérivé de la restauration spatiale itérative de Van Cittert et Jansson permettant de restaurer des flous locaux. En nous basant sur une segmentation d'objets en mouvement, nous restaurons séparément la mosaïque d'arrière-plan et les objets de l'avant-plan. Nous avons adapté notre méthode d'estimation de flou en conséquence.

Dans une premier temps, nous avons appliqué notre méthode à la construction de résumé vidéo avec pour l'objectif la navigation rapide par mosaïques dans la vidéo compressée. Puis, nous établissions comment la réutilisation des résultats intermédiaires sert à d'autres tâches d'indexation, notamment à la détection de changement de plan pour les images I et à la caractérisation du mouvement de la caméra. Enfin, nous avons exploré le domaine de la récupération des erreurs de transmission. Notre approche consiste en construire une mosaïque lors du décodage d'un plan ; en cas de perte de données, l'information manquante peut être dissimulée grace à cette mosaïque.

Mots clés : Indexation, MPEG, mosaïques, super-résolution

Discipline : Informatique

LaBRI (UMR CNRS 5800)
Université Bordeaux 1
351, cours de la Libération
33405 Talence Cedex (FRANCE)

Résumé Etendu

Introduction, contexte et motivation

Aujourd'hui une énorme quantité de vidéos compressées est disponible, cette croissance importante s'expliquant par la numérisation des vidéos existantes et le développement exponentiel des services de réseaux multimédia comme la diffusion de la vidéo numérique et les communications via des téléphones portables. Pour pouvoir exploiter ces données, il s'avère nécessaire de développer des outils efficaces pour indexer leur contenu et aussi pour permettre de naviguer à travers ces données pour en extraire une information. L'approche habituelle consiste à décoder complètement le flux pour pouvoir ensuite l'indexer. Cette approche nécessite des traitements lourds qui ne peuvent pas être réalisés en temps réel. De fait, une grande quantité des vidéos compressées ne peut pas être indexée. Nous proposons dans cette thèse d'exploiter seulement les données qui proviennent d'un flux vidéo partiellement décodé.

Jusqu'à maintenant le standard MPEG-2 [117] est le plus utilisé pour l'encodage des vidéos, qu'elles soient stockées sur support numérique ou bien diffusées sur des canaux de transmission. MPEG-2 est un système d'encodage hybride basé sur la compensation de mouvement et le codage par transformée en cosinus discret (DCT). Une information essentielle pour l'indexation est l'information de mouvement. Cette information présente dans le flux MPEG-1/2 disparaît lors d'un décodage complet.

Des méthodes pertinentes ont déjà été proposées dans le domaine compressé pour différentes tâches d'indexation. Le point commun de ces méthodes est qu'elles sont principalement basées sur l'extraction des *images DC* et/ou des *vecteurs de mouvement* du flux MPEG-1/2. Une image DC est huit fois plus petite que l'image originale, en effet un pixel dans l'image DC correspond à la valeur moyenne d'un bloc 8×8 dans l'image originale. Les images DC et les vecteurs de mouvement peuvent facilement être extraits en décodant partiellement le flux MPEG-1/2.

Yeo et Liu [208] démontrent qu'en résolution DC certaines caractéristiques importantes pour le traitement d'images et l'indexation sont bien préservées. Un exemple de ces résultats est proposé par Bescós [12] qui a développé une méthode efficace pour la détection de changement de plan dans le domaine compressé en utilisant des images DC. Souvent, les vecteurs de mouvement MPEG sont considérés comme trop erronés pour être réellement

exploitables dans une estimation de mouvement. Pourtant, certaines méthodes arrivent à compenser les vecteurs erronés et permettent d'obtenir des résultats suffisamment robustes pour justifier l'utilisation des vecteurs de mouvement. Par exemple, Durik et Benois-Pineau [38] ont développé une méthode robuste pour l'estimation de mouvement à partir des vecteurs MPEG dans le cadre de détection de changement de plan. Dans [33], Coimbra et Davies présentent une méthode de détection de piétons qui utilise les vecteurs de mouvement de MPEG-2.

Dans cette thèse, nous nous sommes particulièrement intéressés à la construction des mosaïques pour le résumé des vidéos compressées en MPEG-1/2. Une mosaïque est construite par recalage et fusion de toutes les images d'un plan vidéo dans un seul repère. Il en résulte une seule image qui donne une vue globale du plan. L'intérêt d'une image résumant le contenu d'un plan vidéo est de permettre une compréhension immédiate de la scène. C'est principalement pour cette raison que, pendant la dernière décennie, des méthodes sophistiquées ont été proposées pour la construction de mosaïques. Les représentations par mosaïques sont intégrées dans le standard de codage MPEG-4 [118] car elles permettent l'encodage efficace d'un plan en le décomposant en deux parties : la mosaïque en arrière-plan et les objets en avant-plan. En outre, elles sont définies dans le standard MPEG-7 [119] en tant que descripteur de plan et peuvent être utilisées pour la navigation et l'indexation des vidéos.

Depuis les travaux de Peleg et Herman [139] et Sawhney et Ayer [156], les méthodes de construction de mosaïque semblent totalement achevées. Pourtant la plupart des méthodes proposées dans la littérature sont définies pour la vidéo non compressée, parmi elles la méthode de Irani et Anandan [72] est intéressante pour la navigation et l'indexation des vidéos. Contrairement à ces méthodes, nous nous focalisons dans cette thèse sur la construction de mosaïque à partir du flux compressé MPEG-1/2 sans un décodage complet. Nous prétendons que les mosaïques construites à partir de la vidéo initiale ou décodée sont souvent trop grandes pour leur affichage sur l'écran sans défilement ou sous-échantillonnage. En conséquence, nous proposons de construire la mosaïque à une résolution plus basse fournissant une vue globale de la scène qui est plus appropriée pour la navigation dans un document vidéo. Pour cela, nous considérons comme séquence d'entrée les images DC des images I. De plus, nous utilisons les vecteurs de mouvement des images P dans une estimation du mouvement global de la caméra pour l'alignement des images DC des images I.

L'intérêt de notre approche réside dans la proposition d'un système complet pour la construction de mosaïques à partir de la vidéo qui se base sur les images DC des images I et les vecteurs de mouvement des images P. Nous tenons compte des différents aspects et problèmes qui peuvent apparaître pendant la construction de mosaïque, à savoir les vecteurs de mouvement bruités et/ou erronés, la suppression des objets en mouvement et leur segmentation approximative, ou le changement d'illumination entre les images d'entrée. Afin d'obtenir une description complète du plan, les objets représentatifs sont ensuite insérés dans la mosaïque. Nous proposons un algorithme rapide et efficace qui couvre l'ensemble des différentes nécessaires à la construction de mosaïque.

Les images DC sont caractérisées par une très faible résolution et des artefacts d'aliasing.

De plus, deux phénomènes peuvent engendrer du flou. Le premier est dû au mouvement de la caméra pendant le temps d'acquisition et le deuxième résulte du moyennage par blocs. Quand une mosaïque est construite à partir des images DC, sa résolution peut être trop faible pour une vue globale de la scène et la qualité de l'image peut être insuffisante à cause des dégradations des images DC. C'est pourquoi nous nous sommes intéressés dans la suite de cette thèse à la super-résolution.

Il existe plusieurs méthodes pour l'augmentation de résolution des images vidéo. La manière la plus facile pour créer une image avec une résolution plus forte est une simple interpolation, par exemple une interpolation bilinéaire ou bicubique. Cependant, les résultats sont très médiocres à cause de l'aliasing et de la perte des hautes fréquences. Fassino et Montanvert [49] ont proposé une méthode pour augmenter la résolution de la vidéo compressée en MPEG. La résolution des images I est augmentée par des projections de l'image I décodée sur des ensembles convexes. Puis, la résolution des images P et B est augmentée par une mise à l'échelle des vecteurs de mouvement et des macroblocs utilisés dans le processus de compensation de mouvement. Cependant, dans le cas de la vidéo, il est possible d'exploiter l'information des images avoisinantes afin d'obtenir une image de résolution plus haute par des algorithmes de *super-résolution*. Supposant que chaque image de la séquence fournit une vue légèrement différente de la scène, toutes ces images peuvent alors être combinées dans une seule image de résolution plus haute. Cette technique permet de restaurer certaines hautes fréquences et d'obtenir ainsi une image de qualité supérieure.

Plusieurs algorithmes de super-résolution ont été développés pour la vidéo non compressée par exemple par Patti et al. [137] et Irani et Peleg [75], et pour la vidéo compressée par Segall et al. [164]. L'idée d'appliquer un algorithme de super-résolution aux mosaïques construites a été proposé par Capel et Zisserman [22] et Zomet et Peleg [215]. Ces travaux se faisaient dans le cadre de la vidéo non compressée, dans cette thèse nous nous inspirons de leur approche mais dans le cadre de la vidéo compressée. Les points originaux de notre approche sont : l'estimation de mouvement à partir de la vidéo compressée, le choix du modèle de flou adéquat et du filtre de restauration, ainsi que l'estimation des paramètres de flou.

Dans cette thèse, nous présentons deux algorithmes de super-résolution. Le premier est basé sur les rétroprojections itératives de Irani et Peleg [75] incluant une *restauration dans le domaine fréquentiel*. La restauration se faisant dans le domaine fréquentiel, cette méthode ne permet que la restauration des flous globaux. Malheureusement, on observe dans la vidéo des situations plus complexes où un flou local apparaît dû par exemple au mouvement des objets. Dans ce cas, le mouvement des objets peut être très complexe mais pour les images DC les objets sont représentés par peu de pixels. En conséquence, il est très difficile d'estimer le mouvement de ces objets et donc de les positionner précisément dans le repère de la super-résolution. Dans [75], un algorithme de super-résolution est appliqué aux objets en faisant l'hypothèse que le mouvement des objets est un mouvement 2D paramétrique, dans notre travail nous proposons une méthode générique qui ne tient pas compte de cette hypothèse. Ainsi, nous proposons de restaurer les objets en mouvement et d'augmenter leur résolution par une méthode de restauration n'utilisant qu'une seule image tandis que nous appliquons une méthode de super-résolution à l'arrière-plan.

L'origine de flou est double. D'une part, le flou provient des images de pleine résolution dû au mouvement pendant le temps d'acquisition. D'autre part, les images DC représentent une version passe-bas des images de pleine résolution, le filtre passe-bas étant effectué localement dans un bloc de 8×8 pixels. Ainsi, le flou global est un résultat de ces deux phénomènes. Pour cette raison, nous proposons plusieurs modèles de flou pour modéliser la fonction de flou global réel. De plus, nous proposons une méthode efficace d'*estimation de paramètres de flou global et local dans la direction de mouvement*. Nous évaluons pour les méthodes de super-résolution la performance des modèles de flou en combinaison avec différents filtres de restauration.

En plus du résumé vidéo, nous abordons dans cette thèse d'autres domaines d'application de notre méthode de construction de mosaïques. D'abord, nous montrons comment des résultats intermédiaires peuvent être utilisés pour d'autres tâches d'indexation, précisément la détection de changement de plan pour les images I et la caractérisation du mouvement de caméra. Une application supplémentaire que nous avons élaborée dans cette thèse est l'utilisation des mosaïques pour la dissimulation d'erreurs (« error concealement » en anglais). Notre approche consiste à construire une mosaïque en fonction des images du plan courant, en cas de perte de données, l'information manquante sera restaurée en fonction des informations présentes dans la mosaïque.

Résumé du contenu de la thèse

Cette thèse est structurée en deux parties. La Partie 1 présente la construction des mosaïques et leur augmentation de résolution par la super-résolution. La Partie 2 présente l'application des résultats de la première partie à différents problèmes : la détection de changements de plan, la caractérisation du mouvement de la caméra et la dissimulation d'erreurs de transmission. Le premier et le dernier des neuf chapitres sont respectivement une introduction et conclusion qui situent les enjeux et la démarche adoptée en général.

Plus précisément la Partie 1 est décomposée en 4 chapitres (numéroté de 2 à 5). Le Chapitre 2 présente les standards de codage vidéo MPEG-1 and MPEG-2 et décrit l'extraction des informations nécessaires à la construction de mosaïques. Nous utilisons les images DC des images I en tant que séquence d'entrée pour la construction de la mosaïque, les vecteurs de mouvement des images P pour l'alignement de la séquence, ainsi que les images DC de l'erreur de compensation de mouvement des images P pour mesurer la confiance dans l'exactitude des vecteurs de mouvement.

Le Chapitre 3, après un état de l'art sur la construction de mosaïque, décrit les techniques développées pour la construction de mosaïque à partir du flux compressé MPEG-1/2. Dans cette thèse, nous avons proposé un *système complet pour la construction de mosaïque à partir de la vidéo compressée*. Toutes les étapes de ce processus ont été étudiées. Nous avons apporté une solution aux différents problèmes qui peuvent être rencontrés : inexactitude des vecteurs de mouvement dans le flux compressé, changements d'illumination, présence des objets en mouvement et la difficulté de leur segmentation. Les résultats obtenus peuvent être considérés comme probants. Par ailleurs, les temps de calcul sont intéressants, les performances

de cette méthode prennent environ trois fois plus de temps que le temps du décodage.

Nous avons proposée une méthode de calcul de la transformation géométrique permettant de projeter une image I dans le repère de la mosaïque. Cette méthode est robuste malgré le bruit et l'absence ou l'inexactitude des vecteurs de mouvement des images P. Le mouvement global de la caméra est estimé pour les images P par un estimateur robuste. Néanmoins, ceci n'est pas suffisant pour obtenir la trajectoire complète du mouvement dans la séquence des images DC. Pour ce faire, nous avons proposé pour les images I d'extrapoler les paramètres de mouvement des images P précédentes. De plus, les vecteurs de mouvement encodés peuvent être erronés. Nous détectons, alors les images P concernées en nous basant sur l'erreur de compensation de mouvement. Dans ce chapitre, nous avons proposé une méthode pour corriger le mouvement erroné et une autre méthode pour le réestimer. D'après nos connaissances une telle *étude exhaustive du mouvement à partir du flux compressé* avec son raffinement n'a pas encore été proposée dans la littérature.

Un autre aspect de notre méthode de construction de mosaïque est la suppression des objets en mouvement. Nous avons présenté une méthode pour la détection des objets en mouvement dans la séquence des images DC des images I. Ceci permet la construction d'une mosaïque sans artefact, mais aussi l'insertion ultérieure des objets représentatifs dans la mosaïque. Ainsi, nous obtenons une description complète de la séquence. Durant cette phase, nous compensons également les défauts de segmentation localisés sur le bord des objets.

Les changements d'illumination entre les images de la séquence engendrent des artefacts dans la mosaïque. Pour les éviter, nous avons développé une méthode qui corrige les changements d'illumination. L'exclusion des objets en mouvement, des textures, des contours ainsi que le rejet des pixels aberrants dans le calcul permettent d'obtenir des résultats de bonne qualité visuelle. Dans le cas d'images bruitées, certaines méthodes de correction se trouvent dégradées par la propagation de l'erreur d'estimation. Afin de réduire ce phénomène, nous avons proposé de corriger la séquence de manière hiérarchique.

Les mosaïques obtenues ont une résolution très basse et elles peuvent être dégradées par le flou et l'aliasing du à la nature des images DC. De ce fait, un utilisateur préférera sûrement une résolution plus haute et une meilleure qualité visuelle pour la visualisation de la scène. Pour répondre à cette attente, nous avons proposée dans cette thèse *deux méthodes de super-résolution*.

Le Chapitre 4 présente une première méthode de super-résolution basée sur l'itération des rétroprojections couplée avec une restauration dans le domaine fréquentiel. Pour cette restauration, nous avons considéré plusieurs modèles de flou : flou isotrope Gaussien, flou anisotrope Gaussien et flou dû à un mouvement linéaire. En outre, nous avons développé une *méthode efficace pour l'estimation des paramètres du flou dans la direction du mouvement*.

Une fois le modèle de flou établi, nous avons comparé deux méthodes classiques de restauration : par filtrage pseudo-inverse et par filtre de Wiener. En principe, le meilleur résultat est obtenu en considérant le modèle de flou dû au mouvement linéaire en combinaison avec le filtre pseudo-inverse.

Dû aux forts effets d'aliasing dans la séquence d'images DC il est parfois impossible de superposer exactement les images dans les régions texturées et les contours. Il en résulte

alors des artefacts dans l'image de super-résolution qui s'amplifient à chaque itération de l'algorithme de super-résolution. Pour cette raison, nous avons intégré un *opérateur de régularisation* dans l'algorithme de super-résolution qui pénalise les contours et les textures dans le processus de rétroprojection.

Néanmoins, cette méthode a des inconvénients. La restauration dans le domaine fréquentiel limite la restauration seulement à des flous globaux alors qu'en général dans les vidéos des situations plus complexes peuvent être rencontrées. C'est notamment le cas des objets en mouvement qui produisent des flous locaux. Par conséquent, cette méthode ne permet pas le traitement des objets en mouvement. En plus, nous obtenons des temps de calcul important du au « padding » et aux transformations de Fourier successives.

Nous présentons une deuxième méthode de super-résolution dans le Chapitre 5 avec pour l'objectif de proposer une solution aux défauts et surtout aux manques de la première méthode. Ainsi, la restauration est réalisée dans le domaine spatial ce qui permet de *restaurer des flous locaux*. Pour réaliser la restauration des flous locaux, nous avons modifié la méthode d'estimation des paramètres du flou présentée au chapitre Chapitre 4.

Le mouvement des objets peut être très complexe et dans le cas des images DC les objets sont normalement représentés par peu de pixels. Pour cette raison, nous avons décomposé la méthode en deux temps. Dans un premier temps, un algorithme de super-résolution est utilisé avec les informations de l'arrière-plan, dans un second temps à partir d'une image de référence les objets sont augmentés en résolution puis finalement insérés dans la mosaïque en super-résolution. Comme pour la méthode précédente, nous avons également testé les différents modèles de flou dans ce schéma de restauration. Nous avons obtenu le meilleur résultat pour le flou Gaussien anisotrope, mais les résultats fournis par le flou Gaussien isotrope étaient proches. Il semblerait que dans cette méthode, la modélisation de la PSF par un modèle de flou dû au mouvement linéaire ne semble pas appropriée, du fait de la discrétisation du masque de convolution.

Nous avons comparé les deux méthodes de super-résolution en utilisant des séquences vidéo sans objets en mouvement. La méthode de super-résolution basée sur la restauration fréquentielle donne des résultats légèrement meilleurs que la méthode de super-résolution basée sur une restauration spatiale. Ces résultats sont obtenus en faisant une mesure quantitative de l'erreur. Par contre, visuellement les artefacts sont moindres dans les résultats produits par la méthode de super-résolution basée sur la restauration spatiale. De plus, nous obtenons un gain important en temps de calcul car la convolution est accomplie à basse résolution et les transformations de Fourier sont omises.

Dans cette thèse, nous nous intéressons également aux différentes applications de notre méthode de super-résolution. Les domaines d'applications concernent plus précisément la détection de changement de plan, la caractérisation du mouvement de la caméra et la dissimulation des erreurs de transmission. La Partie 2 débute par le Chapitre 6 où une brève introduction des domaines d'applications est présentée.

Le Chapitre 7 propose une méthode pour la détection des changements de plan pour les images I ainsi qu'une méthode de caractérisation du mouvement dominant de la caméra. La réalisation de ces deux méthodes s'appuie sur plusieurs composantes proposées dans la

Partie 1. Pour la détection des changements de plan, nous avons utilisé l'estimation robuste du mouvement extrait du flux compressé et l'extrapolation du mouvement pour les images I. Le but était de superposer l'image DC de l'image I avec l'image DC de l'image I précédente afin de mesurer leur similarité. Pour pouvoir comparer ces images, nous avons défini une mesure de similarité basée sur l'opérateur de régularisation développé dans le cadre de la restauration dans Chapitre 4. Cette méthode a permis des comparaisons efficaces entre des images DC même dans le cas d'images texturées. La méthode de caractérisation du mouvement dominant de la caméra s'appuie sur l'estimation robuste du mouvement et nous lui avons adjoint un filtrage temporel et un système de prise de décision statistique.

Le Chapitre 8 aborde l'application de notre méthode de construction de mosaïques dans le contexte de la dissimulation d'erreurs de transmission dans les images I. Dans un premier temps, nous avons examiné les modèles classiques de perte d'information dues aux erreurs de transmission. Pour la modélisation des erreurs de transmission de la vidéo compressée en MPEG-1/2, nous avons considéré que le modèle de Gilbert était suffisant. Puis, nous avons utilisé la méthode de construction de mosaïques de super-résolution pour produire une mosaïque à partir d'une séquence contenant des images I partiellement erronées. La redondance temporelle des images, permet de construire une mosaïque qui contient l'information perdue localement dans une image I erronée. Ainsi, les régions de l'image altérées par les erreurs de transmission sont remplacées par les régions correspondantes dans la mosaïque.

Nous avons comparé nos résultats avec la méthode d'extrapolation sélective des fréquences de Meisinger et Kaup [111, 83]. Nous avons obtenu un PSNR moyen supérieur de 20 dB dans les régions altérées, ce résultat est meilleur que la valeur obtenue par la méthode d'extrapolation sélective des fréquences. En outre, dans le cas d'une perte consécutive des macroblocs notre approche fournit des résultats visuellement meilleurs que l'extrapolation sélective des fréquences. Le résultat contient moins d'artefacts de blocs et s'adapte mieux aux motifs des hautes fréquences de l'image.

Enfin, deux annexes terminent cette thèse. L'Annexe A contient la dérivation de la méthode de super-résolution proposée dans le Chapitre 5 et l'Annexe B présente brièvement les standards MPEG suivants avec pour but de monter que les méthodes développées dans cette thèse peuvent être généralisées pour ces standards.

Perspectives

Nous avons récemment commencé des expériences sur des séquences biomédicales. Nous avons appliqué nos méthodes de super-résolution à des images acquises par Imagerie Résonance Magnétique (IRM) afin d'améliorer la résolution dans le plan d'image et de restaurer le flou engendré par le mouvement des organes dû à la respiration et/ou à la circulation sanguine. Les premiers résultats sont prometteurs et montrent des perspectives intéressantes pour une recherche approfondie.

La méthode proposée, pour la construction de la mosaïque suppose que le choix des objets représentatifs est effectué. Dans ce cas, notre méthode ne peut être considéré comme

automatique. La prochaine étape est une automatisation complète de notre méthode en sélectionnant automatiquement les objets représentatifs. Les perspectives de recherche portent alors sur le choix de ces objets, nous pensons mesurer quantitativement la segmentation de ces objets pour élire le meilleur au sens de cette mesure.

La vidéo de haute définition (HD) qui est le nouveau format de diffusion devient de plus en plus populaire. Dans nos expériences, nous avons observé que si une séquence HD est encodée en MPEG-2, l'information de mouvement ne peut pas être récupérée car l'estimation de mouvement de l'encodeur MPEG-2 échoue. Par contre, de nouveaux standards de compression comme H.264/MPEG-4 AVC ont été développés avec une estimation de mouvement améliorée et plus fiable. Nous avons brièvement exposé dans l'Annexe B ces nouveaux standards MPEG. Nous pensons que si un décodage partiel approprié est réalisé les méthodes proposées dans cette thèse peuvent être étendues à ces standards. Une autre perspective de ce travail est donc l'extension de la méthode de construction de mosaïques de super-résolution aux standards MPEG suivants comme H.264/MPEG-4 AVC ou H.264/MPEG-4 SVC afin de restaurer le flou de mouvement.

Nous avons présenté la dissimulation d'erreurs de transmission comme domaine d'application supplémentaire de notre méthode de super-résolution. Cependant, une recherche plus approfondie mérite d'être menée. La première extension évidente de nos travaux serait de considérer les images P et B ainsi que les objets en mouvement.

Enfin d'autres domaines d'applications peuvent être considérés, comme par exemple la transmission de la vidéo sur des réseaux bas-débit. Dans ce cas, seulement des images de basse résolution peuvent être transmises et un algorithme de super-résolution est appliqué lors de la réception afin d'obtenir la pleine résolution de la vidéo. Une autre application dans le domaine de la télésurveillance concernerait la restauration/augmentation de résolution de zones d'intérêt ainsi que leur suivi.

Super-resolution mosaicing from low-resolution video. Application to video summarisation and error concealment.

Abstract: The digitisation of existing videos, together with the explosive development of multimedia network services such as digital video broadcast or mobile communications, made available tremendous volumes of video in compressed form. This requires efficient indexing and browsing tools, but indexing before encoding is not usual in the industry. A traditional approach is the full decoding of these videos in order to index them afterwards. This is very costly and thus not feasible in real time. Moreover, important information for example motion, lost during decoding, is reestimated although it was already present in the compressed bitstream. Hence, our objective in this PhD is the reuse of data of the MPEG compressed bitstream for the purpose of fast indexing and browsing. More precisely, we extract DC coefficients and motion vectors.

In this PhD, we are in particular interested in the construction of mosaics using DC images of I-frames. A mosaic is constructed by aligning and warping the images of a sequence upon each other in a single reference coordinate system. The latter is generally aligned with one of the input images: the reference image. One single image results giving a global view of the sequence. To this end, we propose in this thesis a complete framework for the mosaic construction from MPEG-1/2 compressed video which takes into account various problems occurring in real video sequences such as moving objects or illumination changes.

An essential task for mosaic construction is the motion estimation between each image of the sequence with respect to the reference image. Our method is based on the robust estimation of global camera motion from P-frames motion compensation vectors. Nevertheless, the estimated motion for a P-frame may be inaccurate as it strongly depends on the accuracy of the encoded motion vectors. We detect the concerned images taking into account the DC coefficients of the associated encoded motion compensation error and propose two methods to correct this motion.

A mosaic constructed from DC images is of very low resolution and suffers from aliasing due to the nature of DC images. In order to increase its resolution and to improve its visual quality, we apply a super-resolution method which is based on iterative backprojections. Super-resolution methods are also based on the alignment and warping of the images of a sequence upon each other, but it is combined with an image restoration. Therefore, we have developed a new estimation method for blur due to the camera motion and a corresponding method for restoration in the frequency domain.

Frequency domain restoration methods allow the restoration of global blurs, but in the case of objects moving independently of the camera motion local blurs appear. For this reason, we propose a new super-resolution algorithm derived from the iterative spatial domain restoration method of Van Cittert and Jansson allowing the restoration of local blurs. Based on a segmentation of moving objects, we restore separately the background mosaic and foreground objects. Consequently, we enhanced our blur estimation method.

In a first step, we applied our method to video summarisation for the purpose of fast

browsing of the video using mosaics. Then, we showed how the reuse of intermediary results can be employed for other indexing tasks such as the shot boundary detection on I-frames and the characterisation of camera motion. Finally, we elaborated the area of transmission error concealment. Our approach consists in constructing a mosaic during the decoding of a shot; in case of data lost, the missing information can be concealed thanks to this mosaic.

Keywords: Indexing, MPEG, mosaics, super-resolution

Discipline: Computer Science

LaBRI (UMR CNRS 5800)
Université Bordeaux 1
351, cours de la Libération
33405 Talence Cedex (FRANCE)

Contents

List of Main Notations

Our notation may be in conflict with the notation in the state of the art as we mainly kept the notation of the authors.

Futhermore, we simplify in some cases the notation for the application of operators by the sign "." that is the notation $B.A.X$ corresponds in this thesis to $B(A(X))$.

Chapter 2

B 8×8 block of DCT coefficients, see Equation (2.11), page 18

DC DC image, see Equation (2.12), page 19

DCT Discrete cosine transform, see Equation (2.1), page 12

Chapter 3

∇_{r} Roberts gradient, see Equation (3.70), page 57

μ Number of available pixels, see Equation (3.77), page 62

ϱ Tukey function, see Equation (3.25), page 37

θ Motion model, see Equation (3.23), page 37

a_j Motion parameter, see Equation (3.23), page 37

$\mathbf{d}$ Motion compensation vector, see Equation (3.23), page 37

$\mathbf{E}$ Identity matrix, see Equation (3.60), page 49

$\mathbf{H}$ Observation matrix, see Equation (3.30), page 38

I Image, see Equation (3.78), page 63

M Mosaic, see Equation (3.78), page 63

O_{b} Characteristic background function, see Equation (3.66), page 55

O_{l} Object label mask, see Equation (3.65), page 55

$\mathbf{r}$ Residual, see Equation (3.24), page 37

T Geometric transformation, see Equation (3.61), page 49

$\mathbf{V}$ Vector of the noise, see Equation (3.30), page 38

$\mathbf{W}$ Matrix of the weights, see Equation (3.31), page 38

$\mathbf{Z}$ Vector of the measures, see Equation (3.30), page 38

Chapter 4

$*$ Convolution operator, see Equation (4.55), page 93

$\bar{\epsilon}$ Error measure, see Equation (4.59), page 94

Ψ Penalty function for regularisation, see Equation (4.99), page 107

ς Upsampling factor, see Equation (4.57), page 93

A Antialiasing regularisation operator, see Equation (4.100), page 107

b Blur size, see Equation (4.87), page 100

B Point spread function, see Equation (4.55), page 93

$\mathrm{B}_{\mathrm{Gauss2D}}$ Isotropic Gaussian blur PSF, see Equation (4.60), page 95

$\mathfrak{B}_{\mathrm{Gauss2D}}$ Isotropic Gaussian blur MTF, see Equation (4.61), page 95

$\mathrm{B}_{\mathrm{Gauss}}$ Anisotropic Gaussian blur PSF, see Equation (4.62), page 95

$\mathfrak{B}_{\mathrm{Gauss}}$ Anisotropic Gaussian blur MTF, see Equation (4.64), page 95

$\mathrm{B}_{\mathrm{box}}$ Linear Gaussian blur PSF, see Equation (4.67), page 96

$\mathfrak{B}_{\mathrm{sinc}}$ Linear Gaussian blur MTF, see Equation (4.69), page 96

e Edge response, see Equation (4.84), page 100

F Super-resolution image, see Equation (4.55), page 93

FT Fourier transform, see Equation (4.91), page 101

G Low-resolution image, see Equation (4.55), page 93

R Spatial domain restoration filter, see Equation (4.57), page 93

$\mathfrak{R}$ Frequency response of the restoration filter, see Equation (4.93), page 102

$\mathfrak{R}_{\mathrm{pinv}}$ Pseudo-inverse filter, see Equation (4.92), page 102

$\mathfrak{R}_{\mathrm{Wiener}}$ Wiener filter, see Equation (4.94), page 102

S Upsampling operator, see Equation (4.57), page 93

S^{-1} Downsampling operator, see Equation (4.55), page 93

Chapter 5

$\bar{\epsilon}_{\overline{\mathrm{roi}}}$ Error measure for the background mosaic, see Equation (5.40), page 134

ϵ_{roi} Error measure for the object, see Equation (5.41), page 134

$\mathrm{B}_{\mathrm{roi}}$ PSF of the ROI, see Equation (5.30), page 131

F_{roi} ROI in the super-resolution image, see Equation (5.30), page 131

G_{roi} ROI in the low-resolution image, see Equation (5.30), page 131

$\mathrm{T}_{\mathrm{roi}}$ Geometric transformation of the ROI, see Equation (5.30), page 131

Chapter 1

Introduction and Motivation

Today tremendous volumes of video in compressed form are available due to the digitisation of existing videos and the explosive development of multimedia network services such as digital video broadcast or mobile communications. This requires efficient indexing and browsing tools, but indexing before encoding is not usual in the industry. Thus, the traditional approach is the full decoding of these videos in order to index them afterwards. This is very costly and thus not feasible in real time, so that a huge quantity of compressed video remains unindexed.

Up to now the MPEG-2 standard [117] is the most frequent encoding for video available in archives or coming from broadcast channels. It consists in a hybrid coding scheme based on motion compensation and transform coding. Thus, important information which can be used for indexing such as motion is lost during decoding. Typically, it is reestimated subsequently, although it was already present in the compressed bitstream. Hence, the challenge is to index those compressed videos without full decoding and instead extracting useful data by only partially decoding the compressed stream.

Some significant work in the compressed domain has been already been presented for different indexing tasks. The common point of these methods is that they are mainly based on the extraction of *DC images* and/or *motion vectors* of MPEG-1/2 compressed video. A DC image is eightfold smaller than the original video frame whereas a pixel in the DC image corresponds to the mean value of an 8×8 block in the original frame. DC images and motion vectors can be easily extracted by partially decoding MPEG-1/2 compressed video.

Yeo and Liu [208] stated that even at DC resolution, global image features useful for image processing and indexing operations are well preserved. For instance, Bescós [12] proposed an efficient method to detected shot boundaries in compressed video based on the use of DC images. Often, MPEG motion vectors are considered as to inaccurate for further use. However, methods presented in literature demonstrate that when appropriate robust methods are developed their use is justified. Durik and Benois-Pineau [38] developed a robust

method to estimate global camera motion from MPEG motion vectors for the purpose of shot boundary detection. In [33], Coimbra and Davies presented a method to detect pedestrians using MPEG-2 motion vectors.

In this PhD thesis, we are particularly interested in the construction of mosaics for video summarisation of MPEG-1/2 compressed video. A *mosaic* image is constructed by aligning and warping the frames of a shot upon each other in a single reference coordinate system. A single image results giving a panoramic view of the shot. In the last decade, powerful methods have been presented for mosaic construction. This can be explained by the fact that this kind of video representation allows a quick understanding of the scene through the interpretation of a static image. Mosaic representations are a part of the MPEG-4 video coding standard [118] as they allow an efficient encoding of a shot by decomposing it into the background mosaic and foreground objects. Furthermore, they are defined as a shot descriptor in the MPEG-7 standard [119] to be used for video browsing and indexing.

Since the works of Peleg and Herman [139] and Sawhney and Ayer [156], mosaicing methods seem to have reached maturity. Nevertheless, most mosaicing methods proposed in literature perform on raw video, amongst others the method of Irani and Anandan [72] proposed for browsing and indexing raw video. Pilu [143] proposes a mosaicing method based on MPEG-1 motion vectors, yet frames are completely decoded for mosaic construction. In contrast to these methods, we focus in this thesis on the mosaic construction from MPEG-1/2 compressed video without completely decoding the compressed stream. We state that mosaics constructed using raw or decoded video frames are often too large to display on the screen without scrolling or downsampling. Therefore, we propose creating mosaics of a lower resolution providing a scene overview appropriate for browsing in a video document. This is achieved by using DC images of I-frames as input image sequence. Moreover, we estimate global camera motion from P-frame motion vectors for image warping.

The originality of our method is that we propose a complete framework to construct mosaics from MPEG-1/2 compressed video at the basis of DC images of I-frames and P-frame motion vectors. We take into account different aspects and problems that can occur during mosaic construction such as the noisiness and inaccuracy of motion vectors, the removal of moving objects and their possible inaccurate segmentation, or illumination changes between input frames. In order to obtain a complete description of the shot, representative objects are inserted in the mosaic afterwards. A fast and efficient algorithm results.

DC images are mainly characterised by very low resolution and aliasing artefacts, but also *blur* arises due to the camera motion during exposure time and block averaging. When a mosaic is constructed from DC images, the resolution may be too low to give a scene overview and the image quality may not be satisfying due to the strong degradations of the DC images.

Several methods exist to increase the resolution of video frames. The easiest way to create an image with a higher resolution is a simple interpolation e.g. a bilinear or bicubic interpolation. However, this does not produce satisfying results due to aliasing and loss of high frequencies. Fassino and Montanvert [49] proposed a method to increase the resolution of MPEG compressed video. The resolution of I-frames is increased by projecting

the decoded I-frame onto convex sets. Then, the resolution of P and B-frames is increased by scaling the size of motion vectors and macroblocks used in the motion compensation process. Anyhow, in case of video it is possible to exploit the information of neighbouring frames in order to obtain an image of higher resolution using *super-resolution* reconstruction methods. Assuming that each image of the sequence provides a slightly different view at the same scene, then they can be combined into one higher-resolution image. This allows to restore high-frequencies and an image of superior visual quality results.

Powerful super-resolution algorithms have been developed for raw video, e.g. by Patti et al. [137], and Irani and Peleg [75], and for compressed video, e.g. by Segall et al. [164]. Thus, we apply a super-resolution algorithm to our DC-resolution mosaics to increase the resolution and restore the blur. Capel and Zisserman [22], and Zomet and Peleg [215] already proposed applying a super-resolution algorithm to mosaics constructed from raw video. In our case of compressed video, the global approach is the same. Nevertheless, the estimation of motion from compressed video, the choice of an adequate blur model and of a restoration filter is different.

In this work, we present two super-resolution methods. The first one is based on the iterative backprojections of Irani and Peleg [75] incorporating *restoration in frequency domain*.

Due to the frequency domain restoration, this method only allows to restore global blurs. Unfortunately, in video, we observe more complex situations where local blurs appear e.g. due to object motion. Hence, we propose a second super-resolution method derived from the *spatial domain restoration* method of Van Cittert [31] and Jansson [78]. Performing restoration in spatial domain this method allows the processing of local blurs. Nevertheless, the motion of objects can be very complex and in case of DC images the objects are typically represented by only few pixels, so that its it very difficult to estimate the motion of objects and to superimpose them accurately enough for super-resolution. In contrast to [75], where moving objects supposed to undergo 2D parametric motion are super-resolved, we propose a generic method. We propose restoring and increasing the resolution of moving objects by a single image restoration method whereas the scene background is super-resolved.

The origin of blur in DC images is twofold. On the one hand, the blur comes from full-resolution frames due to motion during exposure time. On the other hand, DC images represent a low-pass version of the full-resolution frames in which the low pass is done locally inside a block of 8×8 pixels. Therefore, the overall blur is a result of these two phenomena. To this end, we propose several blur models to describe the real overall blur function. Moreover, we propose an efficient method to *estimate global and local blur parameters* in motion direction. We evaluate the performance of the blur models in combination with different restoration filters for both super-resolution methods.

In addition to video summarisation, we address in this PhD thesis other applications of our mosaicing method. First, we show how intermediary results can be used for related indexing tasks, namely the detection of shot boundaries on I-frames and the characterisation of camera motion. Another application we studied in this thesis is the use of mosaics for error concealment. Our approach consists in constructing progressively a mosaic during the decoding of a shot; in case of data lost, the missing information can be concealed thanks to

this mosaic.

This PhD thesis is structured in two parts. Part 1 presents the construction of mosaic from MPEG-1/2 compressed streams with regard to video summarisation, whereas Part 2 addresses additional applications of our mosaicing method.

Part 1 is organised as follows: In Chapter 2 we describe the MPEG-1 and MPEG-2 video coding standards and explain what kind of information we extract from the compressed stream for mosaic construction. We present the complete framework for mosaic construction from MPEG-1/2 compressed streams in Chapter 3. This framework incorporates a robust registration of the sequence of DC images of I-frames using P-frame motion vectors, the segmentation of moving objects and the correction of illumination changes between frames of the input sequence. Chapter 4 presents our first super-resolution method based on restoration in frequency domain. We present different blur models that we consider for DC images and propose an appropriate estimation of the blur parameters in motion direction. Moreover, we test the performance of classical restoration filters in combination with the different blur models. In Chapter 5 we present our second super-resolution method incorporating a spatial domain restoration. To realise the restoration of local blurs, we enhance the blur estimation method of Chapter 4. Moreover, we compare the performance of this super-resolution method with that one presented in Chapter 4.

Part 2 starts with a short introduction of the considered application areas in Chapter 6. Chapter 7 presents a method for shot boundary detection on I-frames and a method for the characterisation of dominant camera motion. Both methods are based on the methods developed in Part 1. Chapter 8 addresses the application of our mosaicing method in the context of error concealment.

Finally, we conclude our work in Chapter 9 stressing on our contributions. Some perspectives of future work are given.

Part I

Mosaicing from the Compressed Stream

Chapter 2

MPEG-1/2 Compressed Streams as a Source of Low-Resolution Data

Massive digitisation of existing video, together with an explosive development of multimedia services via heterogeneous network broadband (such as digital video broadcast) or narrowband (such as mobile communications), made available tremendous volumes of video in compressed form. Indeed, be it in the field of digital libraries or in video communications, a suitable compression standard is always applied to reduce the bit rate while preserving the quality of video as a function of taste. Thus, in digital video archives and in broadcast MPEG-2 is today the common standard, while browsing copies or preview applications still use MPEG-1. Hence, as we are working on MPEG-1/2 compressed streams, we introduce the MPEG-1 and MPEG-2 standards in this chapter. Furthermore, we explain what information we recover directly from the compressed stream for our work.

In 1993, the *Moving Picture Expert Group* (MPEG), founded by the *International Organization for Standardization* (ISO) in 1988, specified a video and sound compression standard, referred to as MPEG-1 [114]. This standard was designed for the application of digital video storage and playback on CDs. It was quickly clear that this standard was not sufficient for current applications. Hence, the MPEG was working on a second compression standard, MPEG-2 [117], addressing a wider range of applications such as digital video storage, video broadcast and communication. While MPEG-1 was suited for coding of non-interlaced video at lower resolutions and bit rates, MPEG-2, completed in 1995, was suited for coding of interlaced video at higher resolutions and bit rates.

In the remainder of this chapter, we present the MPEG-1 and MPEG-2 coding standards in Section 2.1. Then, we explain in Section 2.2 the data we extract from the MPEG-1/2 compressed streams which will be furthermore used in this work. Section 2.3 concludes this chapter.

2.1 The MPEG-1 and MPEG-2 Video Compression Standards

The MPEG coding algorithm is a hybrid coding algorithm mainly based on motion compensation and transform coding. Motion compensation is used to reduce the temporal redundancy between frames by predicting the current video frame from a reference frame. The resulting error signal, the difference between the current frame and its prediction, is quantised and then transform coded in order to reduce the spatial redundancy in the frame. Finally, entropy coding is applied to the data to remove the statistical redundancy. Figure 2.1 illustrates the structure of the encoder.

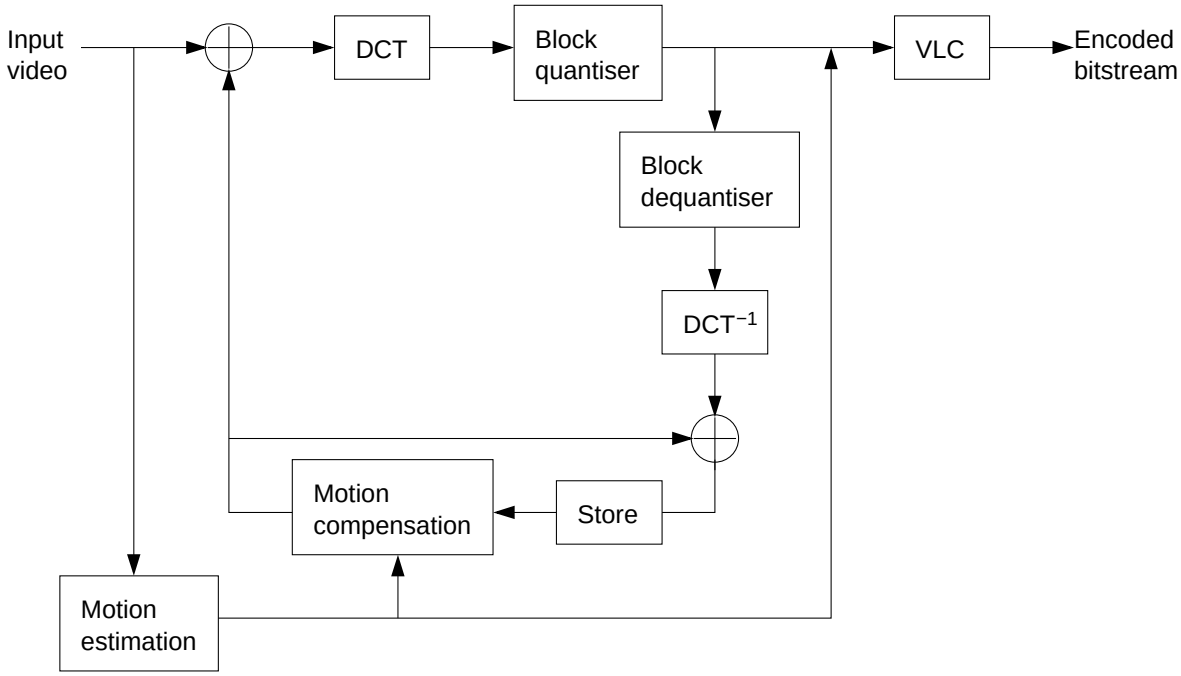


Figure 2.1: The MPEG-1/2 encoder structure [172, 188].

The MPEG-2 standard includes the MPEG-1 standard. It is more generic and supports the efficient coding of interlaced video and scalability. Scalability allows to combine several versions of different quality or resolutions of a video in one stream. MPEG-2 defines a non-scalable and a scalable syntax, whereas the non-scalable syntax is defined as a super-set of the MPEG-1 syntax. Due to its wide range of applications and supported image formats, MPEG-2 introduces the concepts of profiles and levels (defining conformance and performance limits) since certain applications do not require all the coding tools.

Now, we will introduce the properties of the compressed stream and the main coding tools for reducing temporal, spatial and statistical redundancy of the video data. More information on the MPEG-1/2 standards can be found in [172].

2.1.1 Sampling Formats

In MPEG a *frame* consists of three rectangular matrices of integers, a luminance matrix $\mathbf{Y}$, and two chrominance matrices $\mathbf{Cb}$ and $\mathbf{Cr}$. The advantage of the $\mathbf{YCbCr}$ representation is that the $\mathbf{Cr}$ and $\mathbf{Cb}$ components may be represented with a lower resolution than $\mathbf{Y}$ as the human visual system is less sensitive to colour than luminance. This reduces the amount of data required to represent the chrominance components without having an obvious effect on the visual quality. Hence, different sampling patterns exist to represent the colour components $\mathbf{Cb}$ and $\mathbf{Cr}$. The MPEG-2 standard supports the 4:4:4, 4:2:2 and 4:2:0 sampling formats, while MPEG-1 standard supports only the 4:2:0 sampling format.

The sampling patterns are illustrated in Figure 2.2. 4:4:4 sampling means that the three components ($\mathbf{Y}$, $\mathbf{Cb}$ and $\mathbf{Cr}$) have the same spatial resolution. Hence, for each pixel position each component is available. So, the full fidelity of the chrominance components is preserved. In 4:2:2 sampling, the chrominance components have the same vertical resolution as the luminance but half the horizontal resolution. In the 4:2:0 sampling format, $\mathbf{Cb}$ and $\mathbf{Cr}$ each have half the horizontal and vertical resolution of $\mathbf{Y}$.

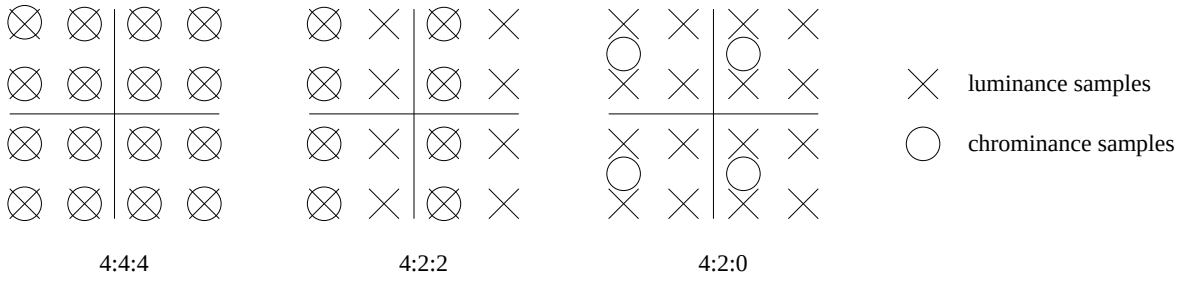


Figure 2.2: The 4:4:4, 4:2:2 and 4:2:0 sampling patterns [152].

2.1.2 Coding Structure

The coding structure of a MPEG-1/2 compressed bit stream is hierarchically organised. It is decomposed into six levels: the video sequence, the group of pictures, the picture, the slice, the macroblock, and the block.

The highest syntactic structure of the coded bitstream is the *video sequence*. It provides general information with respect to the sequence, including the image format (such as the image size, the pixel ratio, and the number of images per second) and the bit rate. It is divided into *groups of pictures* (GOP). A GOP may include three types of *pictures* (also called frames): *Intra-coded frames* (I-frames), *predictive-coded frames* (P-frames) and *bidirectionally predictive-coded frames* (B-frames). I-frames are coded without reference to other frames. They provide access points to the coded sequence where decoding can begin, but are coded with only moderate compression only by transform coding. P-frames are coded more efficiently using motion-compensated prediction from a past reference picture, the directly preceding I or P-frame. B-frames provide the highest degree of compression but require both, a past

and future reference picture, for motion compensation. These are respectively the preceding and succeeding I or P-frames. A GOP starts with an I-frame followed by a fixed number of P and B-frames as shown in Figure 2.3. Regardless of the frame type, each frame is divided into slices. Each *slice* consists of adjacent macroblocks in a row. That can be only one macroblock or up to all macroblocks of the row. Slices are used for synchronisation in case of transmission errors. A *macroblock* refers to a 16×16 section of the luminance component and the spatially corresponding chrominance components. Thus, according to the different chrominance formats presented above, a macroblock consists of a variable quantity of blocks. For the 4:2:0 sampling format, the macroblock consists of 6 blocks (4 $\mathbf{Y}$, 1 $\mathbf{Cb}$, 1 $\mathbf{Cr}$). A 4:2:2 macroblock consists of 8 blocks (4 $\mathbf{Y}$, 2 $\mathbf{Cb}$, 2 $\mathbf{Cr}$), and a 4:4:4 macroblock consists of 12 blocks (4 $\mathbf{Y}$, 4 $\mathbf{Cb}$, 4 $\mathbf{Cr}$). A *block* represents a 8×8 section of a luminance or chrominance component of the image. This section can refer to original pixel values, to residuals or to transform coefficients.

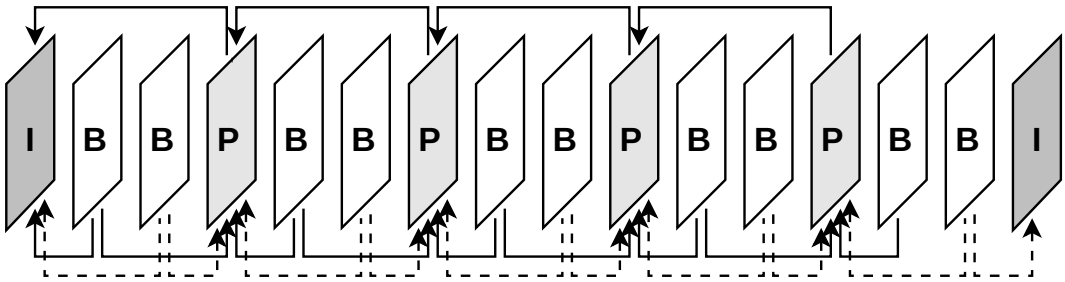


Figure 2.3: A typical GOP structure of 15 frames in MPEG-2 [152].

2.1.3 Temporal Redundancy

The principle of reducing temporal redundancy is to predict the current video frame by exploiting the similarities between two frames. Therefore, motion vectors are estimated using a block matching algorithm which describes how motion is compensated. Based on these motion vectors, a residual frame is computed by subtracting the predicted frame from the current frame. The basic unit for motion-compensated prediction in MPEG-1/2 is the macroblock.

Block Matching

The motion vector for each macroblock is estimated from two original luminance frames using a block matching algorithm. Motion estimation of a macroblock consists in finding a 16×16 sample region in a reference frame that closely matches the current macroblock. Therefore, a search area in the reference frame centered on the current macroblock position is searched. Then, the 16×16 region within the search area that minimises a matching criterion is chosen as the best match. The motion vector $\mathbf{d} = (d_x, d_y)^{\mathrm{T}}$ describes the position of the best matching region relative to the current macroblock position. These motion vectors provide the first source of data we extract from the compressed stream. Once the motion vector $\mathbf{d}$ is

estimated, the pixel values for the current macroblock can be predicted from the previously decoded frame.

Motion Compensation

The best matching region in the reference frame is subtracted from the current macroblock to produce a residual macroblock (luminance and chrominances) that is encoded together with the motion vector. During encoding, the residual is encoded and decoded and added to the matching region to form a reconstructed macroblock which is stored as a reference for further motion-compensated prediction. It is necessary to use a decoded residual to reconstruct the macroblock in order to ensure that encoder and decoder use an identical reference frame for motion compensation. If there is a significant change between the reference and the current frame, it might be more efficient to code the macroblock without motion compensation. Then, the encoder may choose to encode the macroblock in *intra mode*.

Temporal redundancy reduction is followed by spatial and statistical redundancy reduction which is illustrated in Figure 2.4.

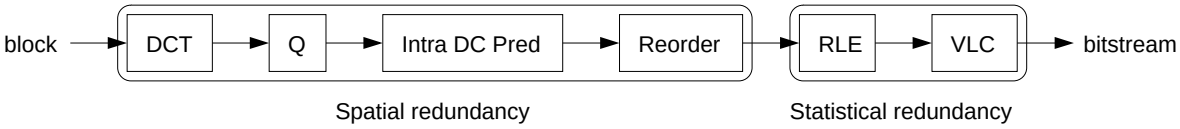


Figure 2.4: The encoding of a block.

2.1.4 Spatial Redundancy

Spatial redundancy is reduced using similarities between neighbouring samples in a frame. Therefore, an orthogonal transform is applied to the samples in order to decorrelate their values. Afterwards, the transform coefficients are quantised to remove insignificant values, thus reducing the information.

Transform Coding

For transform coding, MPEG-1/2 uses the Discrete Cosine Transform (DCT) operating on blocks (first step in Figure 2.4). In case of intra mode the block contains image samples and in case of inter mode the block contains residual samples. The DCT of a $M \times M$ block is defined as:

$$\mathrm{DCT}(u,v) = \frac{2}{M}\mathrm{C}(u,v)\sum_{x=0}^{M-1}\sum_{y=0}^{M-1} f(x,y)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)v\pi}{2M} \tag{2.1}$$

where x,y are spatial coordinates in the image domain, u,v frequencies in the transform domain, and:

$$\mathrm{C}(u,v) \quad = \quad \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \text{ and } v = 0 \\ 1 & \text{otherwise} \end{cases}$$

The inverse transform (DCT^{-1}) is:

$$f(x,y) = \frac{2}{M} \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} \mathrm{C}(u,v)\mathrm{DCT}(u,v) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2M} \tag{2.2}$$

The first DCT coefficient of a block at the frequency $(0,0)$ is called the *DC coefficient*, the other coefficients are called *AC coefficients*. DC coefficients are the second source of data we use in this work. Then, the DCT coefficients are quantised in order to reduce the range and remove insignificant values. Normally, the DCT is perfectly invertible, but due to the quantisation of transform coefficients DCT^{-1} does not recover the original pixels values. So called *quantisation noise* results.

Quantisation

A quantiser (step two in Figure 2.4) maps a signal to a quantised signal with a reduced range with the objective of representing the quantised signal with fewer bits than the original signal. Consequently, the precision of the image data is reduced and insignificant DCT coefficients such as near-zero values are removed.

Figure 2.5 shows the default quantisation matrices for MPEG-2. For intra blocks, where the high energy values are typically located in the low frequencies, the quantisation matrix of Figure 2.5(a) is used. Its function is to quantise high frequencies with coarser quantisation steps. That will suppress high frequencies with no subjective degradation as the human visual system is not very perceptive to high frequencies. Since the energy of transform coefficients in inter mode coding is more uniformly distributed, the constant quantisation matrix of Figure 2.5(b) is used.

The human visual system is very perceptive to luminance changes. Therefore, the DC coefficients of intra blocks, represented by 11-bit values, are quantised in a different manner than the other coefficients. They may be quantised to 8, 9, or 10 bits according to the parameter settings. Thus, the quantised DC value $\mathrm{DCT_Q}(0,0)$ is calculated as:

$$\mathrm{DCT_Q}(0,0) = \mathrm{round}\left(\frac{\mathrm{DCT}(0,0)}{P}\right) \tag{2.3}$$

where the constant P depends on the DC precision. Its values are 8, 4 or 2, respectively, for a precision of 8 bits, 9 bits or 10 bits. Quantisation is a lossy process since it is not possible to determine the exact value of the original fractional number from the rounded integer. Thus, the DC coefficients we extract are degraded by quantisation noise.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

(a) $\mathbf{Q}_{\text{intra}}$

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

(b) $\mathbf{Q}_{\text{inter}}$

Figure 2.5: Default quantisation matrices [117]: (a) for intra blocks, (b) for inter blocks.

The AC coefficients of intra blocks are first quantised by the quantisation matrix $\mathbf{Q}_{\text{intra}}$ e.g. that one of Figure 2.5(a):

$$\text{DCT}'(u,v) = \text{round}\left(\frac{16 \cdot \text{DCT}(u,v)}{\mathbf{Q}_{\text{intra}}(u,v)}\right), \quad u,v \neq 0 \tag{2.4}$$

and then scaled:

$$\text{DCT}_{\text{Q}}(u,v) = \frac{\text{DCT}'(u,v) + \text{sign}(\text{DCT}'(u,v)) \cdot \text{round}\left(\frac{P \cdot S}{Q}\right)}{2 \cdot S}, \quad u,v \neq 0 \tag{2.5}$$

where S is the quantiser scale, and $P = 3$ and $Q = 4$ in [115].

For inter blocks, the DCT coefficients are first quantised by the quantisation matrix $\mathbf{Q}_{\text{inter}}$ e.g. that one of Figure 2.5(b):

$$\text{DCT}'(u,v) = \text{round}\left(\frac{16 \cdot \text{DCT}(u,v)}{\mathbf{Q}_{\text{inter}}(u,v)}\right) \tag{2.6}$$

and then scaled:

$$\text{DCT}_{\text{Q}}(u,v) = \frac{\text{DCT}'(u,v)}{2 \cdot S} \tag{2.7}$$

The quantised DCT coefficients DCT_{Q} are then furthermore encoded.

Intra DC Prediction

Low-frequency transform coefficients of neighbouring intra-coded blocks are often correlated. Thus, energy of DCT coefficients in intra blocks can further be reduced by intra DC prediction (step three in Figure 2.4). Therefore, the DC coefficient of the current block (X in Figure 2.6) is in addition predicted with respect to the DC coefficient of the previous encoded block (A). The residual value $\Delta\text{DCT}_{\text{Q},j}(0,0)$ is encoded:

$$\Delta\text{DCT}_{\text{Q},j}(0,0) = \text{DCT}_{\text{Q},j}(0,0) - \text{DCT}_{\text{Q},j-1}(0,0) \tag{2.8}$$

where $\text{DCT}_{\text{Q},j}(0,0)$ is the quantised DCT coefficient of the current block (X in Figure 2.6) and the quantised DCT coefficient of the previous encoded block (A in Figure 2.6).

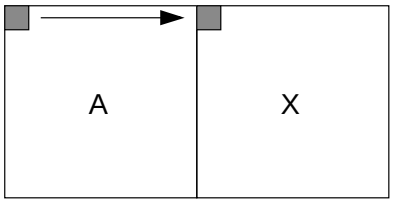


Figure 2.6: Prediction of DC coefficients in intra blocks.

Reordering

The output of the quantiser is a sparse array containing a few non-zero coefficients and a large number of zero-valued coefficients. The significant DCT coefficients of a block of image or residual samples are typically in the low frequency positions around the DC coefficient. Reordering (step four in Figure 2.4) is applied prior to entropy coding in order to group together non-zero coefficients and to enable the efficient representation of zero-valued coefficients.

The optimum reordering path (scan order) depends on the distribution of non-zero DCT coefficients. For a typical frame block a zigzag scan is applied. Starting with the DC coefficient, each quantised coefficient is copied into a one-dimensional array in the order as illustrated in Figure 2.7(a). Then, the non-zero coefficients tend to be grouped together followed by long sequences of zeros.

However, Zigzag scan is not ideal for field blocks (MPEG-2) since the coefficient distribution is skewed. This is because field pictures have stronger high-frequency components in the vertical axis due to the subsampling in the vertical direction. A modified scan order as illustrated in Figure 2.7(b) is used, in which coefficients on the left hand side of the block are scanned before those on the right hand side.

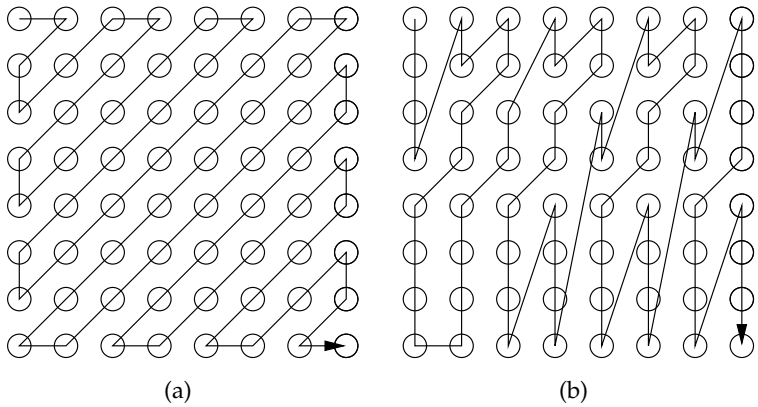


Figure 2.7: Reordering [172]: (a) Zigzag scan order for frame blocks, (b) alternate scan order for field blocks.

2.1.5 Statistical Redundancy

Statistical redundancy in the data is reduced by entropy coding. First, Run Length Coding (RLE) is applied to the reordered quantised DCT coefficients in order to encode efficiently the zero sequences produced by the reordering process. This is followed by Variable Length Coding (VLC).

Run Length Coding

The output of the reordering process is an array that typically contains one or more clusters of non-zero coefficients near the start, followed by sequences of zero coefficients. During RLE (step five in Figure 2.4) the large number of zero coefficients is represented more compactly as a series of (run,level) pairs where run indicates the number of zeros preceding a non-zero coefficient and level indicates the magnitude of the non-zero coefficient.

Variable Length Coding

The principle of VLC (step six in Figure 2.4) is that symbols with a higher occurrence are encoded with code words of shorter length as rare symbols. Then, the pairs of (run,level) are encoded by a Huffman-type entropy coder. VLC is also applied to encode the motion vectors.

2.1.6 Scalability

Scalability allows the reconstruction of video only from pieces of the total bitstream. This is achieved by structuring the bitstream in two layers, a standalone base layer and an enhancement layer. If there is only one layer, the coded video data is called non-scalable video bitstream. If there are two layers, the coded video data is called a scalable bitstream. The objective of scalable video coding is to combine video at different qualities or resolutions in one bitstream. An additional advantage is the ability to provide resilience to transmission errors. The more important data of the base layer can be transmitted on a channel with better error performance, while the less critical enhancement layer data can be sent over a channel with poor error performance. The drawback is that some of the coding efficiency is lost as a result of the extra overhead.

Spatial Scalability

Spatial scalability consists in generating several spatial resolution video layers from a single video source. The base layer provides the basic spatial resolution. The upsampled reconstructed base layer is then used as prediction of the enhancement layer. If both base layer and enhancement layer are decoded the full spatial resolution of the video is achieved.

SNR Scalability

The objective of SNR scalability is to generate a video stream of different qualities where all layers have the same spatial resolution. The base layer provides the basic video quality and the enhancement layer when added to the base layer improves the quality of the video.

Temporal Scalability

Temporal scalability generates a video stream of different temporal resolutions. The video frames are partitioned into the two layers. The base layer provides the basic temporal rate and the enhancement layer is coded with temporal prediction with respect to the base layer. These layers when decoded and temporal multiplexed provide the full temporal resolution of the video source.

2.1.7 Profiles and Levels

Due to an important number of potential applications and image formats supported by MPEG-2, the definition of profiles and levels was introduced. A *profile* is a defined subset of the bitstream syntax and so of the coding options specified by the standard. This signifies that each profile allows the use of a subset of the coding tools which is adapted to a certain application. A *level* is a defined set of constraints imposed on the parameters in the bitstream. Table 2.1 summarises the main characteristics of the profiles and levels defined in the MPEG-2 standard.

Profiles / Levels	Simple 4:2:0	Main 4:2:0	SNR Scalable 4:2:0	Spatially Scalable 4:2:0	High 4:2:2
High		1920/1152/60 1/0/0			1920/1152/60 3/1/1
High-1440		1440/1152/60 1/0/0		1440/1552/64 3/1/1	
Main	720/576/30 1/0/0	720/576/30 1/0/0	720/576/30 2/0/0		720/576/30 3/1/1
Low		352/288/30 1/0/0	352/288/30 2/0/1		

Table 2.1: Overview of the profiles and levels in MPEG-2 [9]: For each combination of profile and level 1) the image size (number of pixels in a row/ number of pixels in a column/ number of frames per second) and 2) the number of layers (total number of layers/ number of layers for spatial scalability/ number of layers for SNR scalability) is shown.

Note that a profile/level combination is usually referred to as "profile@level". In the scope of this PhD, we use video encoded at Main Profile@Main Level or Main Profile@Low Level whereas the latter is compliant with MPEG-1. That is however no crucial constraint for our work, because when the profile changes only adequate partial decoding is necessary to extract our input information: DC coefficients and motion vectors.

2.2 Extraction of Data from MPEG-1/2 Compressed Streams for Video Analysis and Indexing

Lots of video analysis techniques working in the compressed domain are based on the processing of DC coefficients. If only the DC coefficients of the 8×8 block are decoded and divided by 8, an image, the so called DC image, is obtained. The DC image is eightfold smaller than the original frame and a pixel corresponds to the mean value of the pixels in the block. Yeo and Liu [208] showed that even at this resolution, global image features useful for a specific class of image processing operations are well preserved. Operating on these images offers a significant saving of computational time, because of the reduced spatial resolution and only partially decoding the compressed stream.

Another aspect in our work is the need of motion information. Motion can be estimated after full decoding, while motion information has been already available in the compressed stream in form of motion compensation vectors. Thus, the motion compensation vectors can be used in a motion estimation process. This is an additional gain of computational time since full decoding is still omitted and we work only on sparse optical flow.

Nevertheless, the use of motion vectors from the stream is rather challenging as they were not computed for analysis, but for encoding based on a quality criterion. Their quality depends very much on the encoder settings and they are typically noisy. The usage of such motion vectors has already been presented in literature. For instance, Coimbra and Davies [32] approximated optical flow to MPEG-2 motion vectors. In order to overcome the noisiness of the MPEG motion vectors, DCT coefficients are used in a confidence measure for the estimated flow vectors. The method proposed by Pilu [143] fits a global motion model to MPEG-1 motion vectors after filtering. Other examples are the extraction of global motion features of raw MPEG motion vectors [6] or after normalisation [89]. These examples show that MPEG motion vectors can be used if adequate robust methods are developed.

Hence, in this work we consider that data extracted from the MPEG-1/2 compressed streams is sufficient for our purpose i.e. the fast construction of mosaics, namely we use:

- *DC images of I-frames* as input sequence for the mosaic construction
- *P-frame motion vectors* for the estimation of global motion
- *DC images of the encoded P-frame motion compensation error* in a confidence measure for the motion models estimated from MPEG motion vectors
- *DC images of P-frames* for the reestimation of a motion model in the case of low-quality MPEG motion vectors

Figure 2.8 shows the scheme of the MPEG-1/2 decoding process. It is clear, that P-frame motion compensation vectors are easily to extract, just as DC images of I-frames and of the P-frame motion compensation error. Motion compensation vectors can be extracted just after variable length decoding as illustrated in Figure 2.8. The DCT coefficients of intra and residual macroblocks are available after inverse quantisation.

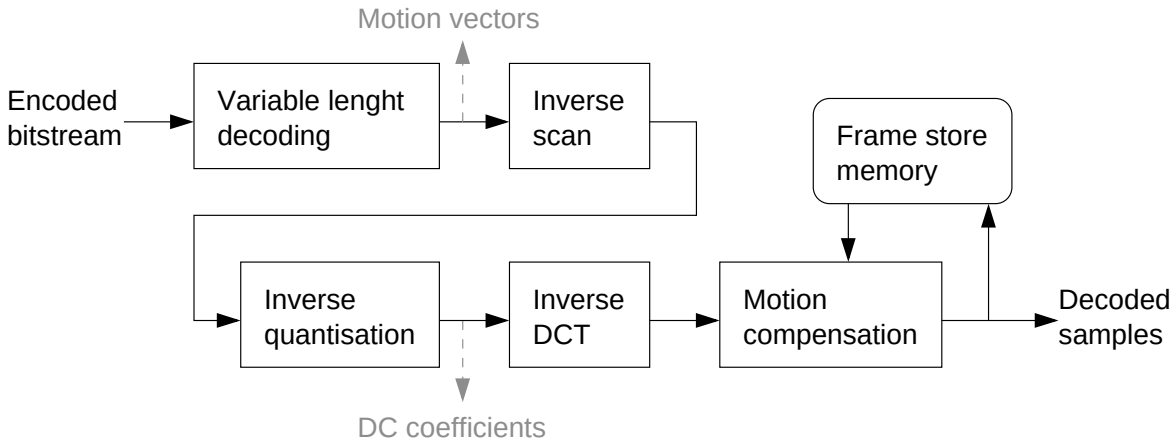


Figure 2.8: The MPEG-1/2 decoding process [114].

In order to extract DC images of I-frames, the compressed stream is parsed for I-frames and for each block in the I-frame the DC coefficient is extracted. Therefore, the quantised DC value is computed inversely to the prediction Equation (2.8). Thus, for the jth block:

$$\mathrm{DCT}_{\mathrm{Q},j}(0,0) = \mathrm{DCT}_{\mathrm{Q},j-1} + \Delta\mathrm{DCT}_{\mathrm{Q},j}(0,0) \tag{2.9}$$

where $\Delta\mathrm{DCT}_{\mathrm{Q},j}(0,0)$ is the encoded difference between the predicted value and the original value of $\mathrm{DCT}_{\mathrm{Q},j}(0,0)$.

Then, DC coefficients are quantised inversely to Equation (2.3):

$$\mathrm{DCT}(0,0) = P \cdot \mathrm{DCT}_{\mathrm{Q}}(0,0) \tag{2.10}$$

Finally, the pixel value of the DC image is $DC = \mathrm{DCT}(0,0)$. The DC images of P-frames can be extracted similarly to I-frames.

To extract this information (P-frames motion vectors, DC images of I-frames, DC images of the motion compensation error of P-frames) from the compressed stream, we used the MPEG Software Simulation Group decoder (MSSG) [121] and inserted output operations at various steps of the decoding process.

Now, the problem is how to extract efficiently the DC images for P-frames from the compressed stream. This means an increase of complexity, since motion vectors have to be taken into account.

The basis of DC image reconstruction is the fact that the DCT is a linear transform. This means that the DCT coefficients of the reconstructed block can be obtained by summing up the DCT coefficients of the residual block B_{diff} and the DCT coefficients of the reference block B_{ref}:

$$B = B_{\mathrm{ref}} + B_{\mathrm{diff}} \tag{2.11}$$

Therefore, in [26] a method is proposed to reconstruct DCT coefficients of a block by DCT domain inverse motion compensation. Based on this, a more sophisticated method for DC image reconstruction is proposed in [177] including DCT domain deinterlacing and DCT domain interlacing.

From (2.11) it is obvious that the DC coefficients of the current frame can be obtained by adding the appropriate DC coefficients of the reference image DC_{ref} to the DC coefficients to the difference image DC_{diff}:

$$DC(x,y) = DC_{\text{ref}}(x,y) + DC_{\text{diff}}(x,y) \tag{2.12}$$

However, due to motion compensation the DC coefficients DC_{ref} are not directly available (see Figure 2.9). The methods of [208, 171] propose to approximate $DC_{\text{ref}}(x,y)$ by a weighted average of the DC coefficients of the blocks pointed by the motion vector:

$$DC_{\text{ref}}(x,y) = \frac{1}{64}\sum_{i=0}^{3}\omega(B_i)DC_{\text{ref}}(B_i) \tag{2.13}$$

where $DC_{\text{ref}}(B_i)$ is the DC coefficient of the block B_i and $\omega(B_i)$ is the number of pixels in the block B_i that is overlapped by the reference block.

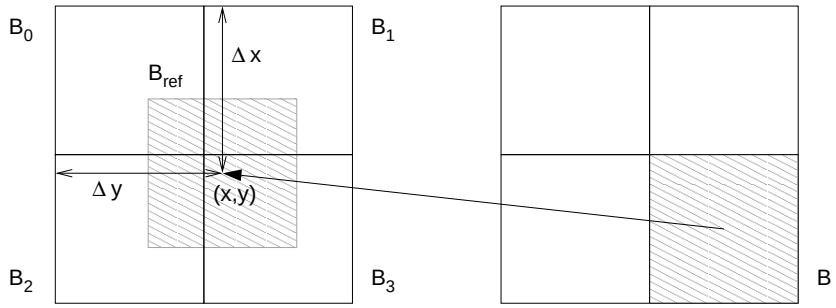


Figure 2.9: Motion-compensated reference block [177].

Based on Equation (2.12), we reconstruct DC images for P-frames. We propose to approximate the DC coefficient DC_{ref} by a bilinear interpolation of the DC coefficients $DC_{\text{ref}}(B_i)$. As the encoded motion compensation vectors are at macroblock basis, we use the scaled motion vector for approximating DC_{ref} in the reference DC image. Thus, DC_{ref} at the position (x,y) is obtained by:

$$\begin{aligned} DC_{\text{ref}} &= (1-\Delta x)(1-\Delta y)\cdot DC_{\text{ref}}(B_0) + (1-\Delta x)\Delta y \cdot DC_{\text{ref}}(B_1) \\ &\quad +\Delta x(1-\Delta y)\cdot DC_{\text{ref}}(B_2) + \Delta x\Delta y \cdot DC_{\text{ref}}(B_3) \end{aligned} \tag{2.14}$$

In the case if a macroblock of the P-frame is intra-coded, then the DC image will contain the DC values of the intra-coded blocks.

2.3 Conclusion

In this chapter, we explained the MPEG-1/2 video compression algorithm and the data we use in our work. The challenge of the extracted data from the MPEG-1/2 compressed video as described above is that we only have rough information for indexing. DC images are of very low resolution and thus degraded by aliasing. In addition, they suffer from blur due to the block averaging. Motion vectors are in general noisy and are sometimes even inaccurate. This can be due to the weakness of the MPEG encoder, if e.g. motion bypasses the search window of the block matching algorithm. Using this data for indexing allows fast computational times, but at the expense of image quality or motion accuracy. Hence, these characteristics have to be taken into account when processing this data and adapted robust algorithms need to be proposed.

We will show in Appendix B that we find similar situations (motion vectors and transform coefficients) in later MPEG standards as they are based on the MPEG-1/2 video coding scheme. There, we show that our method can be enhanced for later MPEG standards. For instance, with the advancements of the MPEG-1/2 video compression, motion estimation has been improved so that more reliable motion vectors can be extracted. In H.264/MPEG-4 AVC, transform coding operates on a reduced block size of 4×4. This allows the extraction of DC images at a higher resolution which are less aliased and less blurred.

Chapter 3

Construction of Mosaics from the Compressed Stream

Several application areas for mosaic representationse exist. Mosaic representations of video segments shot by the same camera are a part of the MPEG-4 video coding standard [118]. They allow an efficient encoding of the whole scene by composing it into the foreground objects and the background still mosaic. They are also defined as a descriptor of a shot in the MPEG-7 standard [119] for the purpose of video indexing. For example they can be used for a fast browsing of the video [72]. Further applications are video editing such as the removal of an object from the scene [129], the creation of virtual environments [184], and video post-production e.g. artificial modifications of the cameras point of view.

A mosaic is constructed by aligning and warping the frames of a shot upon each other in a single reference coordinate system. One single image results giving a panoramic view of the whole shot. Mosaicing methods proposed in literature perform on raw video, but today an enormous quantity of video content is already available in compressed form. Up to now the MPEG-2 standard [117] is the most frequent encoding for video available in archives or coming from broadcast channels. Applying raw video mosaicing methods to this content would mean its full decoding. Furthermore, aligning and warping frames in a mosaicing process requires a motion estimation which is a costly operation. Thus, motion has to be estimated after the full decoding, while it has been already available in the compressed stream.

Thus, in this chapter we focus on the construction of mosaics from MPEG-1/2 compressed video. Section 3.1 gives an overview of state-of-the-art mosaicing methods. In Section 3.2 we present our mosaicing method. Section 3.3 presents some results obtained with this method and Section 3.4 concludes this chapter.

3.1 State of the Art

Since the last decade significant research work has been done in video analysis for the mosaic construction from raw video or image sequences. Thus, in this section we will review main approaches in this domain.

Depending on the application different mosaic representations exist [73]:

- *Static mosaic* [106]: All frames of the sequence are aligned in a fixed coordinate system. This image provides an extended view of the entire static background in the scene, moving objects are not included.
- *Dynamic mosaic* [129]: A sequence of evolving mosaic images is created, where the content of each new mosaic is updated with the most current information from the most recent frame. The sequence of dynamic mosaics can be visualised either with a stationary background with a fixed coordinate system, or in a manner such that each new mosaic is aligned with the corresponding input video frame whereas the mosaic is viewed within a moving coordinate system.
- *Synopsis mosaic* [72]: The aim of this mosaic type is to represent the dynamic events in the scene. In order to provide a summary of the events in the scene, the background mosaic is overlayed with the trajectories of moving objects.
- *Temporal pyramid* [71]: The pyramid is a tree structure whose leaves are the original images and the interior nodes represent mosaics in form of residuals that merge together the visual information of their children. The root consists of a static mosaic of the sequence and the succeeding levels represent residuals estimated over various time scales. Reconstruction of the mosaic at a certain time is achieved by hierarchically combining the static mosaic with the residual mosaics.
- *Multi-resolution mosaic* [98]: Varying image resolutions can be handled by a multi-resolution data structure, which captures information from each new frame at its closest corresponding resolution level in a mosaic pyramid. When a frame is predicted/reconstructed from the mosaic pyramid, the highest existing resolution data in the mosaic which corresponds to the frame is projected onto the frame's resolution.

Given an image sequence, the mosaic construction consists in two basic steps:

- *Registration*: The images of the sequence are aligned into one global coordinate system.
- *Blending*: The aligned images are combined into one single image.

Some methods consider additionally:

- *Selection of frames* [128]: Only a subset of the image sequence may be chosen for the mosaic construction.
- *Reprojection* [21]: After registration, an additional transformation may be applied to the aligned images in order to render an image from a different view.

- *Moving object detection* [75]: Moving objects may be detected and excluded from the blending to avoid artefacts in the mosaic.
- *Illumination correction* [21]: In natural scenes the lighting variations can yield illumination changes in the image sequence. Then inconsistencies due to lighting variations, so called seams, can appear in the mosaic on image borders. In order to avoid this phenomenon an illumination correction is required. It consists in a harmonisation of the illumination conditions of all frames in the sequence.

In the following, we will review each of these steps.

3.1.1 Selection of Frames

The construction of a mosaic can be accomplished starting from an image sequence sorted in a temporal order. Such a sequence often contains a large number of frames. Because of computational costs caused by the alignment of each frame, it is sometimes beneficial just to use a subset of them. In order to reduce the computational costs and to optimise the mosaic quality images may be discarded that do not contain sufficiently new information to the mosaic or that are of low quality.

Nicolas [128] names the following section criteria:

- *Amplitude of motion*: If the motion amplitude with respect to the last selected frame is large enough, then the new frame contains some regions which were invisible in the preceding. This criterion allows to discard in particular frames with static camera motion. This requires the motion estimation for all frames in the sequence.
- *Quality of the image*: The quality of the frames may be different e.g. due to noise or contrast changes. Thus, the frames with a low quality are discarded.

  Within this context, Stern et al. [179] proposed a frame selection method based on the blur in each image. Frames blurred by low-frequency vibrations above a certain blur level are discarded from the computation. Therefore, they define the *effective blur* extend b as 3.46 times the standard deviation of the point spread function (PSF) (the function which describes the blur):

$$b_k = 3.46 \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[(x - t_{k,x})^2 + (y - t_{k,y})^2 \right] \mathrm{B}_k(x, y) \, \mathrm{d}x \, \mathrm{d}y \right]^{1/2} \tag{3.1}$$

$$\tag{3.2}$$

  where x, y are the coordinates, $\mathrm{B}_k(x, y)$ is the PSF of the kth image and $t_{k,x}, t_{k,y}$ are translational components of the kth image. In the above equation, it is assumed that the motion PSF has nonzero values only in a line on the direction of the motion. Then, the blur diversity ratio $\bar{b}_K$ in the sequence of K images is defined as:

$$\bar{b}_K = \frac{b_K}{1/(K-1) \sum_{k=1}^{K-1} b_k} \tag{3.3}$$

where b_K is the effective blur of the most severely blurred image and $1/(K-1)\sum_{k=1}^{K-1} b_k$ is the mean effective blur of the other images in the set. If $\bar{b}_K$ is higher than 1.40, then the most blurred image is discarded. If $\bar{b}_K$ is lower than 1.40, which means that the effective blur of the most images is no larger than 40% of the mean of the others, the whole set of images is used. This work was extended by Stern et al. [180] in order to discard additionally frames that are misregistered or for which the blur was not estimated precisely enough.

- *Complete or partial reconstruction*: This constraint allows to define the percentage of points in the mosaic that have to be defined with respect to the set of points defined in the entire image sequence. A complete reconstruction corresponds to 100%.

- *Respective contribution of the different frames:* The selection of frames can be realised by evaluating respectively the contribution of each frame to the mosaic construction. Principally, two methods are possible, both relying on the alignment of the frames.

  The first method evaluates recursively the contribution of each frame. For every new frame which has not yet been used the number of undefined pixels in the mosaic is computed. The frame with the highest contribution i.e. the highest number of undefined pixels in the mosaic is retained and blended into the mosaic. The process is iterated until e.g. a certain percentage of the mosaic has been reconstructed, or the maximal number of frames is achieved.

  The second method evaluates a-priori the contribution of the frames. This allows to better optimise the selection of frames using the effective contribution:

$$\varpi = \sum_{\mathbf{p}\in\mathcal{M}} \frac{\bar{I}(\mathbf{p},k)}{|\mathbf{p}|} \tag{3.4}$$

  with

$$\bar{I}(\mathbf{p},k) = \begin{cases} 1 & \text{if } I(\mathbf{p},k) \text{ is defined at } \mathbf{p} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

  where $|\mathbf{p}|$ represent the number of frames defined at the pixel $\mathbf{p}$, and $\mathcal{M}$ is the set of pixels in the mosaic which are not yet defined.

3.1.2 Registration

The objective of registration is to align every image into a global coordinate system which represents the whole scene. The image alignment depends on the chosen world coordinate system and the motion model describing the camera motion.

The choice of global coordinate system is usually that of one of the input images, called the *reference image*, then all other images are aligned to that frame. This is illustrated in Figure 3.1. If a virtual coordinate system is chosen, all images of the sequence are aligned

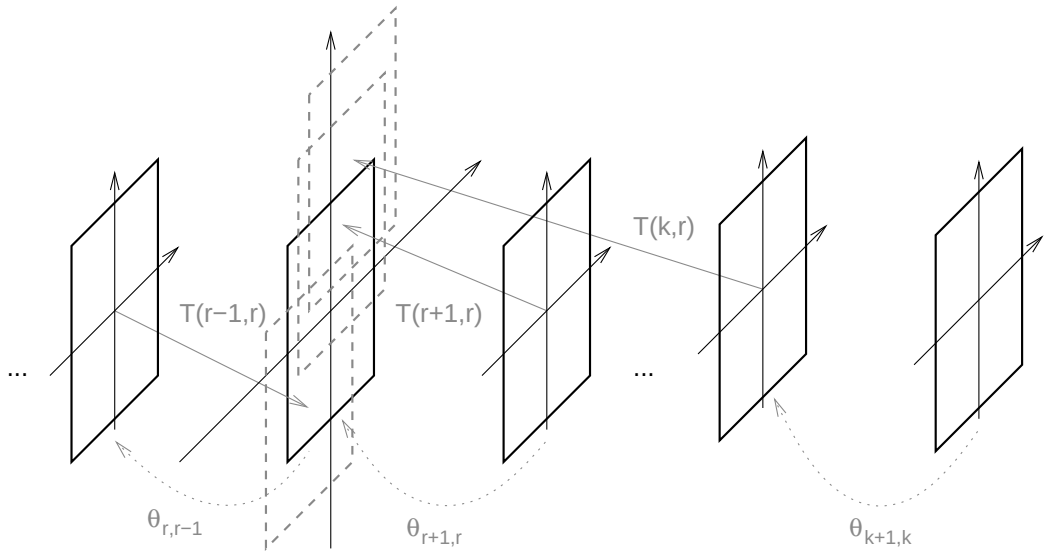


Figure 3.1: Registration of the image sequence with respect to a reference image.

with respect to a reference frame and an additional transformation between the reference frame and the virtual coordinate system has to be defined (see Section 3.1.3).

There are several ways to compute the transformations for the images of the sequence to the reference coordinate system:

- Frame to frame [157]: The transformations are first computed between successive frames for the entire sequence. This is illustrated in Figure 3.1 where r denotes the index of the reference frame. Then, the transformation between a particular image and the reference image $\mathrm{T}(k, r)$ is obtained by concatenating the intermediate transformations. In case of backward intermediate transformations $\theta_{k,k-1}$ the concatenation for the kth image can be expressed as:

$$\mathrm{T}(k,r) = \begin{cases} \theta_{k,k-1} \circ \theta_{k-1,k-2} \circ \cdots \theta_{r+2,r+1} \circ \theta_{r+1,r} & \text{if } k > r \\ \theta^{-1}_{k+1,k} \circ \theta^{-1}_{k+2,k+1} \circ \cdots \theta^{-1}_{r-1,r-2} \circ \theta^{-1}_{r,r-1} & \text{if } k < r \end{cases} \tag{3.6}$$

  where $\circ$ is the concatenation operator.

  The problem here is that alignment errors are accumulated during the repeated composition of transformations. Thus, the alignment can be further refined by directly refining the composed transformations between each image and the mosaic [129, 95].

- Frame to mosaic [155]: The transformation from one frame to the reference frame is computed directly. To handle the problem of large displacements between the mosaic image and the current frame, the transformation computed between previous frame and the mosaic image can be used as an initial estimate.

- Mosaic to frame [73] : In some dynamic applications, it is important to maintain the images in their input coordinate systems. In this case, it is more useful to align the mosaic to the current frame.

- Global alignment [173, 158, 85]: All images of sequence are simultaneously registered minimising a global error criterion.

- Topology determination [158]: During registration a graph is constructed describing the neighbourhood relationships in the image sequence. Based on this graph local alignments between neighbouring frames are refined and global consistency is optimised.

Different types of motion models exist to describe these geometrical transformations. The alignment can be limited to 2D motion models [140], or can utilise more complex 3D motion models [95, 157, 184, 185], layered representations [156, 72], or optical flow [142, 149]. Their computational complexity influences directly the quality of the obtained mosaic image. For example the translational model, supposing only translational motion in the scene which is easy and fast to compute, is too restrictive and would impact the quality of the mosaic. The most common 2D motion models are the affine model [129], the homography [22, 21] and quadratic models [73].

In the case of low-quality cameras using cheap, wide-angle lenses the acquired images may be radially distorted at the periphery. This can be corrected by including a parametric model of the lens distortion in the registration [155].

3.1.3 Reprojection

After registration each point of each image can be transformed to a point in the reference coordinate system. In order to actually render an image from the reference coordinate system an additional transformation T_+ may be applied which maps points from the reference coordinate system to points in the rendered image. T_+ is referred to as the *rendering transformation*. In principle T_+ could be any one-to-one mapping, but in practice it is usually determined by the choice of *reprojection manifold* e.g. a plane or a cylinder. Points in the reference coordinate system are projected onto the manifold, and then projected in the global frame.

The simplest and most commonly used manifold is a *plane* onto which all of the images are reprojected. In this case the rendering transform is a homography which reduces in the trivial case to an identity transform (in this case we do not speak of reprojection). Using a homography as rendering transform, straight lines are preserved. Using a planar manifold and aligning all frames to a single reference frame is reasonable only when there are no considerable depth differences in the scene and the camera motion is mainly a sideways translation and rotation around the optical axis [142]. Significant distortions are created by more general camera motions e.g. sideways rotation or when there are scale changes in the image due to camera translation.

A *cylindrical manifold* was used by Szeliski [185], Mann and Picard [106], and Jaillon and Montanvert [77]. This manifold, concentric with the camera, is suitable for image sequences with a rotating camera which sweeps a very large angle. The method is best suited to image sequences in which the camera rotates around a single axis, in which the images are tangent

planes to the manifold. The rendered image does not suffer from the same projective distortion as the planar manifold. Instead, straight lines in the world are mapped to sinusoids. The rendering transformation maps cylindrical polar coordinates to rectangular image coordinates as follows. A point (X, Y, Z) in the camera centered coordinate system maps to a point $(\vartheta, \nu), \vartheta =]-\pi, \pi]$ on the manifold as [185]:

$$\vartheta = \tan^{-1}\left(\frac{X}{Z}\right) \tag{3.7}$$

$$\nu = \frac{Y}{\sqrt{X^2 + Z^2}} \tag{3.8}$$

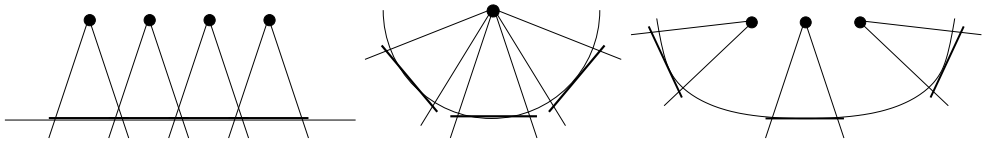
When the camera motion is pure rotation, a homogeneous 2D point $\mathbf{p} = (x, y, 1)$ in the kth image projects to a ray $\mathbf{P} = (X, Y, Z)$ in camera coordinates as [21]:

$$\mathbf{P} = \mathbf{R}_k^{-1}\mathbf{K}_k^{-1}\mathbf{p} \tag{3.9}$$

where $\mathbf{K}_k$ is the calibration matrix for the kth camera, $\mathbf{R}_k$ is the rotation of the kth camera relative to the reference view. These equations are combined to obtain the mapping between image points (x, y) and points on the manifold (ϑ, ν). Evidently, this method requires an estimate of the camera calibration matrices $\mathbf{K}_k$. A method for estimating the internal parameters of a rotating camera can be found in [65].

Another reprojection manifold is the *spherical manifold*. In a similar scheme to the cylindrical manifold, points in reference coordinate system may be parametrised in terms of spherical polar coordinates. This kind of manifold was used by Szeliski and Shum [186].

Peleg et al. [140, 142] proposed the use of *adaptive manifolds*. To avoid distortions in the mosaic they define a manifold projection which depends on the camera motion in the scene. The rendered image is locally similar to the input images without the need for calibration. Figure 3.2 illustrates some examples of adaptive manifolds in case of a pure translation of the camera, a pure rotation of the camera and a combination of translation and rotation of the camera.



(a) (b) (c)

Figure 3.2: Examples of adaptive manifolds [139]: a) a pure translation of the camera, b) a pure rotation of the camera, and c) a combination of translation and rotation of the camera. The projection is onto a smooth manifold passing through the centers of the image planes. The camera (black point) is located at the top of the field-of-view cone, and the image plane is marked by a bold segment.

3.1.4 Moving Object Detection

If moving objects are not taken into account, ghost artefacts may appear as they are combined with the background during blending. A common approach is to detect moving objects and exclude their pixels from the blending using outlier rejection schemes. An *outlier* is defined as a region that has been occluded, an object that suddenly appears in one of the images, or a region that undergoes unexpected motion.

The method of Eekeren et al. [198] detects moving objects by comparing a warped estimate of the background, $\hat{I}_{bg}(\mathbf{p})$ with the current frame $I(\mathbf{p})$. To construct an initial background image, the authors assume that the first two frames of the sequence do not contain moving objects. Then, the warped background estimate is subtracted from the current frame and an residual image results:

$$R_{bg}(\mathbf{p}) = I(\mathbf{p}) - \hat{I}_{bg}(\mathbf{p}) \tag{3.10}$$

The absolute value of each residual is compared with a threshold λ and is marked in the mask O as moving object or as background:

$$O(\mathbf{p}) = \begin{cases} 1 & \text{if } |R_{bg}(\mathbf{p})| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where 1 denotes a moving object and 0 the background.

The assumption that the first two frames do not contain moving object is restrictive and it is generally not often hold for broadcast video. In addition, outlier rejection schemes based on the pixel differences are known as not very robust as they strongly depend on the accuracy of registration and the noise in the image. Therefore, other methods have been proposed based on motion information.

The method of Irani and Peleg [76, 75] segments moving objects by first computing the dominant motion between two images. Then, the image region having this motion has to be determined. After registration the motion of the corresponding region is canceled and the tracked region is stationary in the registered image. The segmentation problem reduces to identifying the stationary region in the registered images.

An effective way of determining semantically significant residuals is to consider not only the residual intensity but also the magnitude of local residual motions between the two aligned frames. Thus, pixels are classified as moving or stationary using a local analysis based on the weighted average of the normal flow magnitudes over a small neighbourhood $\mathcal{N}(\mathbf{p})$ of each pixel $\mathbf{p}$ (typically a 3×3 neighbourhood). (The normal flow is the component of the optical flow in the direction of the spatial gradient.) The weights are taken to be $\|\nabla I(\mathbf{p}, k)\|^2$ [76]:

$$S(\mathbf{p}, k) = \frac{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} |I(\mathbf{p}_i, k) - I(\mathbf{p}_i, k-1)| \cdot \|\nabla I(\mathbf{p}, k)\|}{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \|\nabla I(\mathbf{p}_i, k)\|^2 + C} \tag{3.12}$$

where $I(k)$ and $I(k-1)$ are the two registered frames and $\nabla I(k)$ the spatial gradient of $I(k)$. The constant C is used to avoid numerical instabilities and to suppress noise.

The reliability of the measure S at each pixel is determined by the numerical stability of the two well-known optical flow equations [76]:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{pmatrix} \tag{3.13}$$

where for each pixel $\mathbf{p}$ the sum is taken over the neighbourhood $\mathcal{N}(\mathbf{p})$. I_x, I_y are the spatial derivatives in x and y-direction and I_t is the temporal derivative of the image I.

The reliability $R(\mathbf{p}, k)$ is then expressed by the inverse of the condition number of the matrix in (3.13):

$$R(\mathbf{p}, k) = \frac{\lambda_{min}}{\lambda_{max}} \tag{3.14}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues, respectively.

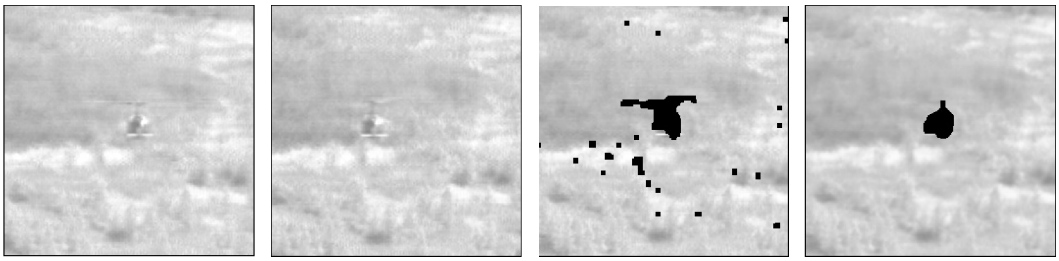
In order to classify correctly large regions having uniform intensity, a multi-resolution scheme is used as in low-resolution levels the uniform regions are small. First, all the pixels at the lowest resolution level are initialised as unknown to be moving or stationary. Then, for each pixel at each resolution level S and R are computed. If S is high (i.e. the pixel is moving) or if it is low with high R (i.e. the pixel is stationary) then value S of the pixel at that resolution level is set to the new value of S. Otherwise, if the local information available at the current resolution level is not sufficient for classification, then the value S of the previous lower resolution level is maintained.

This algorithm yields a continuous function, which is an indication of the magnitude of the displacement of each pixel between the two images. Taking a threshold on this function yields the partitioning of image into moving and stationary regions.

Then, the segmented objects are tracked along the image sequence. This is done by temporal integration where for each object a dynamic internal representation image is constructed. This image is constructed by taking a weighted average of recent frames, registered with respect to the tracked motion. This image contains after a few frames a sharp image of the tracked objects, and a blurred image of all other objects. Each new frame in the sequence is compared to the internal representation image of the tracked object. Figure 3.3 shows an example of the object detection and tracking method of Irani and Peleg.

Kumar et al. [95] propose an overall algorithm for moving object detection and registration. Given two views of a general 3D scene and for arbitrary 3D motion, first a parametric 2D motion model is computed and then a 3D parallax motion field to align the two images. If the majority of the image is aligned, the algorithm stops and unaligned areas are labelled as moving objects, otherwise the motion parameters are refined. Unaligned areas are labelled as moving objects. The test for alignment is accomplished by thresholding the magnitude of the normal flow.

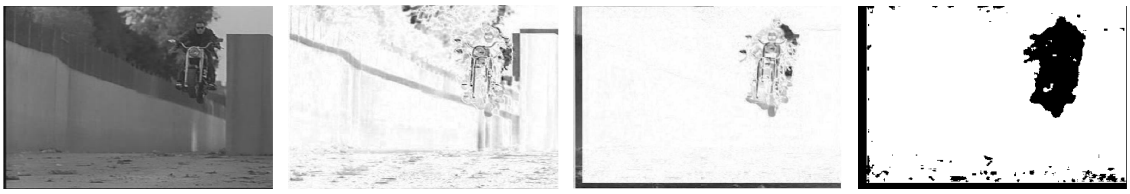
The method of Odobez and Bouthemy [130] takes advantage of a multiresolution framework and an incremental scheme. It minimises a M-estimator criterion to ensure the goal of robustness to outliers formed by the points corresponding to secondary motions or to areas where the classical image motion equation used is not valid. The outlier rejection scheme is based on the Tukey's biweight function. The authors consider a six-parameter affine motion model.



(a) (b) (c) (d)

Figure 3.3: Detection and tracking the dominant object using temporal integration [75]: (a) and (b) two frames of the sequence where both the background and the helicopter are moving, (c) the segmented object (background) using affine dominant motion computed between the first two frames, (c) the segmented object after a few frames using temporal integration.

Sawhney and Ayer [156] extend the approach [130]. The algorithm employs an automatic computation of the scale parameter that is crucial in rejecting the non-dominant components as outliers, in contrast to predefined schedules for scale used by [130]. Additionally to 2D affine and plane projective motion models for describing image motion using direct methods, they employ a true 3D model of motion and scene structure with uncalibrated cameras. Here, the M-estimator is based on the sum of squares function and the German-McLure function. An example of the outliers rejection method of Sawhney and Ayer is shown in Figure 3.4.



(a) (b) (c) (d)

Figure 3.4: Example of outlier rejection [156]: (a) One frame, (b) the difference without motion compensation, (c) the difference after compensating the dominant motion, (d) the outlier mask. (Low differences are shown in white and high difference in black).

Hasler and Süsstrunk [66] propose a statistical outlier modeling scheme. First, the inliers are modelled by a Laplacien distribution of the residuals (the registration error) in order to discriminate outliers from inliers. Then, the two models are combined in an OutlierMix model enabling to find the proportion of outliers and to compute for each pixel a probability to belong to the inliers. Later, this method has been adopted by Zibetti and Mayer [205].

A different approach is that one of Shum and Szeliski [173]. Instead of detecting moving object in the scene, they perform local alignment using optical flow after the global alignment.

However, the presented approaches work on raw video, but as we are working in the

compressed domain. Thus, outlier rejection schemes based on motion information will not give satisfying results for MPEG motion vectors as they are very noisy. Therefore, we use a moving object detection method based on the work of Manerba [104] which we will present in Section 3.2.3. This approach is based on an outlier rejection of P-frame motion vectors and a color segmentation of I-frames that are merged together to extract moving objects in I-frames.

3.1.5 Illumination Correction

Often illumination changes can be observed between the images of the same scene. Although the registration is accurate, seams appear in the mosaic due to photometric variations across image boundaries. Traditional mosaicing methods tempt to eliminate such seams during blending e.g. blending the different images using a Laplacien pyramid [17, 77]. However, the original factors causing that boundary seams were not considered.

Recent methods focus on the correction of illumination conditions by modeling the illumination changes. Pinel [144] and Nicolas [128] use a linear model of the luminance component to correct illumination changes for each pair of images. Only the pixels at the contours of the intersection of the two images are considered for the computation of the parameters of the linear model as shown in Figure 3.5.

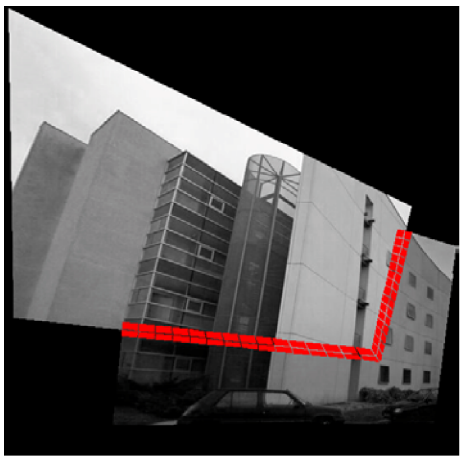


Figure 3.5: Region of the computation of the linear model for illumination correction [144].

The method of Capel [21, 24], referred to as *photometric registration*, considers two sources of photometric difference: (i) automatic camera adjustments such as automatic gain control, white balance or exposure, and (ii) illumination changes due to variation in the ambient light level or daylight level, or due to relative motion between the scene and the light source. The model treats each of the R, G and B colour channels independently. Within each channel, the variation between two images is modelled as a linear transformation, having two parameters: a multiplicative term ι and an additive term κ. The transformation can be written in

vector from as [21]:

$$\begin{pmatrix} R_2 \\ G_2 \\ B_2 \end{pmatrix} = \begin{bmatrix} \iota_R & 0 & 0 \\ 0 & \iota_G & 0 \\ 0 & 0 & \iota_B \end{bmatrix} \begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix} + \begin{pmatrix} \kappa_R \\ \kappa_G \\ \kappa_B \end{pmatrix} \tag{3.15}$$

where $(R_1, G_1, B_1)^{\mathrm{T}}$ is the pixel in the reference image and $(R_2, G_2, B_2)^{\mathrm{T}}$ the pixel in the image to be corrected.

After geometric alignment, the colours of the corresponding pixels in the two images may be used to directly estimate the parameters of the colour transformation between them. Treating each channel separately, the estimation procedure for ι and κ is clearly a simple line-fit to the intensities of corresponding pixels (i_1, i_2). However, at many pixels, the difference in intensities, $i_1 - i_2$, cannot be explained by the linear photometric model. The model-outliers arise from various sources: saturation at the low or high end of the pixel intensity range, specularities or occlusions, shadows. Thus, its is important that the line-fitting algorithm is robust to these outliers. To this end, Capel proposes in [21] to use the MSAC (M-estimator SAmple Consensus) algorithm [192] which is a variation of the RANSAC algorithm. Instead of counting the number of inliers, as RANSAC does, MSAC determines the likelihood of the data given the proposed model parameters according to a simple robust M-estimator. For each point the $(i_1, i_2)_n$, the distance d_n to the proposed line (ι, κ) is computed, and the overall cost associated with the solution is given by [21]:

$$C = \sum_n \rho(d_n^2) \tag{3.16}$$

where

$$\rho(d^2) = \begin{cases} d^2 & \text{if } d^2 < \lambda_d \\ \lambda_d & \text{otherwise} \end{cases} \tag{3.17}$$

Capel chose the threshold λ_d as 1.96σ so that Gaussian inliers are only incorrectly classified as outliers 5%, assuming that σ is around 5 "grey-levels". In the final step, the parameters are refined by performing orthogonal regression on the MSAC inliers.

Hasler and Süsstrunk [67] propose a method for colour correction between images that were not captured under controlled conditions i.e. with varying white-point and exposure, and unknown colour rendering algorithms. They assume that the camera performs for each individual image an exposure and white balancing analysis, followed by a tone mapping through the Opto-Electronic Conversion Function (OECF) that is constant for the image sequence.

The work of Kim and Hong [85] is close that one of Hasler and Süsstrunk [67]. Instead of using only pairwise local relations as in [85], Kim and Hong propose a global approach using multiple images. The camera parameters and scene radiances are estimated simultaneously in a single framework. In order to estimate parameters robust to outliers such as misaligned pixels or moving objects in the overlapping region, they used a Huber-type skipped mean estimator [60].

Zhao [212] proposes a method to correct significant and complicated illumination changes. He shows that existing robust estimation techniques that treat pixel intensity variation due to illumination changes as outliers do not offer a good solution in case of significant illumination changes. Therefore, a shape-from-shading framework is proposed based on an illumination model.

3.1.6 Blending

The final stage in the mosaic construction is to combine the images in their overlapping regions. A simple approach is a *temporal averaging* of the intensity values of the aligned images mentioned by Nicolas [129]. Then, the mosaic M is computed as:

$$M(\mathbf{p}) = \mu(\mathbf{p}, K) \sum_{k=1}^{K} \hat{I}(\mathbf{p}, k) \tag{3.18}$$

where $\hat{I}(\mathbf{p}, k)$ are the aligned images and $\mu(\mathbf{p}, K) = \frac{1}{|\mathbf{p}|}$ with $|\mathbf{p}|$ as the number of available pixels at position $\mathbf{p}$. This method is working well if the alignment is accurate and no moving objects are present in the scene, otherwise burring and ghost artefacts may appear.

Instead, *temporal median filtering* can be used which tends to eliminate moving objects which could appear blurred in the mosaic. Dupuy and Benois-Pineau [37] proposed a 1D mosaicing approach, where the median filtering allowed for the removal of object. In the 2D image domain the pixel values of the mosaic are given by:

$$M(\mathbf{p}) = \text{median}_{k=1,\ldots,K}(\hat{I}(\mathbf{p}, k)) \tag{3.19}$$

The use of median filtering is justified as the moving objects crossing a pixel represent outliers in the sequence of background pixels. As the median filter is robust up to 50% outliers, this is a simple and direct method of object removal. A drawback of temporal median filtering is that it is computationally very expensive, requiring a sort operation for every pixel in the mosaic image, which can become very expensive when large numbers of overlapping images are involved.

Therefore, moving objects can be directly excluded from the blending by using *outlier rejection maps*. Denoting $O_{\text{b}}(k)$ as the outlier rejection maps, the blending can be done by a temporal averaging as:

$$M(\mathbf{p}) = \mu(\mathbf{p}, K) \sum_{k=1}^{K} O_{\text{b}}(k) \hat{I}(\mathbf{p}, k) \tag{3.20}$$

with

$$O_{\text{b}}(k) = \begin{cases} 1 & \text{if } \mathbf{p} \text{ belongs to the background} \\ 0 & \text{if } \mathbf{p} \text{ belongs to a moving object} \end{cases} \tag{3.21}$$

In fact, a pixel is only blended into the mosaic if $O_{\text{b}}(k) = 1$.

Additionally, weighting functions may be associated with the input images. Capel [21] mentions *feathered blending* where the weighting function decreases with the distance of a pixel from the image center aiming at ignoring alignment inaccuracies near image boundaries. An example weighting function is a bi-quadratic function [21]:

$$\mathrm{f}(x,y) = \left(1 - \left(x - \frac{w}{2}\right)\right)\left(1 - \left(y - \frac{h}{2}\right)\right) \tag{3.22}$$

where w and h are, respectively, the width and height of the image. Further weighting functions have been suggested by Nicolas [129] such as a weighting function that minimises the temporal delay or that maximises the mosaic resolution.

A different approach is the *cut and paste* method of Peleg et al. [140, 142]. In order to avoid blurring due to inaccurate alignment, only one of the input images is selected to represent a region in the mosaic. Pixel values in the mosaic are taken from a single image whose center, after alignment, is closest to the corresponding pixel as illustrated in Figure 3.6. The reasons are that the alignment is usually better at the center than at the image borders and the image distortion is minimal at the center of the images.

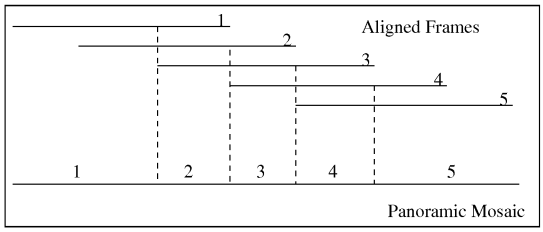


Figure 3.6: Cut and paste for mosaic blending [139].

Recently, super-resolution techniques have been used for mosaic blending by Mann and Picard [106], Zomet and Peleg [215], Capel and Zisserman [22], and Altunbasak and Patti[4]. We will focus on super-resolution techniques in the Chapter 3.

3.2 Construction of Mosaics from MPEG-1/2 Compressed Streams

Here, we present our approach for the construction of mosaics from MPEG-1/2 compressed streams. In this work we address the problem in its totality. Taking compressed video as input, the system has to produce the mosaic whatever is the complexity of the scene. Hence, the overall method has to contain the following steps illustrated in Figure 3.7:

1. *Data extraction:* DC images and motion compensation vectors of P-frames are extracted from the compressed stream. As we stated in Chapter 2, DC images are a source of low-resolution signal (intra frame for I-frames or inter frame for P-frames). Macroblock motion vectors of P-frames convey motion information. Thus, data extraction consists in partial decoding of the stream to extract this data.

2. *Registration:* The motion compensation vectors are used in a motion estimation scheme in order to register the DC images of I-frames in a reference coordinate system.

3. *Moving object detection:* Based on the registration moving objects are detected in the sequence of DC images of I-frames. This allows the removal of moving objects in order to create a static background mosaic. Some objects can be reinserted in the mosaic afterwards.

4. *Illumination correction:* The sequence of DC images of I-frames is illumination corrected to reduce seams in the mosaic.

5. *Blending:* First, the registered and illumination corrected DC images of I-frames are blended into a global frame excluding the detected moving objects to construct the static background mosaic. Second, some objects are inserted in the mosaic.

6. *Post-processing:* In a last step, post-processing is applied to the mosaic in order to fill holes due to the exclusion of objects in the blending, to remove eventual artefacts and to obtain a more realistic object insertion.

In the following we will explain each of these steps.

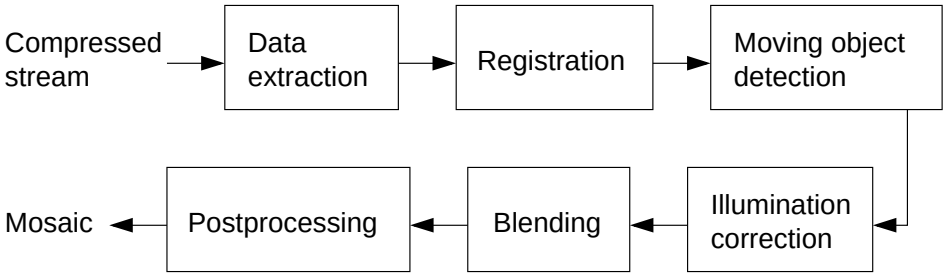


Figure 3.7: Global scheme of the mosaic construction.

3.2.1 Data Extraction

The extraction of data from MPEG-1/2 compressed video is the first step in Figure 3.7. The mosaic construction is mainly based on the use of DC images of I-frames as input sequence for the mosaic construction. DC images of I-frames serve as input sequence for the mosaic construction. P-frame motion vectors are used for the estimation of global motion. Furthermore, we use DC images of the encoded P-frame motion compensation error in a confidence measure for the motion models estimated from P-frame motion. In the case of low-quality MPEG motion vectors, we can decide based on this confidence measure, if a motion model estimated from these vectors has to be corrected or not. As an alternative solution to the correction, the motion models can also be reestimated using DC images of P-frames.

The extraction of (P-frames motion vectors, DC images of I-frames, DC images of the motion compensation error of P-frames) is realised by the same MSSG decoder [121]. The computation of DC images of P-frames and the data extraction were described in Section 2.2.

3.2.2 Registration

In order to register the kth frame of the video sequence in the coordinate system of the mosaic, a geometric transformation $\mathrm{T}(k,r)$ has to be defined from the kth frame of the video sequence to the reference frame of the mosaic which is the rth frame in the sequence. We propose computing this transformation using motion information from the compressed stream. As we use only I-frames for the mosaic construction, the geometric transformation $\mathrm{T}(k,r)$ is the concatenation of pair-wise frame-to-frame global motion models $\theta_{k,k-1}, \theta_{k-1,k-2}, \ldots, \theta_{r+1,r}$ in backward direction or $\theta^{-1}_{k,k+1}, \theta^{-1}_{k+1,k+2}, \ldots, \theta^{-1}_{r-1,r}$ in forward direction. Taking into account the roughness of motion vectors in the compressed stream and their sparseness along the time, the motion estimation scheme we propose comprises three steps:

1. The estimation of backward global affine motion models for pairs of time-successive frames where motion vectors are available in the compressed stream. These pairs are I/P and P/P. For the sake of computational efficiency we use only primary motion vectors of P-frames while B-frame motion vectors are skipped.

2. The interpolation of motion models, when motion vectors are not available i.e. for P/I-pairs.

3. The correction or reestimation of low-quality motion models estimated from inaccurate compressed motion vectors.

We will now explain the estimation of the global motion models using P-frame motion vectors. To this end, we use the method proposed by Durik and Benois-Pineau [38].

Robust Motion Estimator for P-Frames

We consider two main types of motion in a scene: the motion of the objects and the global motion of the scene content. The latter is principally due to the motion of the camera or to the change of focus. Here, we are interested in global camera motion in order to define the geometric transformations for the mosaic construction.

We use the six-parameter affine model to describe global camera motion. According to [72], the approximation of instantaneous image motion of a general 3D scene can be approximated by an 2D parametric motion model under the following conditions associated with the scene geometry and/or camera motion: (i) the scene is planar, (ii) when the 3D scene is sufficiently distant from the camera, or when the deviations from a planar scene surface are small relative to the overall distance of the scene from the camera, (iii) the camera undergoes a pure rotational motion or when the camera translation is negligible, (iv) the camera zooms in or out. Nevertheless, in [38, 187] the six-parameter affine model is proved to be sufficiently rich to characterise motion observed in an image plane of video. According to this model, θ, the elementary motion compensation vector $\mathbf{d} = (d_x, d_y)^T$ at the pixel position

$\mathbf{p} = (x, y)^T$ is expressed as:

$$\mathbf{d}(\mathbf{p}, \theta) = \begin{cases} d_x = a_1 + a_2(x - x_0) + a_3(y - y_0) \\ d_y = a_4 + a_5(x - x_0) + a_6(y - y_0) \end{cases} \tag{3.23}$$

where $a_1, \ldots, a_6$ are the global motion parameters of the camera, and $(x_0, y_0)^T$ denotes the reference point which is here the image center. In the case of MPEG motion prediction, $(x, y)^T$ corresponds to the center of a macroblock in the current image. Then, the motion compensation vector $\mathbf{d}$ points from the center of the macroblock to its position in the anchor frame.

The motion estimator proposed in [38] is based on a multi-resolution scheme which uses at each multi-resolution level a weighted least square estimation in order to minimise a robust functional of motion residuals (Tukey estimator [130]). The objective of the estimator is to minimise the residuals $\mathbf{r}$ between the the MPEG motion vectors $\mathbf{d}$ and their estimation $\hat{\mathbf{d}}$:

$$\mathbf{r}_i = \mathbf{d}_i - \hat{\mathbf{d}}_i \tag{3.24}$$

A robust estimation is identical with a good outlier rejection scheme. This is not possible by a hard limit rejection method. For this purpose the Tukey function is chosen. It is a cost function which provides the possibility to minimise the residuals and to limit the influence of the outliers continuously. The Tukey function ϱ is defined as:

$$\varrho(\mathbf{r}, \lambda_\varrho) = \begin{cases} \frac{\mathbf{r}^6}{6} - \frac{\lambda_\varrho^2 \mathbf{r}^4}{2} + \frac{\lambda_\varrho^4 \mathbf{r}^2}{2} & \text{if } \|\mathbf{r}\| < \lambda_\varrho \\ \frac{\lambda_\varrho^6}{6} & \text{otherwise} \end{cases} \tag{3.25}$$

with λ_ϱ as a threshold.

Then (3.25) is minimised by a classical weighted least square estimation. The influence of the Tukey function is approximated by weights which are computed from its derivative. Thus, the cost function can be written as:

$$\sum_i \varrho(\mathbf{r}_i) = \sum_n \frac{1}{2} w_i \mathbf{r}_i^2 \tag{3.26}$$

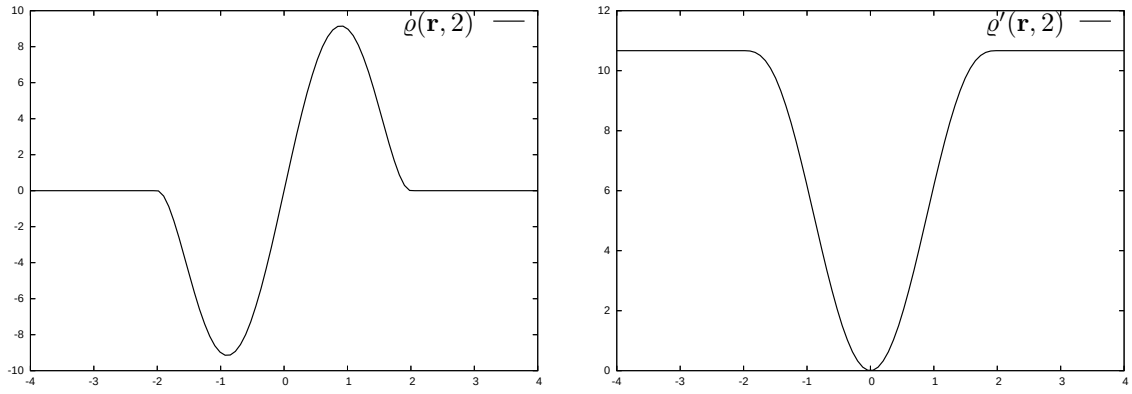
A necessary condition for the minimisation is that the derivatives of the error measure (3.26) with respect to each component a_j of θ are null:

$$\sum_i \varrho'(\mathbf{r}_i) \frac{\partial \mathbf{r}_i}{\partial a_j} = \sum_i w_i \mathbf{r}_i \frac{\partial \mathbf{r}_i}{\partial a_j} = 0, \quad 1 \leq j \leq 6 \tag{3.27}$$

The derivative of the Tukey function ϱ' is defined as follows:

$$\varrho'(\mathbf{r}, \lambda_\varrho) = \begin{cases} \mathbf{r}(\mathbf{r}^2 - \lambda_\varrho^2)^2 & \text{if } \|\mathbf{r}\| < \lambda_\varrho \\ 0 & \text{otherwise} \end{cases} \tag{3.28}$$

The Tukey function and its derivative are depicted in Figure 3.8.



(a) (b)

Figure 3.8: The Tukey estimator: (a) Tukey's biweight function ϱ and (b) its derivative ϱ' ($\lambda_\varrho = 2$).

Thus, the weights are given by:

$$w_i = \frac{\varrho'(\mathbf{r}_i)}{\mathbf{r}_i} \tag{3.29}$$

Based on a common formulation of the least square optimisation problem, the linear model (3.23) can be written in general matrix form:

$$\mathbf{Z} = \mathbf{H}\theta + \mathbf{V} \tag{3.30}$$

where $\mathbf{Z}$ are the measured motion compensation vectors, $\mathbf{H}$ is the observation matrix containing the macroblock centers, θ is the vector of the motion parameters and the vector $\mathbf{V}$ is the measurement noise. Then, the vector of the motion parameters can be estimated by the weighted least squares estimation as:

$$\hat{\theta} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{Z} \tag{3.31}$$

We denote N as the number of motion vectors. So, the matrices and vectors from (3.30) and (3.31) are constructed in the following way:

$\hat{\theta}$ is the 6×1 column vector of the estimated parameters from (3.23):

$$\hat{\theta} = (a_1, a_2, a_3, a_4, a_5, a_6)^T \tag{3.32}$$

$\mathbf{Z}$ is the $2N \times 1$ column vector of the measures, here this are the MPEG compensation vectors:

$$\mathbf{Z} = (d_{x,1}, \ldots, d_{x,N}, d_{y,1}, \ldots, d_{y,N})^T \tag{3.33}$$

$\mathbf{H}$ is the $2N \times 6$ observation matrix:

$$\mathbf{H} = \begin{pmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & x_N & y_N \end{pmatrix} . \tag{3.34}$$

$\mathbf{W}$ is the $2N \times 2N$ diagonal matrix of the weights:

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \cdots & \cdots & 0 \\ 0 & w_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & w_{2N-1} & 0 \\ 0 & \cdots & \cdots & 0 & w_{2N} \end{pmatrix} \tag{3.35}$$

where w_i are obtained by the derivative of the Tukey function (3.29).

The robust estimator incorporates three steps of outlier rejection. In the first step macroblocks on the image borders and intra-coded macroblocks are rejected explicitly. The MPEG motion estimation method does not provide good results for the macroblocks on the image borders. Thus, the border rows and columns are excluded from the estimation. In the case of intra-coded macroblocks the MPEG block matching algorithm did not find the matching macroblock in the anchor picture. They are marked as intra-coded in the bitstream and a zero motion compensation vector is encoded. It is important to reject them before the estimation since the zero value of the motion vector does not mean that there is no motion and thus tampers the estimated model.

The second step is an implicit outlier rejection in the multi-resolution process based on the Tukey function. Vectors with weights (3.29) equal zero are rejected directly. Then, for the remaining vectors the weights are recomputed and thresholded. If the weight of a vector is smaller than a predefined threshold, the motion vector is rejected as an outlier. Then, the remaining vectors are considered for the estimation 3.31. The weight expresses the accuracy of a MPEG motion vector according to the estimated model.

According to [38], the constant λ_ϱ in the Tukey function (3.25) is chosen as follows:

$$\lambda_\varrho^l = \begin{cases} \infty & \text{if } l = 1 \\ C_\varrho \cdot \sigma^l & \text{if } l = 2 \\ \frac{C_\varrho \sigma^l + \lambda_\varrho^{l-1}}{2} & \text{otherwise} \end{cases} \tag{3.36}$$

where σ^l is the standard deviation of the residuals at the multi-resolution level l, and C_ϱ is a constant in the interval $[2, 3]$. At the lowest multi-resolution level all weights are set to 1. This means that all vectors are accepted without outlier rejection for the first estimation of the global motion model. This corresponds to an infinite value of the constant λ_ϱ in (3.25).

In addition the motion vectors are separated into four groups. Two groups are created separating d_x and d_y. Each of them is subdivided in two groups regarding if the estimated vector is greater or smaller than the corresponding MPEG vector. This fine scheme allows the efficient filtering of MPEG motion vectors of strong magnitude occurring in flat areas, on occluding borders and in uncovered areas. For each group the σ value is different, that is why they are processed separately in the outlier rejection scheme. This implies the computation of four constants λ_ϱ, one for each group. Finally, a motion vector is rejected as an outlier if at least one of its coordinates is rejected.

The third step is an optional filtering of the outlier maps as they are very noisy after the multi-resolution. Based on the assumption that a non outlier macroblock with a majority of outliers in the neighborhood is likely to be an outlier too, a median filter is applied only to non outlier macroblocks.

Figure 3.9 shows an example of the robust estimator. The first row shows a P-frame (Figure 3.9(a)) and its anchor frame (Figure 3.9(b)) of the "Hiragasy" sequence (see Figure 3.27(c)). The extracted MPEG motion vectors are shown in Figure 3.9(c). We can notice that they are very noisy, mainly on macroblocks belonging to the objects and the image borders. Figure 3.9(d) shows the motion vectors obtained with the estimated motion model which describes well the motion of the camera between the two frames. The corresponding binary outlier rejection mask in Figure 3.9(e) was obtained by thresholding the weights w_i of the last multi-resolution level. Typically, we use a threshold equal 0.

Extrapolation of the Motion Parameters for I-Frames

In order to compute the geometric transformation from an I-Frame to the reference frame, we compute the motion models for the I and P-Frames between them. However, the motion model for an I-Frame can not be estimated with the method presented above since it is intra-coded. Hence, we propose to extrapolate the motion parameters from the previous P-Frames. Here, the previous P-Frames means the P-Frames up to the last I-Frame i.e. we consider the P-Frames in a GOP. The advantage of an extrapolation, on the contrary to an interpolation, is that we can take into account the possibility of a strong change in motion e.g. a shot boundary. In this case, the interpolation of a model for an I-frame would cause an erroneous smoothed motion as illustrated in Figure 3.10.

We extrapolate each motion parameter a_j, $1 \leq j \leq 6$, of the affine model (3.23) separately by a linear regression. The linear regression problem for the extrapolation of the motion parameters at I-frames is formulated as follows. Assuming that the time between successive I-frames is short enough to consider continuous and linear motion, we approximate a straight line for each set of motion parameters $a_j(q)$, $1 \leq q \leq Q$, in a GOP with Q as the number of P-frames in the GOP. Thus, we suppose that for each motion parameter a_j the following linear model is satisfied:

$$a_j(q) = \gamma_j + \eta_j \cdot q \tag{3.37}$$

with γ_j, η_j as the parameters of the straight line for the jth motion parameter. According to this model, the predicted motion parameter for the I-frame corresponds then to $a_j(Q+1)$

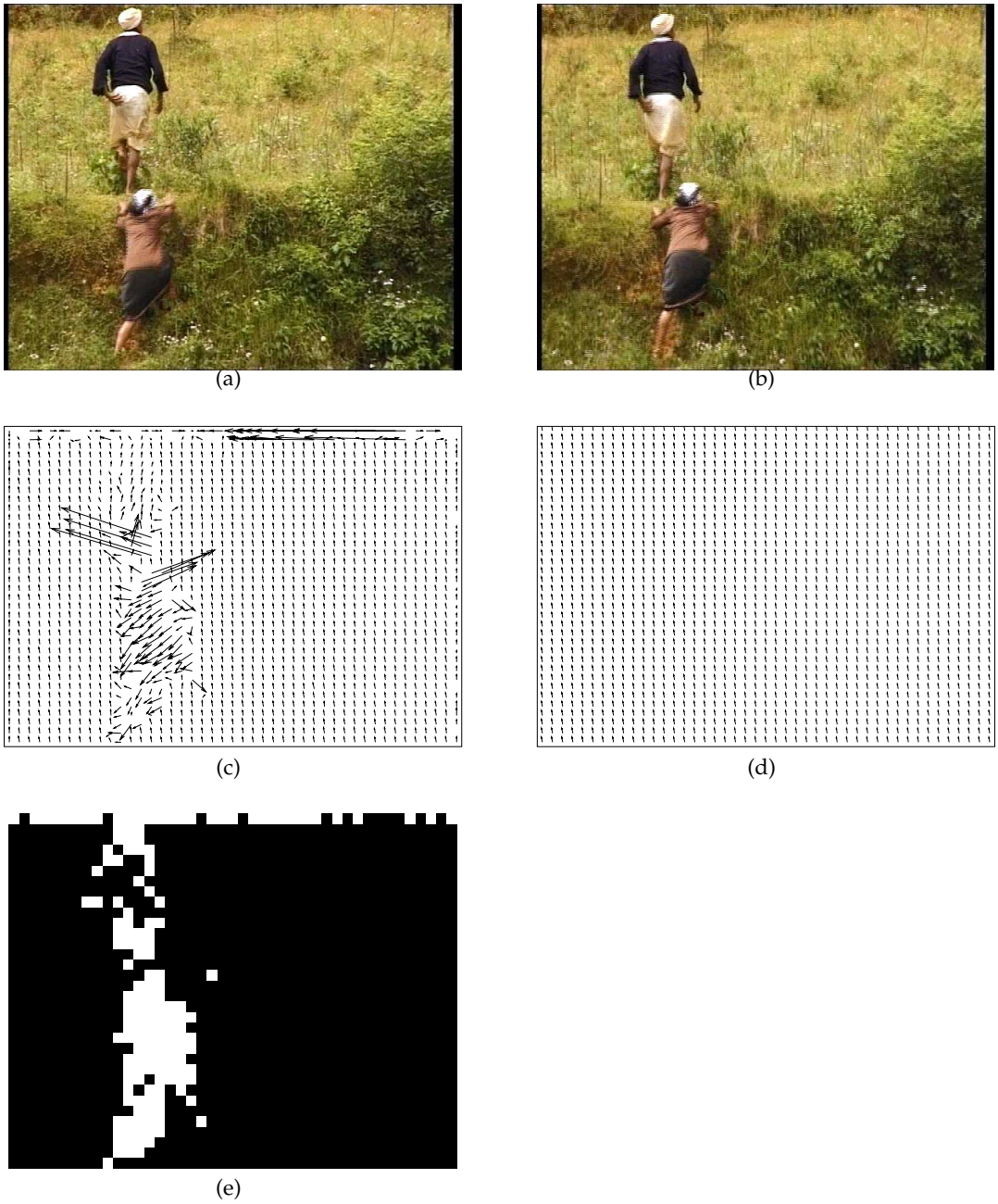


Figure 3.9: Global motion estimation for a P-frame of the "Hiragasy" sequence CERIMES-SFRS® : (b) The P-frame, (a) its reference frame, (c) the encoded MPEG motion vectors, (d) the motion vectors consistent to the estimated motion model, (e) the outlier map (outliers are marked in white).

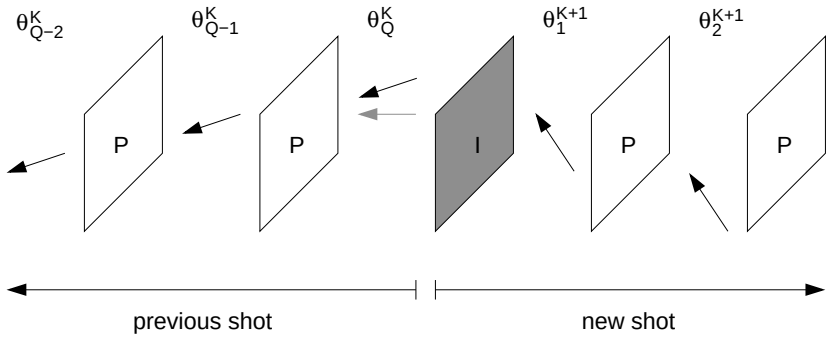


Figure 3.10: Extrapolation versus interpolation in the case of a strong motion change on an I-frame: the black arrow corresponds to the extrapolated motion model and the grey arrow corresponds to the interpolated motion model.

This assumption of a linear camera model is reasonable as it is restricted to only one GOP which is usually of 0.5 seconds duration. Furthermore, we assume that the camera motion along the whole sequence can be approximated by a piecewise linear model.

To estimate the parameter vector $\mathbf{l}_j = (\gamma_j, \eta_j)^T$, we use a weighted least square estimation similar to Equation (3.31):

$$\mathbf{l}_j = (\mathbf{H}_\mathrm{l}^\mathrm{T}\mathbf{W}_\mathrm{l}\mathbf{H}_\mathrm{l})^{-1}\mathbf{H}_\mathrm{l}^T\mathbf{W}_\mathrm{l}\mathbf{Z}_{\mathrm{l},j} \tag{3.38}$$

$\mathbf{Z}_{\mathrm{l},j}$ is the $Q \times 1$ column vector of the measures:

$$\mathbf{Z}_{\mathrm{l},j} = (a_j(1), \cdots, a_j(Q-1), a_j(Q))^T \tag{3.39}$$

$\mathbf{H}_\mathrm{l}$ is the $Q \times 2$ observation matrix according to the time line of the measures:

$$\mathbf{H}_\mathrm{l} = \begin{pmatrix} 1 & 1 \\ \vdots & \vdots \\ 1 & Q-1 \\ 1 & Q \end{pmatrix} \tag{3.40}$$

$\mathbf{W}_\mathrm{l}$ is the $Q \times Q$ diagonal matrix of the weights:

$$\mathbf{W}_\mathrm{l} = \begin{pmatrix} w_{\mathrm{l},1} & 0 & \cdots & \cdots & 0 \\ 0 & w_{\mathrm{l},2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & w_{\mathrm{l},Q-1} & 0 \\ 0 & \cdots & \cdots & 0 & w_{\mathrm{l},Q} \end{pmatrix} \tag{3.41}$$

The objective of the weights $w_{\mathrm{l},q}$ is to counterbalance inaccurate MPEG encoder motion estimations. Therefore, we use the encoded motion compensation error of P-frames. In fact, if the motion estimation of the MPEG encoder was inaccurate the motion compensation error

is strong, if it was accurate this error will be weak. This behaviour can be observed by calculating the mean low frequency energy of the motion compensation error. Thus, we use the DC image of the motion compensation error for the mean low frequency energy computation.

Let us denote $DC_{\mathrm{err}}(q)$ the DC image of the motion compensation error for the P-frame at time q. For its computation we consider only inter macroblocks that are consistent with the estimated motion model, so called motion *inliers*, otherwise the macroblocks are likely to belong to a moving object. Thus, the mean low frequency energy E is:

$$\mathrm{E}(q) = \frac{1}{|\mathcal{D}(q)|} \sum_{\mathbf{p} \in \mathcal{D}(q)} DC_{\mathrm{err}}(\mathbf{p}, q)^2 \tag{3.42}$$

where $\mathcal{D}(q)$ is the set of pixels in the DC image corresponding to motion inliers in the qth P-frame and $|\mathcal{D}(q)|$ its cardinality.

Then, E is used as a confidence measure for the accuracy of the motion model estimated from the MPEG motion vectors and we introduce it in the weight computation:

$$w_{\mathrm{l},q} = \frac{1}{\mathrm{E}(q)} \tag{3.43}$$

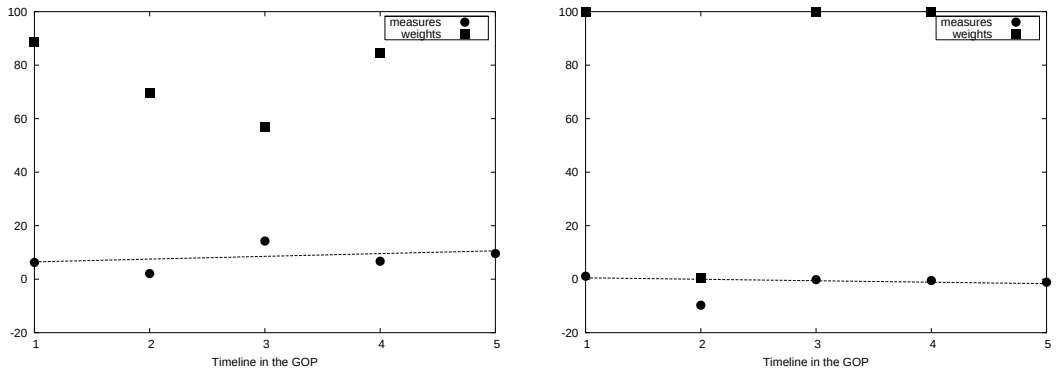
The weights are then normalised in $[0, 1]$ as:

$$w_{\mathrm{l},q} = \frac{w_{\mathrm{l},q}}{\max_{j=1,\ldots,Q} w_{\mathrm{l},j}} \tag{3.44}$$

The model of Equation (3.37) requires at least two measures, i.e. two P-frames, in order to approximate a straight line. It can happen that less P-frames are available due to frame loss or an unusual encoding. If only one P-frame is available, a solution might be to repeat its motion. If no P-frame is available, zero motion might be used. These situations have to be detected when the MPEG stream is parsed. Nevertheless, this approach seams too much simplified, since in the case of mosaicing the frame superposition is most likely not accurate enough. A more appropriate solution is to reestimate motion e.g. using the method we will present below.

Figure 3.11 shows two examples for the extrapolation of a motion parameter in the GOP according to (3.37). The GOP structure in the sequence is $IBBPBBPBBPBBPBB$ with four P-frames. The measures in Figure 3.11(a) are noisy with varying weights. The motion parameter of the third P-frame holds the smallest weight i.e. the MPEG encoder motion estimation was quite inaccurate, thus its influence to the straight line is weak. In Figure 3.11(b) only one measure, that one of the second P-frame, differs. Its MPEG encoder motion estimation was very inaccurate and the weight is close to 0. Thus, this measure has no influence to the straight line which finally interpolates the other accurate measures.

We are aware that in this analysis of quality of motion estimation we do not consider the number of intra-coded macroblocks in the stream. Our assumption is that intra-coded macroblocks are really outliers in the motion estimation process and that at least some intra-coded block are available.



(a) (b)

Figure 3.11: Extrapolation of the motion parameters in a GOP: The graphs (a) and (b) show respectively an example of a linear regression straight line for a motion parameter. $x = 5$ corresponds to the predicted value.

Motion Correction for P-Frames

It is very important to exactly superimpose the images in mosaicing methods, otherwise ghost artefacts can appear. To obtain an exact superposition we need very accurate motion models in the computation of $\mathrm{T}(k)$. As stated above, the DC error energy E of a P-frame (3.42) is an indicator of the quality of the MPEG motion compensation and thus of the estimated model (3.31). If E is high for a P-frame, the motion model $\hat{\theta}$ is most likely erroneous and needs to be corrected. In the linear regression of Equation (3.38), we use E as a confidence measure for the accuracy of the motion model estimated from the MPEG motion vectors. Here, we use it as well for the decision if a motion model of a P-frame has to be corrected or not. The correction means that the values of the motion parameters obtained from encoded motion vectors (3.31) will be abandoned and replaced by the values on the straight line (3.37).

We propose a decision rule based on the online mean value of E in the sequence:

$$\text{if } \mathrm{E}(t) > \frac{C_{\mathrm{E}}}{t} \sum_{\tau=0}^{t} \mathrm{E}(\tau) \text{ then } a_j(t) = \gamma_j + \eta_j \cdot q \tag{3.45}$$

where C_{E} is a constant, and q the position of the current P-frame in the GOP. We determined experimentally the value of C_{E} which is usually in the range $[1, 5]$ depending on the characteristics of the sequence. This decision rule means that if the error energy of the current image varies significantly with respect to the mean value, then the encoded motion vectors were not accurate and the estimated motion model has to be corrected.

Motion Reestimation for P-Frames

The motion correction presented above is insufficient if a strong motion change occurs. To this end, we propose reestimating the motion by the gradient descent method. The objective

thereby is to estimate the motion vector field that minimises the error between the motion compensated image using this motion vector field and the image itself. We minimise this error in the DC domain using the DC image of the P-frame instead of the completely decoded frame.

In this work, we follow the region-based motion estimation of Wu [204]. Indeed, the low-resolution DC image can be considered as a region of dominant motion, that one of the camera. Consequently, we consider only motion inliers to obtain a region of dominant camera motion in the DC image. This method was developed in the Master thesis [52]. In the following, we present this method to estimate the affine six-parameter model.

The error criterion to be minimised is the mean square error (MSE) computed on the motion inliers (following the global camera motion):

$$\text{MSE}(t) = \frac{1}{|\mathcal{D}(t)|} \sum_{\mathbf{p} \in \mathcal{D}(t)} \text{DFD}(\mathbf{p}, \mathbf{d})^2 = \frac{1}{|\mathcal{D}(t)|} \sum_{\mathbf{p} \in \mathcal{D}(t)} \left(I(\mathbf{p}, t) - I(\mathbf{p} + \mathbf{d}, t-1) \right)^2 \tag{3.46}$$

with DFD as the displaced frame distance, $\mathbf{p} = (x, y)^{\mathrm{T}}$ a pixel and $\mathbf{d} = (d_x, d_y)^{\mathrm{T}}$ the associated motion vector consistent with the model (3.23).

Denoting $\tilde{\theta} = (a_1, \ldots, a_6)^{\mathrm{T}}$ as the parameter vector of the motion model to estimate, it will be optimised for the DC frame (excluding motion outliers) in order to obtain the optimal motion compensation vectors $\mathbf{d}(\tilde{\theta})$. We estimate $\tilde{\theta}$ by the iterative gradient descent scheme:

$$\tilde{\theta}^{i+1}(t) = \tilde{\theta}^{i}(t) - \frac{\mathbf{M}}{2 \cdot |\mathcal{D}(t)|} \mathbf{G}^i \tag{3.47}$$

with

$$\mathbf{G}^i = \sum_{\mathbf{p} \in \mathcal{D}(t)} \begin{pmatrix} \frac{\partial}{\partial a_1} \text{DFD}(\mathbf{p}, \mathbf{d}^i)^2 \\ \vdots \\ \frac{\partial}{\partial a_6} \text{DFD}(\mathbf{p}, \mathbf{d}^i)^2 \end{pmatrix} \tag{3.48}$$

where $\mathbf{G}^i$ is the vector of gradients of the error function (3.46), $\tilde{\theta}^i(t)$ is the estimation of the parameter vector at the ith iteration, $\mathbf{d}^i$ is the optimised motion vector with respect to $\tilde{\theta}^i(t)$, and $\mathbf{M}$ is the gain matrix related to $\tilde{\theta}^i(t)$.

Developing $\mathbf{G}^i$ we obtain:

$$\frac{\partial}{\partial a_j} \text{DFD}(\mathbf{p}, \mathbf{d}^i)^2 = 2 \cdot \text{DFD}(\mathbf{p}, \mathbf{d}^i) \cdot \frac{\partial}{\partial a_j} \text{DFD}(\mathbf{p}, \mathbf{d}^i), \quad 1 \leq j \leq 6 \tag{3.49}$$

Using the definition of the DFD:

$$\begin{aligned} \frac{\partial}{\partial a_j} \text{DFD}(\mathbf{p}, \mathbf{d}^i) &= -\frac{\partial}{\partial a_j} I(\mathbf{p} + \mathbf{d}^i, t-1) \\ &= -\frac{\partial}{\partial x} I(\mathbf{p} + \mathbf{d}^i, t-1) \cdot \frac{\partial}{\partial a_j}(x + d_x) \\ &\quad -\frac{\partial}{\partial y} I(\mathbf{p} + \mathbf{d}^i, t-1) \cdot \frac{\partial}{\partial a_j}(y + d_y) \end{aligned} \tag{3.50}$$

Since the motion compensation vectors are a function of the parameters a_j, $\mathbf{G}^i$ can be expressed explicitly. Assuming that:

$$\begin{aligned}\frac{\partial}{\partial x}I(\mathbf{p},t-1) &= \nabla I_y(\mathbf{p},t-1)\\ \frac{\partial}{\partial y}I(\mathbf{p},t-1) &= \nabla I_x(\mathbf{p},t-1)\end{aligned} \tag{3.51}$$

where ∇I_x and ∇I_y are respectively the spatial gradients in x and y-direction.

Then, we obtain:

$$\begin{aligned}\frac{\partial}{\partial a_1}\text{DFD}(\mathbf{p},\mathbf{d}^i) &= -\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\\ \frac{\partial}{\partial a_2}\text{DFD}(\mathbf{p},\mathbf{d}^i) &= -(x-x_0)\cdot\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\\ \frac{\partial}{\partial a_3}\text{DFD}(\mathbf{p},\mathbf{d}^i) &= -(y-y_0)\cdot\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\\ \frac{\partial}{\partial a_4}\text{DFD}(\mathbf{p},\mathbf{d}^i) &= -\nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\\ \frac{\partial}{\partial a_5}\text{DFD}(\mathbf{p},\mathbf{d}^i) &= -(x-x_0)\cdot\nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\\ \frac{\partial}{\partial a_6}\text{DFD}(\mathbf{p},\mathbf{d}^i) &= -(y-y_0)\cdot\nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\end{aligned} \tag{3.52}$$

Finally, Equation (3.47) can be expressed as:

$$\tilde{\theta}^{i+1}(t) = \tilde{\theta}^i(t) - \frac{1}{|\mathcal{D}(t)|}\sum_{\mathbf{p}\in\mathcal{D}(t)}\mathbf{M}\cdot\text{DFD}(\mathbf{p},\mathbf{d}^i)\cdot\mathbf{F}^i \tag{3.53}$$

with

$$\mathbf{F}^i = -\begin{bmatrix}\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\\ (x-x_0)\cdot\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\\ (y-y_0)\cdot\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\\ \nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\\ (x-x_0)\cdot\nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\\ (y-y_0)\cdot\nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\end{bmatrix} \tag{3.54}$$

The gain matrix $\mathbf{M}$ is that of Caffario [127]:

$$\mathbf{M} = \frac{1}{\|\nabla I_x(\mathbf{p}+\mathbf{d}^i,t-1)\|^2 + \|\nabla I_y(\mathbf{p}+\mathbf{d}^i,t-1)\|^2 + C_{\mathbf{M}}^2}\begin{bmatrix}\varepsilon_1 & 0 & 0 & 0 & 0 & 0\\ 0 & \varepsilon_2 & 0 & 0 & 0 & 0\\ 0 & 0 & \varepsilon_3 & 0 & 0 & 0\\ 0 & 0 & 0 & \varepsilon_4 & 0 & 0\\ 0 & 0 & 0 & 0 & \varepsilon_5 & 0\\ 0 & 0 & 0 & 0 & 0 & \varepsilon_6\end{bmatrix} \tag{3.55}$$

with $C_{\mathbf{M}}$ as a constant depending on the noise, and $\varepsilon_1, \varepsilon_6 = 0.1$ and $\varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5 = 0.001$. We chose the gain values of the translational parameters higher than these of the affine parameters in order to obtain a fast convergence of the method. This can be explained by the fact

that a_1, a_4 represent directly pixel displacements, while the others represent ratios and have typically very small values.

As the gradient descent method is very likely to converge to a local minimum and not to the global minimum, depending on the initial values of $\tilde{\theta}^0$, we use the motion parameters of the straight line (3.37) as a first approximation to $\tilde{\theta}$. Thus:

$$\tilde{\theta}^0 = (\gamma_1 + \eta_1 \cdot q, \gamma_2 + \eta_2 \cdot q, \gamma_3 + \eta_3 \cdot q, \gamma_4 + \eta_4 \cdot q, \gamma_5 + \eta_5 \cdot q, \gamma_6 + \eta_6 \cdot q)^{\mathrm{T}} \tag{3.56}$$

where γ_j, η_j are the parameters of the straight line to approximate a_j, and q is the index of the current P-frame in the GOP.

The algorithm stops either if a fixed number of iterations is achieved ($i = 6$ in our case) or if the following error criterion is satisfied:

$$\frac{\left|\mathrm{MSE}^i - \mathrm{MSE}^{i-1}\right|}{\max\left(\mathrm{MSE}^i, \mathrm{MSE}^{i-1}\right)} < 0.1 \tag{3.57}$$

Figure 3.12 shows the MPEG motion vectors and the motion vectors after the correction for a frame extracted from a zoom sequence in the TRECVID 2005 video data. The MPEG compensation vectors (Figure 3.12(e)) are very noisy, so that the robust global motion estimator can not fit an appropriate motion model and detects a static camera, although the zoom can be clearly observed in the difference image Figure 3.12(d). Using the decision rule from Equation (3.45) with $C_{\mathrm{E}} = 4.0$, we detect this erroneous motion model and correct it. The motion vectors obtained with the corrected model are shown in Figure 3.12(f) which correspond to the observed zoom.

Concatenation of Motion Models in a Global Geometric Transformation

When the motion models have been calculated for the sequence, they need to be inverted depending on the choice of reference frame and then concatenated in order to obtain the geometrical transformation $\mathrm{T}(k, r)$ from the kth frame to the coordinate system of the reference frame, the rth frame.

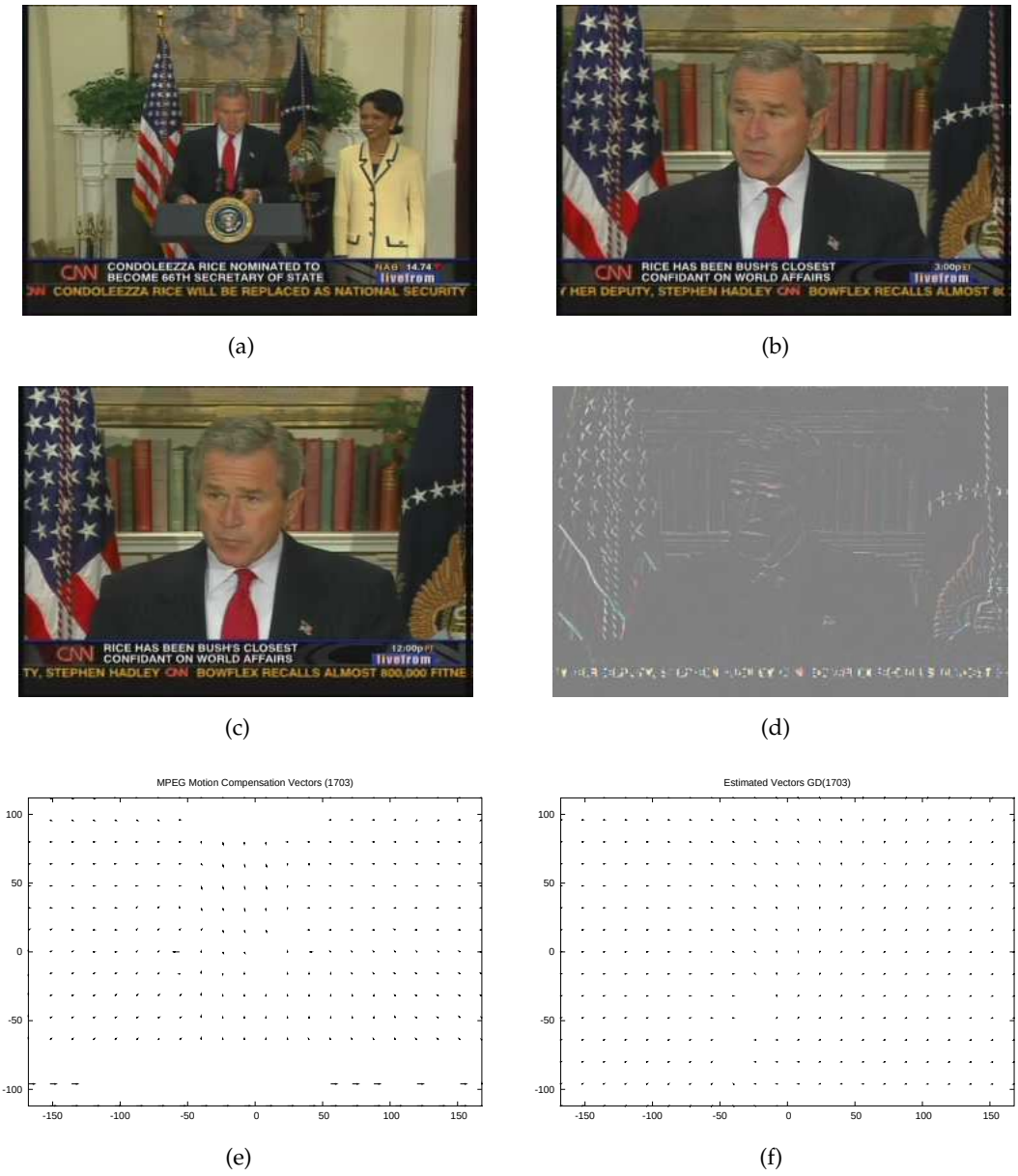
When compensating by motion the kth frame for mosaic construction, we use motion compensation vectors pointing from the mosaic into the kth frame. This avoids holes in the mosaic as for each pixel position in the mosaic a pixel value is available pointed by the motion vector.

The estimated motion models are computed at macroblock basis and thus express motion at full-resolution of the video. Thus, for the use on DC images, the motion models have to be scaled. As the estimated motion models are backward, the motion compensated position of a pixel $\mathbf{p}$ of the kth frame in the previous frame is according to (3.23):

$$\hat{\mathbf{p}}(\mathbf{p}, k-1) = \mathbf{p} - z \cdot \mathbf{T}(k) + \mathbf{A}(k)\mathbf{p} \tag{3.58}$$

with

$$\mathbf{T} = \begin{pmatrix} a_1 \\ a_4 \end{pmatrix} \text{ and } \mathbf{A} = \begin{pmatrix} a_2 & a_3 \\ a_5 & a_6 \end{pmatrix} \tag{3.59}$$



(a) (b)

(c) (d)

(e) (f)

Figure 3.12: Example of reestimating a motion model for a P-frame: (a) and (c) the first and last frame of the zoom sequence, (b) the P-frame with an erroneous motion model, (d) the difference image of (b) and is reference frame, (e) the MPEG motion compensation vectors of the frame (b), (f) the motion compensation vectors after the motion correction of the frame (b). This sequence was extracted from the TRECVID 2005 corpus for low-level feature extraction.

where $\mathbf{T}$ is the matrix of the translational parameters of the motion model, $\mathbf{A}$ is the matrix of the affine parameters of the motion model, and z the scale factor of the model model. In order to scale the full-resolution motion models to the DC domain z has to be chosen as $1/8$.

Depending on the choice of the reference frame, the motion models of the image in the future of the reference frame have to be inverted before concatenating them in the geometrical transformation. The forward motion vector pointing from the image at time $t-1$ to the image at time t can be obtained from the backward motion model as:

$$\hat{\mathbf{p}}^{-1}(\mathbf{p}, t-1) = (\mathbf{A}(t) + \mathbf{E})^{-1}(\mathbf{p} - \mathbf{T}(t)) \tag{3.60}$$

where $\mathbf{E}$ is the identity matrix.

Finally, the geometrical transformation of the kth image to the rth frame, the reference frame, can be computed as:

$$\mathrm{T}(\mathbf{p}, k, r) = \begin{cases} \hat{\mathbf{p}}^{-1}(\ldots \hat{\mathbf{p}}^{-1}(\hat{\mathbf{p}}^{-1}(\mathbf{p}, r+1), r+2), \ldots k) & \text{if } k > r \\ \hat{\mathbf{p}}(\ldots \hat{\mathbf{p}}(\hat{\mathbf{p}}(\mathbf{p}, r), r-1), \ldots k) & \text{if } k < r \\ \mathbf{p} & \text{if } k = r \end{cases} \tag{3.61}$$

The color value of the decimal position in the frame is bilinearly interpolated similar to Equation (2.14). In the following, we will use the notation $\mathrm{T}(k)$ instead of $\mathrm{T}(k, r)$ in cases if the choice of the reference frame can be neglected.

Comparison with Other Motion Estimation Tools

In order to assess the performance of our motion estimation method with respect to common methods, we chose:

- Motion2D: The Motion2D software [113] estimates 2D parametric motion models between two successive images. It can handle several types of motion models including the six-parameter affine model. The estimation method is based on a robust multi-resolution scheme where an error functional is minimised using an iterative reweighted least square method.
- Proesmans+LSF: We used the implementation of Proesmans's et al. optical flow algorithm [146] which is available at [131]. This is essentially a multi-scale, anisotropic diffusion variant of Horn and Schunk's algorithm. In [110], it is shown that this algorithm performs well in comparison to common optical flow estimation algorithms. Afterwards, we calculate a least square fit (LSF) of the motion vectors to determine the corresponding six-parameter affine model.

In order to compare the performance of our method in the compressed domain with the two methods for raw video, we compute the MSE of motion compensations with each model on the decoded full-resolution frames. We compute the MSE only on the luminance component as this is the case in standard motion estimation approaches. Here, motion compensation is performed at temporal I/P-resolution. Figure 3.13 visualises the graphs of the MSE

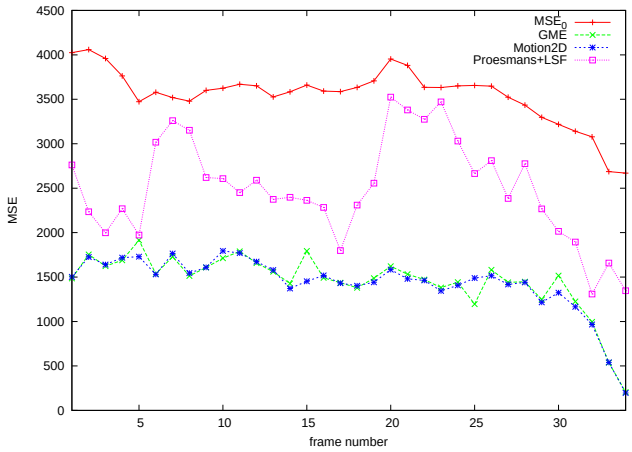


Figure 3.13: Mean square error (MSE) of the motion compensation error for 35 I/P-Frames of the sequence "Tympanon", respectively, without motion compensation (MSE_0), for the GME, Motion2D and Proesmans+LSF motion estimation.

without motion compensation (MSE_0) and with motion compensation for the three methods respectively versus the frame number. As it can be seen from Figure 3.13 GME which uses motion vectors from the MPEG stream and a correction of the motion model gives on average the same quality as Motion2D in the pixel domain. As both of them are based on a robust estimator of Tukey [130] and so incorporate an outlier rejection scheme, they outperform the Proesmans+LSF method. In this experiment, we chose $C_{\mathrm{E}} = 1.25$ for the motion correction (3.45) whereas 3 of 35 frames have been corrected. In all cases we obtain a lower MSE with the model correction than without it.

The comparison of the motion parameter values in Figure 3.14 shows that GME and Motion2D estimated parameters are very close. Some discrepancies are observed for the motion parameters a_1 and a_4. For a_1 the frames 5, 15, 25 and 30 and for a_4 the frames 5, 15, and 20 are concerned. These frames are I-Frames where the motion has been extrapolated. In contrast to this, Proesmans+LSF shows some noisy values, but the principal difference appears in the parameter a_4 which describes the vertical movement of the camera. Its value never exceeds 15 pixels, whereas GME and Motion2D show values higher than 20. Although we used 4 multi-resolution levels like for Motion2D, this method seems to be limited in case of large camera motions.

These experiments show that the use of MPEG motion vectors is justified despite their noisiness. The time figures are strongly in favor of GME. Without counting the decoding time it takes 0.615 seconds for processing 60 I/P-Frames with respect to 21.664 seconds for Motion2D, and 75 minutes for Proesmans+LSF. These times were measured on an Intel Pentium 4 3.00GHz processor on the sequence "Tympanon" CERIMES-SFRS® (see Figure 3.27(a)). This ratio of computational time is typical for other sequences we processed.

As illustrated in Figure 3.7 our mosaicing method consists of several steps. A method for the registration of the image of the sequence from MPEG-1/2 compressed video was shown above. Then, in the next section, we present a method for the extraction of moving objects in

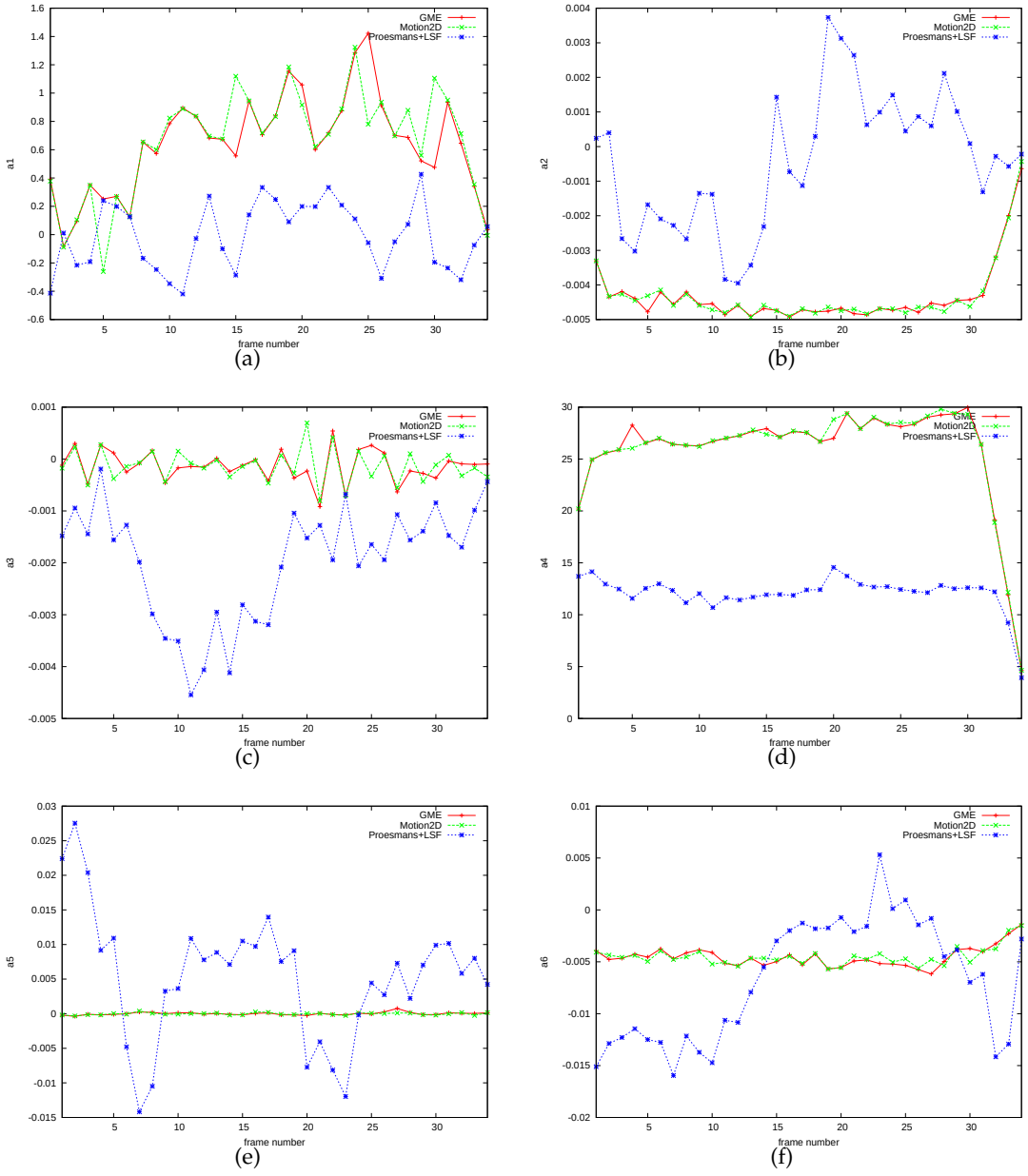


Figure 3.14: (a)–(f) Graphs of the motion parameters a_j for the GME, Motion2D and Proesmans+LSF motion estimation, respectively, for the sequence "Tympanon".

the sequence which is the next step in Figure 3.7.

3.2.3 Moving Object Detection

As shown in Figure 3.15 ghost artefacts can appear when moving objects are not excluded from the mosaic blending. The sequence of DC images corresponding to this mosaic is shown in Figure 3.27(c). Thus, the second step of the mosaic construction (see Figure 3.7) is the detection of moving objects in the video sequence. The objective hereby is to avoid ghost artefacts by introducing masks in the blending process defining if a pixel will contribute to the mosaic or not. We describe below how these masks are obtained.

Figure 3.15: Mosaic with ghost artefacts when moving objects are not excluded from the blending for the sequence "Hiragasy".

Methods presented in literature exclude moving objects from the blending based on outlier rejection schemes. Intensity residual based schemes compare typically the current frame with a warped background image. For instance, the method proposed in [198], detects moving object by subtracting a warped background image from the current frame (see Equation 3.11). To construct an initial background image, the authors assume that the first two frames of the sequence do not contain moving objects. This is a very restrictive assumption and it is not often hold for broadcast video. Thus, we consider using a method based on motion vectors, namely motion vectors extracted from the compressed stream. This has two drawbacks. The encoded motion vectors are very noisy so that a simple outlier rejection scheme would not give satisfying results, e.g. see Figure 3.9 (outlier borders are not really objects). We need to detect moving objects at I-frame basis, but no motion vectors are available in the compressed stream for I-frames. Therefore, we propose using a moving object detection based on Manerba [104]. This method combines a motion analysis and outlier rejection of P-frame motion vectors with a color segmentation of I-frames. Finally, the color and motion information are merged together at I-frame basis to extract the foreground objects. The reader can find more details in [104]. Here, we resume briefly the method.

The method [104] uses the same robust motion estimator (see Section 3.2.2) to extract macroblocks with local motion. They are characterised in Equation (3.31) by weights equal or close to zero. Thus, when thresholding these weights a binary mask of outlier macroblocks is obtained as shown in Figure 3.9(e). In this example some outliers appear on the top border in direction of the camera motion (a tilt up). These outliers correspond to new macroblocks

entering into the frame, thus there are no corresponding macroblocks in the reference frame and noisy motion vectors result in this region as other macroblocks are referenced. In order to filter the outliers on the frame border, the estimated camera motion model θ is used. By warping the anchor frame onto the current frame, the geometry of the region entered into the frame can be determined. If outliers are present in this region, they are assumed to be caused by the camera motion and are not further considered for the object mask.

Repeating this for each P-frame in the shot a first guess of objects in each P-frame results. As each mask is obtained independently from the others, they remain noisy in time. Therefore, a 3D filtering is applied to smooth the detection along the time. A 3D volumetric mask of a GOP is constructed. Then, 3D morphological filtering is applied in order to eliminate noise, i.e. isolated macroblocks that, although following the global motion, do not represent a foreground object. Once noise and outliers due to camera motion have been filtered out, a 3D region growing algorithm is applied in order to smooth the object mask along the time.

The extracted motion masks for P-frames are a good guess of the object shape in P-frames, but this is not sufficient for a robust object extraction. Therefore, this information is merged with a color segmentation of DC images of I-frames. As motion masks are only available for P-frames, the motion mask O_{M} for an I-frame is obtained by spatially interpolating the motion masks of the neighbouring P-frames:

$$O_{\mathrm{M}}(x, y, t) = \min(\mathrm{D}.O_{\mathrm{M}}(x, y, t-1), \mathrm{D}.O_{\mathrm{M}}(x, y, t+1)) \tag{3.62}$$

where D is the morphological dilation operator with a 4-connected structural element of radius 1. In this way the approximate position and shapes of the objects in the I-frame are obtained.

Using the color information of DC images of I-frames, the object shapes can be refined. A prefiltering using a partial reconstruction filter is applied to the DC images in order to reduce the noise in the DC image. In [104], a morphological color segmentation was developed, but it tends to oversegment the images. Therefore, it was replaced by the hierarchical color segmentation method of Fuh et al. [50]. A result of the color segmentation is shown in Figure 3.16.

(a) (b)

Figure 3.16: Result of the color segmentation of a frame of the sequence “Comportements1”: (a) a DC image, (b) the obtained segmentation for this image.

If the motion mask of the I-frame is superimposed with the color segmentation, it happens that a region of the color segmentation is not completely included in the motion mask. If the ratio of the pixels of a region included in the mask with respect of the total number of

the pixels in the same region exceeds a threshold, typically 80%, the whole region is considered to be a part of the object.

Once the objects are extracted, it is often necessary to delete flat zones belonging to the background. In these zones the MPEG motion estimation typically fails and noisy motion vectors result. Thus, macroblocks belonging to a flat region are likely to be associated with motion vectors that do not follow the camera motion and so are erroneously detected as moving objects. As these zones are characterised by a very low color gradient, the average gradient energy of the region per macroblock is used to eliminate such false detections. If the average gradient energy drops below a predefined threshold the region is declared as a flat zone and is excluded.

Due to the low resolution of the frames or the lack of relative motion between the camera and the objects, detection errors may occur. In order to avoid miss-detections the results of neighbouring frames can be taken into account to filter the objects along the time. Therefore, objects are tracked along the sequence of I-frames. To do this, motion is estimated for each detected object in the current I-frame and then forward projected onto the succeeding I-frame. This projection is performed recursively along the P-frames between them.

The object is assumed to undergo affine motion described by Equation (3.23). The motion model for an object in a P-frame is obtained by a least square estimation using the encoded MPEG motion vectors of macroblocks belonging to the object:

$$\hat{\theta}_{\mathrm{O}} = \left(\mathbf{H}_{\mathrm{O}}^{\mathrm{T}}\mathbf{H}_{\mathrm{O}}\right)^{-1}\mathbf{H}_{\mathrm{O}}^{\mathrm{T}}\mathbf{Z}_{\mathrm{O}} \tag{3.63}$$

The vector $\mathbf{Z}_{\mathrm{O}}$ is formed by the MPEG motion compensation vectors of the object, $\mathbf{H}_{\mathrm{O}}$ are the corresponding macroblock centers, and $\hat{\theta}_{\mathrm{O}}$ is the same backward six-parameter affine model describing the object motion. In order to estimate a motion model for I-frames, in [104] an interpolation of the parameters of the neighbouring P-frames was proposed. Here, we extrapolate the motion parameters of the P-frames in the previous GOP by a least square estimation (similar to Equation (3.38), but the weights are set to 1).

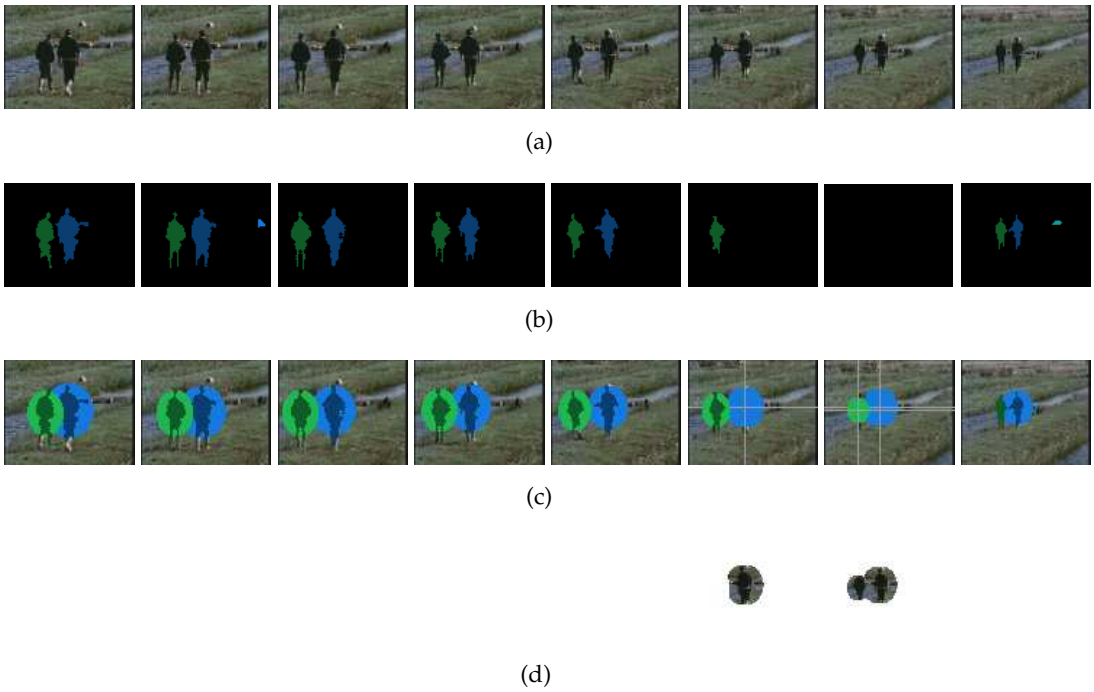
Then, the motion models are inverted and the object is recursively forward projected. If the projection result overlaps with a detection of the succeeding I-frame, the object is identified for the considered pair of frames. In this way, the object labels are propagated, starting from the first I-frame, along the sequence establishing a correspondence between the same object in different frames.

Under the assumption that an object can not appear and disappear during a short sequence of frames, the sequence of objects at temporal I-frame resolution is filtered in time in order to find the object even if the motion based detection failed. Using the reconstructed object trajectory the shape of an object can approximated by a conic function in the frames where the object was not detected. Therefore, a kind of tube is reconstructed where each section provides the object position in the frame along the time. Using the assumption that the object trajectory is linear, the tube is represented by a quadric function in a three dimensional space (2D + t):

$$\nu_{11}x^2 + \nu_{22}y^2 + \nu_{12}t^2 + \nu_{12}xy + \nu_{13}xt + \nu_{23}yt + \nu_{14}x + \nu_{24}y + \nu_{34}t + \nu_{44} = 0 \tag{3.64}$$

with ν_{ij} as the parameters of the quadric.

Finally, merging the color and the cross section of the tube the object shape can be obtained. An example is shown in Figure 3.17. The labels of the two persons are propagated along the sequence. Even after the lost of the persons in the sixth and seventh frame, the persons are identified in the eighth frame. The shapes of the lost objects for the sixth and seventh frame are approximated by the tube.



(a)

(b)

(c)

(d)

Figure 3.17: Results of the moving object detection: (a) The sequence of DC images of I-frames "Aquaculture1" extracted from the documentary "Aquaculture en méditeranée" CERIMES-SFRS®, (b) the object masks after the tracking, (c) the reconstructed tube, (d) the objects approximated by the tube.

The resulting object label masks O_l are of the form:

$$O_\mathrm{l}(\mathbf{p}) = \begin{cases} l & \text{if } \mathbf{p} \text{ belongs to a moving object} \\ 0 & \text{if } \mathbf{p} \text{ belongs to the background} \end{cases} \tag{3.65}$$

where $l > 0$ is the label of the object.

Thresholding these masks, we obtain a characteristic function of the background:

$$O_\mathrm{b}(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p} \text{ belongs to the background} \\ 0 & \text{if } \mathbf{p} \text{ belongs to a moving object} \end{cases} \tag{3.66}$$

These mask will be used in the following to exclude moving objects in the illumination correction and in the blending.

3.2.4 Illumination Correction

Illumination changes in the image sequence can be due to different viewing points of the scene, a change of global illumination conditions of the scene, occlusions between the scene and the lighting source and the sensor characteristics. This can cause seems in the mosaic at the image borders and the region where moving objects have been excluded (e.g. see Figure 3.22(d)). Hence, illumination correction is the third step in Figure 3.7.

We assume a linear model for illumination changes between two images like in [144, 128]. Assuming that the kth frame in the sequence is corrected with respect to the $r$the frame, the linear model is:

$$\mathbf{Y}(k) = \iota \cdot \mathbf{Y}(r) + \kappa \tag{3.67}$$

where $\mathbf{Y}$ denotes the luminance component of the image, and ι and κ are the parameters to adjust contrast and brightness. This is in contrast to [21] where a linear model for the RGB components is assumed (see Equation (3.15)), but considering only the luminance component is more robust.

In [144, 128] only the pixels at the contours of the intersection of the two images are considered for the computation of the parameters ι and κ as shown in Figure 3.5. In the case of DC images which are of very small size this number of pixels might not be sufficient for a robust estimation. Thus, we consider the entire intersection. The main problem here is to exactly superimpose the two images as the linear model is very sensitive to estimation errors. First, moving objects can not be superimposed by the estimated model. Second, due to the strong aliasing of DC images, edges and textures may not be superimposed exactly even if the geometric transformation is accurate. Therefore, we need a robust estimation scheme with an efficient outlier rejection. To realise this:

- We reject in advance pixel correspondences belonging to moving objects.
- We reject in advance pixel correspondences belonging to edges and textures.
- We use a weighted least square estimation with an outlier rejection based on the Tukey function similar to that of the motion estimation.

Let $\hat{\mathbf{p}} = (\hat{x}, \hat{y})$ be the motion compensated coordinates of the pixel $\mathbf{p}$ obtained through the geometric transformation $\mathrm{T}(k, r)$. Then, the intersection of the images $\mathbf{Y}(k)$ and $\mathbf{Y}(r)$ is given by:

$$\mathcal{Y}_{\cap} = \{(\mathbf{p}_i, \hat{\mathbf{p}}_i) \mid \mathbf{p}_i \in \mathbf{Y}(k) \;\&\; \hat{\mathbf{p}}_i \in \mathbf{Y}(r)\} \tag{3.68}$$

We use the background function of Equation (3.66) to exclude pixel correspondence belonging to moving objects. We obtain:

$$\mathcal{Y}_{\cap,\mathrm{O}} = \{(\mathbf{p}_i, \hat{\mathbf{p}}_i) \mid \mathbf{p}_i \in O_{\mathrm{b}}(k).\mathbf{Y}(k) \;\&\; \hat{\mathbf{p}}_i \in O_{\mathrm{b}}(r).\mathbf{Y}(r)\} \tag{3.69}$$

where $O_{\mathrm{b}}(k)$ and $O_{\mathrm{b}}(r)$ define, respectively, the background of the kth and rth image.

In order to determine if a pixel $\mathbf{p} \in \mathbf{Y}(k)$ corresponds to an edge or texture in $\mathbf{Y}(r)$, we compute the norm of Roberts gradient as:

$$\|\nabla_{\mathrm{r}}\mathbf{Y}(\hat{\mathbf{p}}, r)\| = |\mathbf{Y}(\lfloor\hat{x}\rfloor, \lfloor\hat{y}\rfloor, r) - \mathbf{Y}(\lceil\hat{x}\rceil, \lceil\hat{y}\rceil, r)| + |\mathbf{Y}(\lfloor\hat{x}\rfloor, \lceil\hat{y}\rceil, r) - \mathbf{Y}(\lceil\hat{x}\rceil, \lfloor\hat{y}\rfloor, r)| \quad (3.70)$$

with $\lfloor\cdot\rfloor$ as the floor and $\lceil\cdot\rceil$ as the ceil operator.

By thresholding the norm of the Roberts gradient, the pixel correspondences belonging to edges or textures can be determined. Excluding them we obtain:

$$\mathcal{Y}_{\cap,\mathrm{O},\nabla} = \{(\mathbf{p}_i, \hat{\mathbf{p}}_i) \mid \mathbf{p}_i \in O_{\mathrm{b}}(k).\mathbf{Y}(k) \;\&\; \hat{\mathbf{p}}_i \in O_{\mathrm{b}}(r).\mathbf{Y}(r) \;\&\; \|\nabla_{\mathrm{r}}\mathbf{Y}(\hat{\mathbf{p}}_i, r)\| < \lambda_{\nabla}\} \quad (3.71)$$

with λ_{∇} as the threshold. We choose $\lambda_{\nabla} = 12.75$ which corresponds to 5% of the grey level range. Based on this reduced pixel set we can now define our robust estimation.

We use a weighted least square estimation to estimate the vector $\mathbf{L} = (\iota, \kappa)^{\mathrm{T}}$:

$$\mathbf{L} = (\mathbf{H}_{\mathrm{L}}^{\mathrm{T}}\mathbf{W}_{\mathrm{L}}\mathbf{H}_{\mathrm{L}})^{-1}\mathbf{H}_{\mathrm{L}}^{T}\mathbf{W}_{\mathrm{L}}\mathbf{Z}_{\mathrm{L}} \quad (3.72)$$

Denoting $N = |\mathcal{Y}|_{\cap,\mathrm{O},\nabla}$ as the number of pixels in the restrained intersection set, the matrices and vectors are constructed as:

$\mathbf{Z}_{\mathrm{L}}$ is the $N \times 1$ column vector of the measures:

$$\mathbf{Z}_{\mathrm{L}} = (\mathbf{Y}(\mathbf{p}_0, k), \mathbf{Y}(\mathbf{p}_1, k), \cdots, \mathbf{Y}(\mathbf{p}_{N-1}, k))^{T} \quad (3.73)$$

$\mathbf{H}_{\mathrm{L}}$ is the $N \times 2$ observation matrix:

$$\mathbf{H}_{\mathrm{l}} = \begin{pmatrix} \mathbf{Y}(\hat{\mathbf{p}}_0, r) & 1 \\ \mathbf{Y}(\hat{\mathbf{p}}_1, r) & 1 \\ \vdots & \vdots \\ \mathbf{Y}(\hat{\mathbf{p}}_{N-1}, r) & 1 \end{pmatrix} \quad (3.74)$$

where the luminance value $\mathbf{Y}(\hat{\mathbf{p}}_i, k)$ is obtained by bilinear interpolation.

$\mathbf{W}_{\mathrm{L}}$ is the $N \times N$ diagonal matrix of the weights as in Equation (3.35) where the weights are computed by the derivative of the Tukey function as defined in Equation (3.28):

$$w_{\mathrm{L},i} = \frac{\varrho'(\mathbf{r}_{\mathrm{L}}(\mathbf{p}_i), \lambda_{\varrho})}{\mathbf{r}_{\mathrm{L}}(\mathbf{p}_i)} \quad (3.75)$$

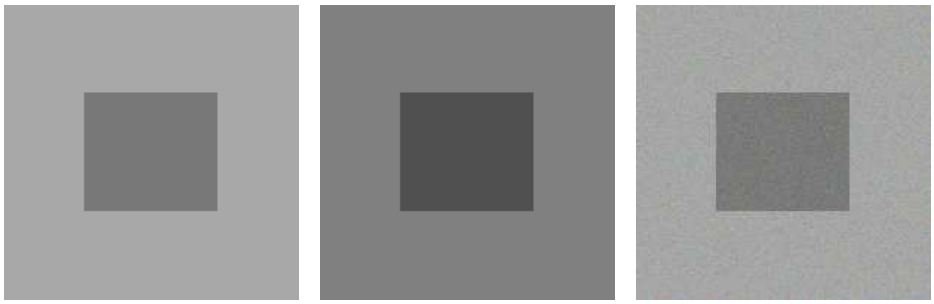
with $\lambda_{\varrho} = 5$. We determined its value experimentally. Here, the residuals are:

$$\mathbf{r}_{\mathrm{L}}(\mathbf{p}_i) = \mathbf{Y}(\mathbf{p}, k) - (\iota \cdot \mathbf{Y}(\hat{\mathbf{p}}_i, r) + \kappa) \quad (3.76)$$

In order to obtain a first guess of ι and κ, we compute a least square estimation i.e. all the weights are set to 1. Then a weighted least square estimation is performed using the weights as defined in Equation (3.75).

Despite the robust estimation scheme, small estimation errors may appear as the model of Equation (3.67) may be violated due to noise. In order to test the performance of this method, we created the test images of Figure 3.18 with $\iota = 1$ and $\kappa = 40$. Our first test

was to shift the image of Figure 3.18(b) in order to simulate aliasing and registration errors. The graphs of the estimated parameters ι and κ for different shift sizes are shown in Figures 3.19(a) and 3.19(b). We observe that for a shift size greater than 1, the parameter ι decreases and the parameter κ increases obtained by the least square estimation. This, means a reduced contrast at the expense of a high brightness offset. Nevertheless, the aberrant values are rejected in the weighted least square estimation and the correct values of ι and κ result. In this example, we obtain a robustness of the method until a shift size of 4.5. For shift sizes lager than 4.5 the matrix $\mathbf{H}_{\mathrm{L}}^{\mathrm{T}}\mathbf{W}_{\mathrm{L}}\mathbf{H}_{\mathrm{L}}$ is not invertible as all the weights are near zero.



(a) (b) (c)

Figure 3.18: Test images for the luminance correction: (a) and (b) the test images, (c) result of the luminance correction of (b) after adding noise of the variance 0.001.

The second test consist in adding noise to both test images (Figures 3.18(a) and 3.18(b)). The graphs of Figures 3.19(c) and 3.19(d) show, respectively, the values ι and κ for different noise variances. We observe that the obtained results are very close for the least square estimation and the weighted least square estimation. ι decreases and κ increases as the imaging model is violated by the noise. Nevertheless, for small noise variances the estimation error is small and the difference is not perceptible. This can be observed in Figure 3.18(c) which shows the luminance corrected image for a noise variance of 0.001.

Considering now an image sequence that has to be corrected, there is the question about the reference image for illumination correction. If the image sequence is long, generally there is no intersection of all images of the sequence. One possibility is to choose one reference image and then to correct recursively the neighbouring frames as illustrated in Figure 3.20(a). The problem hereby is that estimation errors are propagated as shown in Figure 3.21(a) resulting in a lost of contrast along the sequence.

In order to minimise the propagation of estimation errors, we propose correcting all images of the sequence with respect to the chosen reference image if the intersection is non empty. If there is no intersection with the reference image, we choose the nearest image to the reference which has a non empty intersection with the current image as reference frame. This produces a kind of hierarchy as illustrated in Figure 3.20(b). The sequence corrected with this scheme is shown in Figure 3.21(b) where the error propagation is limited and is thus not visible.

Figure 3.22(a) shows a sequence with a strong luminance change. The mosaic constructed

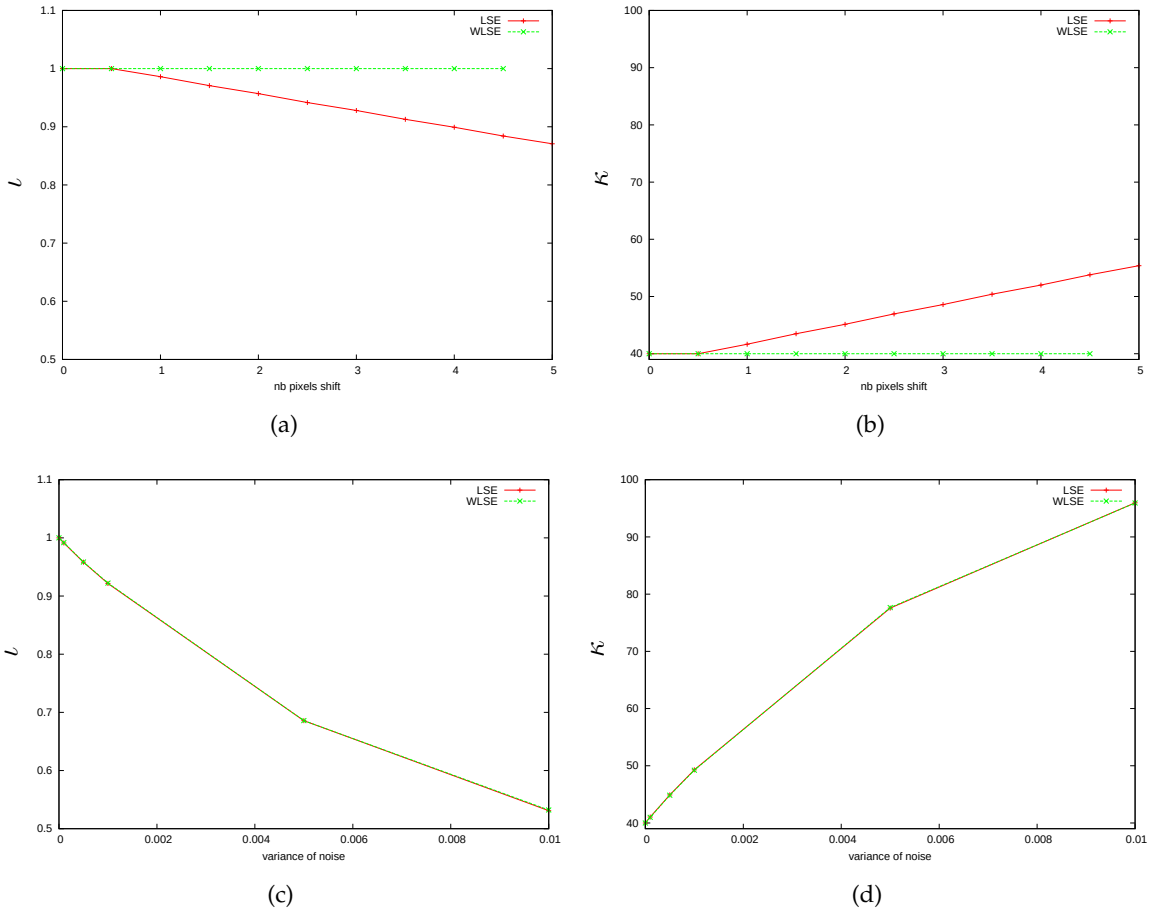


Figure 3.19: Graphs of the estimation of luminance parameters: (a) and (b) are the graphs for the parameters ι and κ, respectively, in case of shifts, (c) and (d) are the graphs for the parameters ι and κ, respectively, in case of noise.

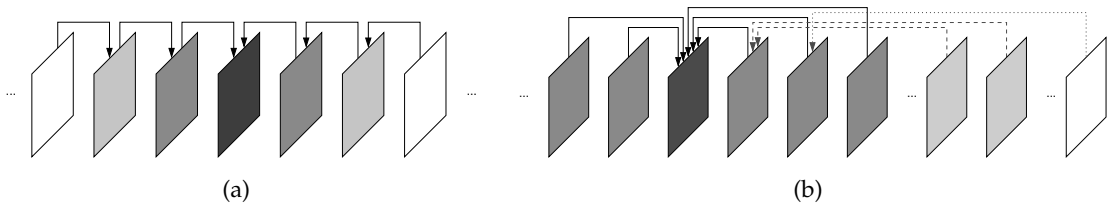
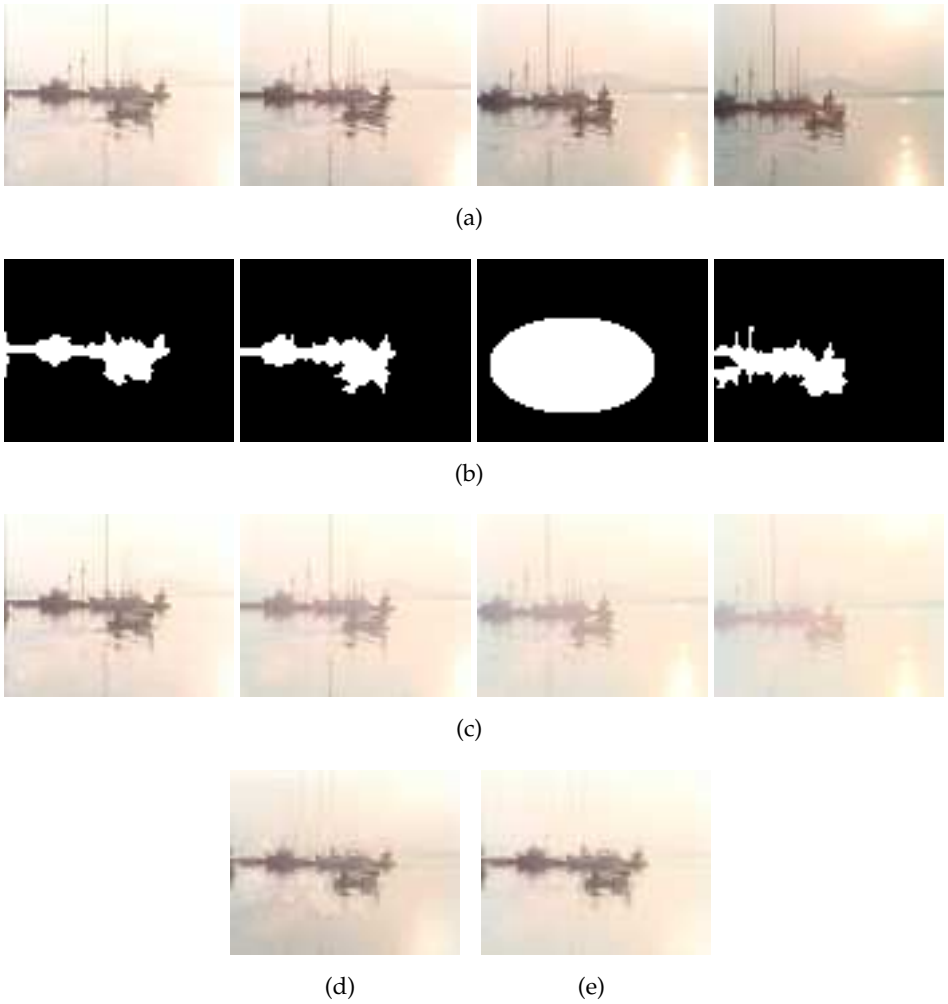


Figure 3.20: Luminance correction of an image sequence: (a) recursively, (b) hierarchically.

(a)

(b)

Figure 3.21: Example of luminance correction for the sequence "Chancre1" extracted from the documentary "Le chancre coloré du platane" CERIMES-SFRS® : (a) recursively, (b) hierarchically. The eighth frame was chosen as the reference.



(a)

(b)

(c)

(d) (e)

Figure 3.22: Example of luminance correction for the sequence "Aquaculture2" extracted from the documentary "Aquaculture en méditeranée" CERIMES-SFRS® : (a) The original sequence, (b) the object masks, (c) the illumination corrected sequence with the first image as reference, (d) the mosaic using the sequence in (a), (e) the mosaic using the sequence in (b).

from this sequence is shown in Figure 3.22(d). The object of the first frame was reinserted in the mosaic. We observe seams at the image borders and spurious irregularities where objects have been excluded. The sequence shown in Figure 3.22(c) has been illumination corrected with respect to the first image of the sequence. We observe a lost in contrast of the objects along the sequence as the illumination correction has been computed for the background. Nevertheless, this has no impact to the mosaic construction as only the background is blended into the mosaic. Note that the third object has been approximated by the tube. The mosaic constructed from the illumination corrected sequence is shown in Figure 3.22(e). We observe much less artefacts than in Figure 3.22(d). We mention here that for this sequence, motion has been estimated using the Motion2D software [113]. The reason is that the encoded MPEG motion vectors are continuously very noisy due to the large flat zones and thus we obtain no satisfying results with our motion estimation method. We show this example in order to demonstrate that our method can handle strong illumination changes if motion was estimated accurately.

When the input sequence is illumination corrected, the frames now can be blended into the mosaic. This is the next step in Figure 3.7.

3.2.5 Blending

Blending is the fifth step in our global scheme for mosaic construction (see Figure 3.7). Hence, in this section we describe our blending method which takes into account moving objects. First, we construct the background mosaic using the characteristic background function which defines for each frame which pixels belong to the background and will be blended into the mosaic. Second, some of the objects are blended afterwards into the background mosaic to obtain a complete description of the video sequence. However, the premise thereby is that the object masks are accurate i.e. that the objects were accurately segmented by the moving object detection described above. Thus, we will propose a method to deal with the matter if an object was not well segmented.

We use a simple approach for the mosaic blending based on a temporal average of the aligned images [129]:

$$M = \mu(K) \sum_{k=1}^{K} \mathrm{T}(k).I(k) \tag{3.77}$$

where $\mathrm{T}(k)$ is the geometric transformation from the image $I(k)$ to the mosaic M, K is the number of images $I(k)$ in the sequence, and $\mu(\mathbf{p}, K) = \frac{1}{|\mathbf{p}|}$ with $|\mathbf{p}|$ as the number of available pixels at position $\mathbf{p}$.

As we showed above, when using a temporal mean ghost artefacts can appear in the mosaic due to moving objects. Therefore, we use the background function of Equation (3.66) in the blending process. Introducing this function in Equation (3.77), we obtain:

$$M = \mu(K) \sum_{k=1}^{K} \mathrm{T}(k) \left[O_{\mathrm{b}}(k).I(k)\right] \tag{3.78}$$

where $O_{\mathrm{b}}(k)$ is the characteristic background function. A pixel is only blended into the mosaic if $O_{\mathrm{b}}(\mathbf{p}, k) = 1$. As we are working with RGB color images Equation (3.78) is applied to each color component.

This blending method supposes that the object masks are accurate. Nevertheless, in DC video sequences due to the low resolution and insufficient difference of objects' and background motions, segmentation errors often occur. For the method to be able to handle such real situations, a specific processing has to be proposed for inaccurate object segmentations.

Processing for Inaccurately Segmented Objects

It happens that the moving objects are not well segmented by the method presented above. Such an example is shown in Figure 3.23(a). In order to avoid possible artefacts due to the fact that the object is not entirely excluded from the mosaic blending, we propose two methods.

The first approach is a dilation of the object mask. The dilated object mask is shown in Figure 3.23(b). The problem here is that the pixel contributions in the mosaic are limited and so holes can appear in the mosaic or the size of existing holes can increase. To this end, we propose a method to fill holes in the mosaic in the next section.

The second approach is to use a temporal median in the blending in the corona around the excluded objects instead of the temporal mean. This approach is costlier than the first approach as median filtering is inherently a costly task. The corona around the object, as shown in Figure 3.23(c), is obtained by subtracting the dilated object mask from the original object mask. Then, all the coronas are warped to the mosaic by the geometric transformations and blended together. A mask is obtained which determines the pixels where temporal median filtering is applied.

(a) (b) (c)

Figure 3.23: Example of an inaccurate segmented object of the sequence "Hiragasy": (a) The inaccurate segmented object, (b) the object with a dilated mask (1 pixel), (c) the object corona (1 pixel width).

Figure 3.24(a) shows a mosaic where some objects have not been accurately segmented and ghost artefacts appear. The result obtained with a dilated mask is shown in Figure 3.24(b) and the results obtained with median filtering is shown in Figure 3.24(c). Figure 3.24(d) illustrates the regions where temporal median has been applied for the blending. It can be observed that the mosaic is noisy in the regions where median filtering has been applied as the mean value smooths much more than the median. Thus, we retain the dilation of the object masks as it is additionally less costly.

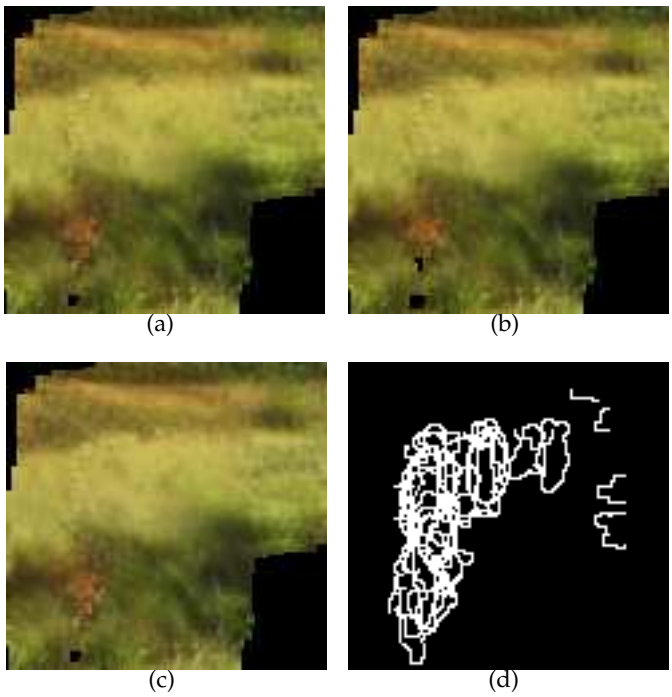


Figure 3.24: Example of processing for inaccurately segmented objects for the sequence "Hiragasy": (a) The mosaic obtained with the original masks, (b) the mosaic obtained with dilated masks (1 pixel), (c) the mosaic obtained with a region-wise temporal median, (d) the map of combined object coronas (1 pixel) indicating where the temporal median has been applied. We increased the contrast of (a), (b) and (c) for a better visualisation.

Insertion of an Object in the Mosaic

The blending approach presented above, creates a mosaic of the scene background. If we come back to the goal of this work which is the mosaic construction to visualise a video scene, the background mosaic alone is not sufficient. The image overview has to summarise all elements of the visual content. Thus, we blend some objects into this background mosaic in order to obtain a complete description of the video sequence. There are different approaches to construct such a "complete" mosaic. Depending on the approach, the reference frame is chosen. We distinguish:

- Background centered approach: The reference frame is chosen depending on the camera motion e.g. in case of a zoom the most contracted image is chosen in order to avoid the oversampling of the image sequence.
- Object centered approach: The reference frame corresponds to the frame where the objects are the best represented. This avoids the motion estimation and compensation of the objects which is a delicate task if the object is small or if the motion is complex.

Here, we focus on the object centered approach and we choose the frame with the best segmented objects as reference frame. We consider the objects in this frame as the representative objects of the scene and blend them directly into the mosaic. The choice of the best segmented objects is a subject of study by itself. Several criteria can be used to determine the quality of object segmentation [25]. In the present work, we suppose that the best segmented object has been already selected and focus on its insertion and improvement of the visual quality of the mosaic.

3.2.6 Postprocessing

In some cases, postprocessing is necessary in order to improve the visual quality of the mosaic (the last step in Figure 3.7). First, holes can appear in the mosaic due to the exclusion of objects during the blending. In this case, we propose *interpolating the lacking pixels* from the neighbourhood. Second, even when an illumination correction has been applied to the input sequence, slight seams may still appear in the mosaic on the borders of the background masks where objects have been excluded. Likewise, the dilation of the object mask may not be sufficient to entirely exclude an object in case of a very inaccurate segmentation. Thus, artefacts can appear. To this end, we propose applying a simple *spatial median filtering in the mosaic on the borders of the background masks*. Third, the insertion of an object causes typically seams at the borders of the object. To obtain a more realistic insertion of the object, we propose applying a *mean filter on the object borders*.

Interpolation of Holes

When excluding objects from the blending, it can happen that holes in the mosaic appear as shown Figure 3.25(a). To this end, we present a simple method for the detection and interpolation of holes.

A mask of the holes in the mosaic can be created straightforward during blending. If no pixel is available at the position $\mathbf{p}$, i.e. $\mu(\mathbf{p}, K) = 0$ then a hole appears at $\mathbf{p}$ in the mosaic and is marked in the mask. An example of such a mask is shown in Figure 3.25(b).

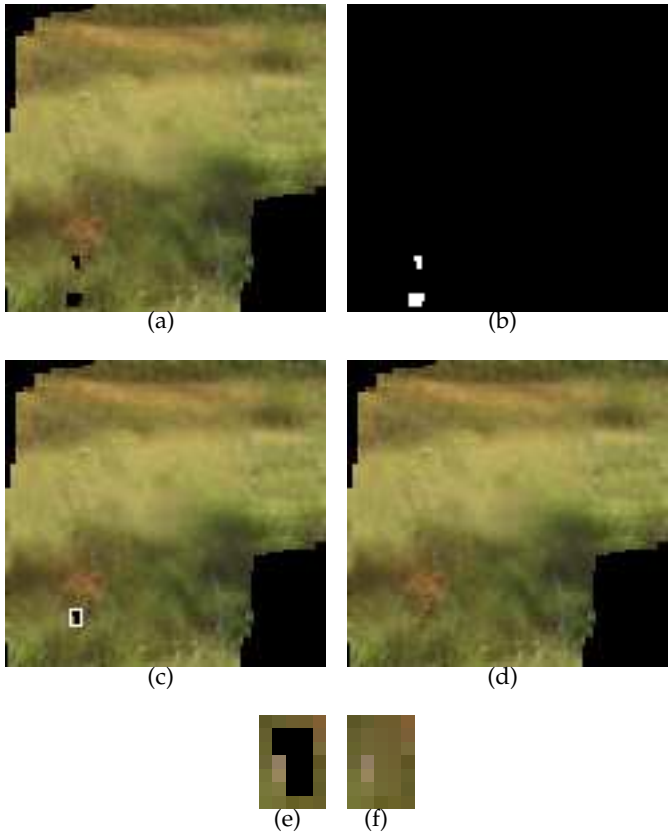


Figure 3.25: Interpolation of holes for the sequence "Hiragasy": (a) The mosaic with holes, (b) the mask of the detected holes, (c) the computed bounding box (white) for a hole, (c) the interpolated mosaic, (d) a zoom on region determined by the bounding box of (c), a zoom on the same region in the interpolated mosaic (d).

Here, we propose interpolating missing pixel values from the pixels in the neighborhood of a hole. Despite such an interpolation is nowadays subject of various research works on texture filling and sophisticated methods have been proposed [11], we will use a conventional least square estimation. Indeed, the holes are usually not very large, thus a linear model taking into account the possible gradient of luminance can be sufficient. Thus, the missing pixel values can be selected on a plane. For each color component, the value of a pixel $\mathbf{p} = (x, y)$ in a hole will be approximated as:

$$M(x, y, c) = \xi_c \cdot x + \upsilon_c \cdot x + \chi_c, \quad c = R, G, B \tag{3.79}$$

where $\xi_c, \upsilon_c, \chi_c$ are the parameters of the plane.

To chose the available pixel values for the estimation of the plane parameters, we determine the bounding box of a hole as illustrated in Figure 3.25(c). Figure 3.25(e) shows a

zoom on the region determined by the bounding box. These pixels are used to interpolate the missing pixels.

The least square estimation is formulated to estimate the parameter vector $\mathbf{P} = (\xi_c, \upsilon_c, \chi_c)^{\mathrm{T}}$ as:

$$\mathbf{P} = \left(\mathbf{H}_{\mathrm{P}}\mathbf{H}_{\mathrm{P}}^{\mathrm{T}}\right)^{-1}\mathbf{H}_{\mathrm{P}}^{\mathrm{T}}\mathbf{Z}_{\mathrm{P}} \tag{3.80}$$

Denoting V as the number of valid pixels in the bounding box, the matrices are constructed as follows:

$\mathbf{Z}_{\mathrm{P}}$ is the $V \times 1$ vector of the measures which are here the values of one colour components of the valid pixels in the bounding box:

$$\mathbf{Z}_{\mathrm{P}} = \begin{pmatrix} M(x_1, y_1, c) \\ \vdots \\ M(x_V, y_V, c) \end{pmatrix} \tag{3.81}$$

$\mathbf{H}_{\mathrm{P}}$ is the $3 \times V$ observation matrix containing the pixel indices of the valid pixels in the bounding box corresponding to $\mathbf{Z}_{\mathrm{P}}$:

$$\mathbf{H}_{\mathrm{P}} = \begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_V & y_V & 1 \end{pmatrix} \tag{3.82}$$

Finally, the missing pixels values can be predicted as the values on the plane. The interpolated mosaic is shown in Figure 3.25(d) and a zoom on the interpolated region is shown in Figure 3.25(f).

Median Filtering for Artefact Reduction in the Background

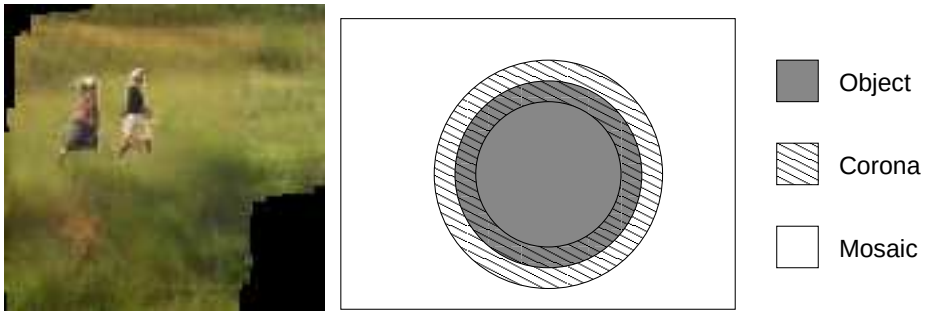
The dilation of the object masks may not be sufficient to entirely exclude an object in case of a very inaccurate segmentation. Thus, seams can appear at the borders of the background masks where objects has been excluded from the blending. We propose applying a spatial median filter in order to remove these seams on the background mask borders. However, a simple median filter can create false colors as the color components are treated separately. In order to avoid this, the median filter can be replaced by a vector median as proposed in [27].

To determine the regions where the median filter has to be applied, we compute for each object a corona and combine the warped coronas in a map similar to that one shown in Figure 3.24(d). In contrast to Figure 3.23(c), we enlarge the corona to filter inside the region of removed object as well as illustrated in Figure 3.26(b). The coronas are obtained by subtracting the dilated object mask from the eroded object mask. We then apply in the indicated regions a 3×3 spatial median filter on each color component.

Mean Filtering for Object Insertion

As shown in Figure 3.26(a) the insertion of an object in the mosaic can cause seams at the borders of the inserted object. In order to smooth these seams and thus to obtain a more

realistic insertion of the object, we propose applying a 3×3 mean filter in the corona of the inserted object as illustrated in Figure 3.26(b).



(a) (b)

Figure 3.26: Mean filtering of the mosaic: (a) Mosaic without mean filtering on the border of the inserted object for the sequence "Hiragasy", (b) corona of an object for filtering.

3.3 Results

In this section we present some results obtained with the proposed mosaicing method. The DC images of the test sequences are shown in Figure 3.27. Here, we suppose that the shot changes have already been detected in the video. We will present a method for the shot change detection on MPEG-1/2 compressed video in Chapter 7. Then, for each shot a mosaic can be constructed giving a global view of the scene with the representative objects inserted. These mosaics provide an approximate information on the scene content, the foreground objects in the scene, and the motion of the camera. We evaluate visually the quality of the mosaics with regard to the dynamic scene content.

The first sequence, "Tympanon", represents a static scene with a tilt up and a small zoom in of the camera. Thus, no object extraction, and no postprocessing has been applied for the mosaic construction. The mosaic is shown in Figure 3.28(a).

The second sequence, "Comportements1", represent a scene with one moving object and a small pan left of the camera. In this case, we apply the full method (see Figure 3.7). The resulting mosaic is shown in Figure 3.28(b).

The third sequence, "Hiragasy", represents a complex scene with a camera tracking two persons. As shown in Figure 3.23(a) some objects are not accurately segmented, but due to the dilation of the object masks the artefacts in the background mosaic are reduced (see Figure 3.24(b)). Finally, the remaining artefacts are eliminated by the median filter in the postprocessing step. The insertion of the object is also satisfying, the borders observable in Figure 3.26(a) has been removed by the mean filtering. In addition, holes have been interpolated as shown in Figure 3.25. We state that the overall results is visually satisfying.

The computational times for these sequences are shown in Table 3.1. They were obtained on an Intel Pentium 4 3.00GHz processor using a non optimised C++ code and the VXL image library [200]. These computational times are very attractive. The videos are encoded at

(a) Tympanon

(b) Comportements1

(c) Hiragasy

Figure 3.27: DC image test sequences: (a) A shot extracted from the documentary "La joueuse de Tympanon" CERIMES-SFRS®, (b) a shot extracted from the documentary "Comportements alimentaires des hommes préhistoiriques" CERIMES-SFRS® and the corresponding dilated masks $O_{\mathrm{b}}(k)$, (c) a shot extracted from the documentary "Hiragasy" CERIMES-SFRS® and the corresponding dilated masks $O_{\mathrm{b}}(k)$.

(a) Tympanon (b) Comportements1 (c) Hiragasy

Figure 3.28: Mosaics for the sequences of Figure 3.27 CERIMES-SFRS®.

30 fps with a GOP size of 15 frames. Then a sequence of 12 I-frames e.g. the "Tympanon" has a duration of 6s. Hence, our computational times are almost 3 times real time which allows to create the video summary in a fast way. Unfortunately, the computations are not real time. This is mainly due to the non-optimised data extraction. For instance, the motion estimation for the sequence "Tympanon" takes only 0.615s. This means that the data extraction takes 6.551s using the MSSG decoder [121], but this step can be optimised.

Another time consuming task in the mosaic construction is the computation of the global geometric transformations (see Equation (3.61)). According to this equation, for each pixel the motion models between the current image and the reference image have to be concatenated. Hence, for long sequences, the computational time for the concatenation will take a significant ratio of the mosaic construction time. Therefore, there is a difference of circa 8s for the computational time (2) between the sequence "Comportements1" which consists of 6 frames and the other two sequences, "Tympanon" and "Hiragasy", which consists of 12 frames.

Sequence	No. I-frames	(1)	(2)	(3)
Tympanon	12	7.166s	13.545s	20.711s
Comportements1	6	6.161s	3.114s	9.275s
Hiragasy	12	8.326s	10.022s	18.348s

Table 3.1: Computational times for the sequences of Figure 3.27: (1) the computational time for data extraction, motion estimation and object extraction, (2) the computational time for concatenation of the motion models in the geometric transformations, illumination correction, blending and postprocessing, (3) the total computational time.

The mosaics shown in Figure 3.28 represent the scene content in an appropriate way

without artefacts due to illumination changes or inaccurate object segmentations, but they may seem of too low resolution for the user. Additionally, they suffer from blur and aliasing artefacts due to the nature of DC images. Therefore in the next chapter, we present a method to increase the resolution and to improve the visual quality of the mosaic in terms of blur and aliasing.

3.4 Conclusion

In this chapter we presented a framework for the construction of low-resolution mosaics from MPEG-1/2 compressed video. The proposed method gives a *complete solution* to various problems which can be encountered in real video: inaccuracy of motion vectors in the compressed stream, illumination changes, the presence of moving objects and difficulties in their segmentation. The principal points of this method are: a deep study and elaborated solution for the computation of geometric transformations for image registration. Here, remaining in the framework of recursive approach, we propose a solution to limit error propagation and inaccuracy; a full solution for blending, including the seamless removal and insertion of objects, luminance correction and filling holes.

The principal steps of the method are: First, DC images of I-frames, DC images of P-frames, DC images of the motion compensation error of P-frames, and motion vectors of P-frames are extracted from the compressed stream. Second, the global camera motion is estimated from P-frames motion vectors. Nevertheless, this is not sufficient to obtain a complete motion trajectory in the sequence. Therefore, the motion parameters are extrapolated in a GOP to predict the motion parameters for I-frames. It happens that the encoded motion vectors are inaccurate. We detect the concerned frame based on the DC images of the motion compensation error of P-frames and propose a method either to correct the motion model or to reestimate the motion model on the DC images of P-frames. Here, we stress that such a thorough study of motion from compressed streams with its refinement has not been proposed in literature to our knowledge. Then, the motion models are concatenated in a geometric transform for each DC image of the sequence. Third, the estimated global camera motion is then used in a moving object detection method where it is combined with a color segmentation of the DC images of I-frames. Fourth, the estimated global camera motion is used in a robust illumination correction method for the DC images of I-frames. Therein, we exclude the detected moving objects from the computation as they cause estimation errors. Fifth, the illumination corrected DC images of I-frames are blended into the background mosaic. Moving objects are excluded from the blending using the result of the moving object detection. Our mosaic construction allows not only the removal of moving objects, but also the reinsertion of objects. At least, post-processing is performed to close holes and remove eventual seams in the mosaic.

The proposed method is almost fully automatic, except the choice of the best segmented objects to be inserted in the mosaic and thus the reference frame for the registration of the image sequence. This is in focus of future work.

The time performance is promising. We obtain almost three times real time which allows

the fast construction of a video summary. Moreover, these computational times can still be reduced by optimising computations such as the data extraction or the concatenation of motion models.

Nevertheless, the obtained mosaics presented in this chapter are of very low resolution and are degraded by blur and aliasing artefacts due to the nature of DC images. To visualise the scene content the user would prefer a higher resolution and a better visual quality. Therefore, we will present in the next chapter a super-resolution method for the mosaic blending. This allows to increase the resolution of the mosaic and to improve the visual quality.

Chapter 4

Super-Resolution

The easiest way to create an image with a higher resolution is a simple interpolation for example a bilinear interpolation or a bicubic interpolation. Digital images are degraded by aliasing and a loss of high frequencies. Therefore, the construction of an image with a higher resolution just by interpolation will not produce a satisfying result.

One solution to increase spatial resolution is to reduce the pixel size of the imaging system. However, as the pixel size increases the amount of light available decreases. Noise is generated that degrades the image quality severely. Hence, there exists the limitation of the pixel size reduction. The current sensor technology has almost reached this level.

Super-resolution reconstruction is a promising signal processing approach to overcome the inherent resolution limitations of low-resolution imaging systems. An high-resolution image (or sequence), the so called super-resolution image (or sequence), is obtained from a sequence of observed low-resolution images. The major advantage of this approach is that it may be cost less and the existing imaging systems can be still used.

Super-resolution reconstruction is used in a wide range of applications where multiple frames of the same scene can be obtained. One application is to reconstruct high quality video stills for printing or frame freeze. A typical single frame in video signal is generally of poor quality and is not suitable e.g. for hard-copy printout. Synthetic zooming of a region of interest is another important application in surveillance, forensic, medical, and satellite imaging. For forensic or surveillance purposes it is often needed to magnify objects in the scene such as faces or the licence plate of a car. In biomedical imaging most of the acquisition systems (MRI, CT, ultrasound, microscopes ...) are limited in resolution quality and higher resolution images are helpful for a doctor to make a correct diagnosis [94, 53]. Super-resolution can be used in satellite imaging applications such as remote sensing or LANDSAT, to improve the resolution of a target considered. Another application is the standard conversion of the video signal e.g. from NTSC to HDTV. Super-resolution has also been applied

for demosaicking of a sensor [92, 48], to reduce artefacts of compressed video [57], and to deinterlace video [136]. As we mentioned in Section 3.1, super-resolution reconstruction can be used for mosaic blending as well.

The premise for super-resolution reconstruction is a sequence of low-resolution images captured from the same scene, shifted with sub-pixel precision. If the low-resolution images are shifted by integer units, then each image contains the same information, and thus there is no new information that can be used for super-resolution reconstruction. To obtain different looks at the same scene, some relative scene motion must exist from frame to frame. The scene motions can occur due e.g. to the motion of the camera or the movement of local objects. If these motions are known or can be estimated with sub-pixel precision, the super-resolution reconstruction is possible by combining these low-resolution images as shown in Figure 4.1.

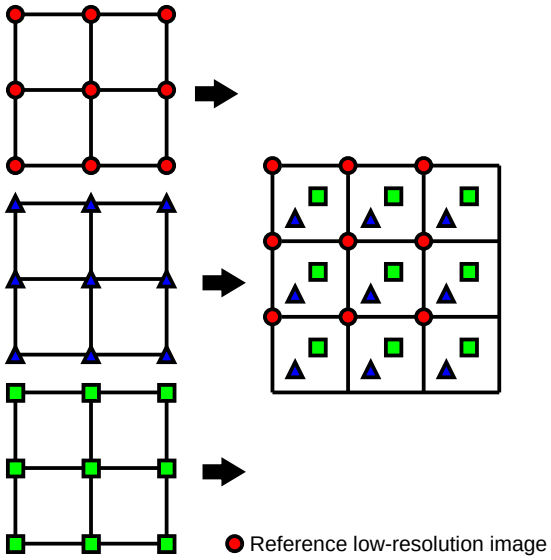


Figure 4.1: Principle of super-resolution: At the left three low-resolution images captured from the same scene with sub-pixel shifts that can be combined in the super-resolution image at the right.

During the acquisition of a digital image, there is a loss of spatial resolution caused by optical distortions (out of focus, diffraction limit,...), motion blur due to the limited shutter speed, noise that occurs within the sensor, and the pixel size in the sensor. Thus, the digital low-resolution image suffers from blur, noise and aliasing artefacts. Therefore, the super-resolution algorithm covers restoration techniques to produce a high quality image from noisy, blurred and aliased images.

In Chapter 3, we presented a complete framework to construct mosaics from the MPEG-1/2 compressed stream (see Figure 3.7). The resulting mosaics are of very low-resolution and are degraded by blur and aliasing artefacts. Hence, in this chapter we propose a super-resolution algorithm for the blending of the mosaic (step five in Figure 3.7). The objective is to increase the resolution of the mosaic and to improve its visual quality. This

requires an additional steps for the mosaic construction which is the estimation of the blur in the acquired images.

This chapter is organised as follows. We review in Section 4.1 common super-resolution reconstruction techniques for raw and compressed video, and also methods for the estimation of blur. Then, we present in Section 4.2 our super-resolution method for mosaic blending. In this section, we develop a new method for the estimation of blur and its appropriate restoration. Section 4.3 shows some results of the proposed method. We conclude the proposed method in Section 4.4.

4.1 State of the Art

The several super-resolution reconstruction methods presented in literature differ in the type of reconstruction method, the assumed observation model which relates the low-resolution images with the super-resolution image, the domain (spatial or frequency) in which the algorithm is applied, the acquisition of the low-resolution images, the estimation of the blur and the sub-pixel motion. In the following, we give a brief overview of existing super-resolution reconstruction methods for raw and compressed video and focus then on blur estimation techniques. Borman and Stevenson [13, 14, 15] and Park et al. [133] have already presented some good reviews on super-resolution techniques for raw video. A review on super-resolution techniques for compressed video is given by Segall et al. [164].

4.1.1 Modelling of the Imaging Process

Considering the desired super-resolution image F, an observation model can be formulated relating each low-resolution image G to the super-resolution image through the imaging process. There are several ways to model the imaging process. In general three kinds of imaging operations are considered: warping, blurring, and decimating. There are different assumptions on the order of them. One is that warping is followed by blurring and decimating as illustrated in Figure 4.2(a). For instance, this kind of model has been used by Irani and Peleg [75], Zhao and Sawhney [214], Hardie et al. [63], and Baker and Kanade [7], Elad and Feuer [41]. The second assumption is that blurring is followed by warping and decimating as illustrated in Figure 4.2(b). This kind of model has been used by Patti and Altunbasak [135], Ur and Gross [197], and Chiang and Boult [30]. Both models are physical as the blur may come after warp when it is dominated by the camera blur, but it may also occur before the warp if the atmosphere is the major blurring source [202]. According to Wang and Qi [202] the first model coincides with the imaging blurring process when the camera blur is dominant, but it is usable only if the motion of the super-resolution sequence is known a-priori. When the motion has to be estimated from the low-resolution images, using this model may cause a systematic error. The use of the second model is more appropriate, and leads to better performance. They show that if imaging blur is spatio-temporally shift-invariant, and image motion purely translational both models have no distinction.

Although, there are some different considerations on the observation model, these mod-

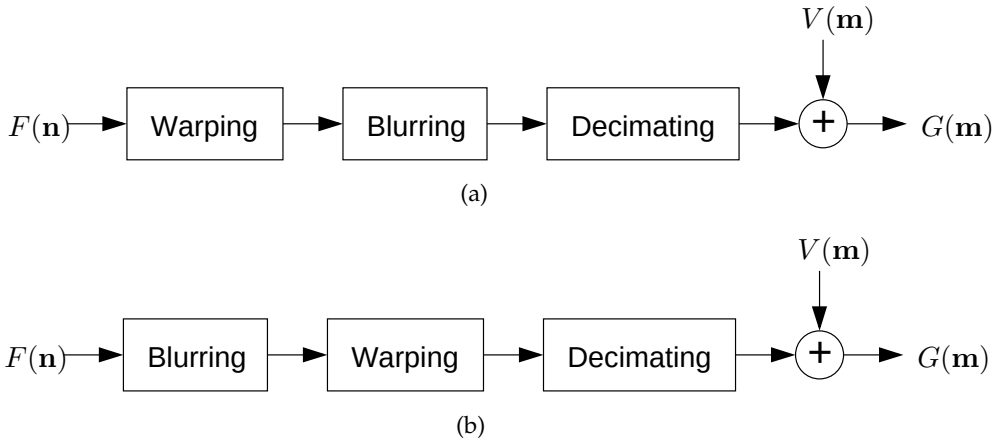


Figure 4.2: Two kinds of imaging assumptions for super-resolution reconstruction [202]: (a) The warping-blurring model, (b) the blurring-warping model.

els can be unified in a simple matrix vector form as the low-resolution pixels are defined as a weighted sum of the related super-resolution pixels. Assuming that each blurred image is corrupted by additive noise, the observation model can be represented without loss of generality as [133]:

$$\mathbf{G}(k) = \mathrm{H}(k)\mathbf{F} + \mathbf{V}(k) \tag{4.1}$$

where $\mathbf{G}(k)$ is the vector of kth LR image, $\mathbf{F}$ the vector of the desired HR image, $\mathrm{H}(k)$ the matrix of the weights representing blurring, motion and subsampling, and $\mathbf{V}(k)$ is the vector of stochastic uncorrelated noise. Then, the pixels of each low-resolution image are related to the pixels in the super-resolution through the observation model (4.1). Inverting the problem for the reconstruction of $\mathbf{F}$ we obtain an ill-posed inverse problem as the resulting set of equation is undetermined. The problem is that a multiplicity of possible solutions exist given a set of low-resolution observations. A solution is to constrain the solution space according to a-priori knowledge on the super-resolution image. Hence, we describe in the following the main super-resolution techniques with respect to this general observation model (4.1) and for certain methods we will develop the observation model.

4.1.2 Frequency Domain Methods

Frequency domain methods represent a major class of super-resolution methods. They are based on the shifting property of the Fourier transform, the aliasing relationship between the continuous and the discrete Fourier transform, and the fact that the signal is band-limited. Using these properties, aliased discrete Fourier coefficients are related to samples of the continuous Fourier transform. The super-resolution image is then recovered by an inverse discrete Fourier transform.

The frequency domain method of Tsai and Huang [194] is the seminal work for super-resolution reconstruction. They derived a system of equations that describes the relationship

between low-resolution images and the desired super-resolution image by using the relative motion between the low-resolution images. Let $F(x, y)$ denote the continuous super-resolution image and $\mathfrak{F}(u, v)$ be its continuous Fourier transform. Then, $F(x, y)$ is shifted by global translations:

$$F(x, y, k) = F(n_x + \Delta_{xk}, n_y + \Delta_{y_k}) \tag{4.2}$$

where Δ_{xk} and Δ_{y_k} are the components of translation in x and y-direction, respectively, and $1 < k < K$. The shifted images are then impulse sampled yielding K observed images $F_k(m_x, m_y)$ with $m_x \in \{0, 1, \dots M_x\}$ and $m_y \in \{0, 1, \dots M_y\}$:

$$F_k(m_x, m_y) = F(m_x T_x + \Delta_{xk}, m_y T_y + \Delta_{y_k}) \tag{4.3}$$

where T_x and T_x are the sampling periods in x and y-direction. The continuous Fourier transform of the scene $\mathfrak{F}(u, v)$ and the discrete Fourier transforms of the shifted and sampled images $\mathfrak{F}_k(\mathfrak{m}_x, \mathfrak{m}_y)$ are related via aliasing [14]:

$$\mathfrak{F}_k(\mathfrak{m}_x, \mathfrak{m}_y) = \frac{1}{T_x T_y} \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} \mathfrak{F}_k \left(\frac{2\pi \mathfrak{m}_x}{T_x M_x} + \frac{2\pi m_x}{T_x}, \frac{2\pi \mathfrak{m}_y}{T_y M_y} + \frac{2\pi m_y}{T_y} \right) \tag{4.4}$$

Then, using the shifting property of the continuous Fourier transform [14]:

$$\mathfrak{F}_k(u, v) = \exp\left(2\pi i(\Delta_{xk} u + \Delta_{y_k} v)\right) \mathfrak{F}(u, v) \tag{4.5}$$

If $F(x, y)$ is bandlimited, there exist L_x, L_x such that $|F(x, y)| = 0$ for $|x| \geq (L_x \pi / T_x), |y| \geq (L_y \pi / T_y)$. Assuming that $F(x, y)$ is bandlimited, the shifting property (4.5) can be used to rewrite the aliasing relationship (4.5):

$$\mathfrak{F}_k(\mathfrak{m}_x, \mathfrak{m}_y) = \frac{1}{T_x T_y} \sum_{m_x=0}^{L_x} \sum_{m_y=0}^{L_y} \exp\left(2\pi i \left[\Delta_{xk} \left(\frac{2\pi \mathfrak{m}_x}{T_x M_x} + \frac{2\pi m_x}{T_x} \right) + \Delta_{y_k} \left(\frac{2\pi \mathfrak{m}_y}{T_y M_y} + \frac{2\pi m_y}{T_y} \right) \right] \right) \cdot \mathfrak{F} \left(\frac{2\pi \mathfrak{m}_x}{T_x M_x} + \frac{2\pi m_x}{T_x}, \frac{2\pi \mathfrak{m}_y}{T_y M_y} + \frac{2\pi m_y}{T_y} \right) \tag{4.6}$$

This relationship can be written in matrix form [14]:

$$\mathbf{G} = \Phi \mathbf{F} \tag{4.7}$$

where $\mathbf{G}$ is a $K \times 1$ column vector with the kth element of the discrete Fourier Transform coefficients of $\mathfrak{F}_k(\mathfrak{m}_x, \mathfrak{m}_y)$ of the observed images $F_k(n_x, n_y)$, $\mathbf{F}$ is a $L_x L_y \times 1$ column vector with the samples of the unknown continuous Fourier Transform $\mathfrak{F}(u, v)$, Φ is a $K \times L_1 L_2$ matrix which relates the discrete Fourier Transform of the observed low-resolution images to the samples of the continuous super-resolution image. Thus, the reconstruction of the desired super-resolution image requires to determine Φ by solving the system of equations for $\mathbf{F}$ and then using the the inverse discrete Fourier Transform to obtain the reconstructed image. Since the matrix Φ requires knowledge of the translation parameters $\Delta_{xk}, \Delta_{y_k}$, these parameters are estimated before reconstruction.

Several extensions of the method [194] have been proposed. Tekalp et al. [189] include sensor blur and observation noise. Then, the equivalent of Equation (4.6) is solved using a least square approach. A recursive least squares solution for Equation (4.6) was proposed by Kim et al. [87] where all the low-resolution images are assumed to have the same blur and the same noise characteristics. Kim and Su [88] further developed the earlier work [87] to consider different blurs and for each low-resolution image. Tikhonov regularisation is adopted to overcome the ill-posed problem resulting from zeros in the blur operator. A discrete cosine transform based method was proposed by Rhee and Kang [151]. By replacing the discrete Fourier transform through a discrete cosine transform computationally efficient reconstruction is achieved. They also adopt regularisation to compensate for ill-posedness in the case of insufficient subpixel information or inaccurate motion information.

The advantage of frequency domain methods is their theoretical simplicity and low computational complexity. A drawback is that they are limited to the linear shift invariant case i.e. motion, blur and downsampling are space invariant. This means for motion that only global translational motion is allowed. In addition, it is difficult to include spatial domain a-priori knowledge for regularisation due to the lack of data correlation between the spatial and frequency domain.

4.1.3 Regularised Approach

Generally, super-resolution reconstruction is an ill-posed problem because of an insufficient number of low-resolution images and ill-conditioned blur operators. Therefore, the regularised approach adopts procedures to stabilize the inversion of the ill-posed inverse problem. Traditionally, regularisation has been described from both the algebraic and the statistical perspectives. In both cases, regularisation takes the form of constraints on the space of possible solutions often independent of the measured data.

Deterministic Approach

The deterministic approach accomplishes regularisation by adding penalty terms in the cost function. For example this can be in form of a Lagrangian $J(\lambda)$ [8]. The problem is then formulated to find:

$$\min \left[J(\lambda) = \sum_{k=1}^{K} |\mathbf{G}(k) - \mathrm{H}(k)\mathbf{F}|^2 + \lambda \|\mathbf{C}\mathbf{F}\|^2 \right] \tag{4.8}$$

where $\mathbf{G}(k)$ is the vector of kth LR image, $\mathbf{F}$ the vector of the desired HR image, and the matrix $\mathrm{H}(k)$ represents upsampling, blur, and warping. The first term $\sum_{k=1}^{K} |\mathbf{G}(k) - \mathrm{H}(k)\mathbf{F}|^2$ of (4.8) expresses the fidelity of the super-resolution image to the observed data. The second term $\lambda \|\mathbf{C}\mathbf{F}\|^2$ is regularisation term where the operator $\mathbf{C}$ is generally a high pass filter. This is essentially a smoothness constraint which suggests that most images are naturally smooth with limited high-frequency activity, and thus it is appropriate to minimize the amount of high-pass energy in the restored image. λ represents the Lagrange multiplier, referred to as the regularisation parameter, that controls the trade-off between fidelity to the data and

smoothness on the solution. Choosing λ could be done manually or automatically e.g. by the cross-validation method of Nguyen et al. [125]. Larger values of λ will lead to a smoother solution. This is useful when only a small number of low-resolution images are available (the problem is underdetermined) or the fidelity of the observed data is low due to registration errors and noise. On the other hand, if a large number of low-resolution images are available and there is only less noise, a small λ will lead to a good solution. The minimisation of (4.8) leads to the following iteration [133]:

$$\mathbf{F}^{i+1} = \mathbf{F}^{i} + \lambda_2 \left[\sum_{k=1}^{K} \mathrm{H}^{\mathrm{T}}(k) \left(\mathbf{G}(k) - \mathrm{H}(k)\mathbf{F}^{i} \right) - \lambda \mathbf{C}^{\mathrm{T}} \mathbf{C} \mathbf{F}^{i} \right] \tag{4.9}$$

where λ_2 represents the convergence parameter.

Hong et al. [70] proposed a multichannel regularised super-resolution method in which the regularisation functional is used to calculate the regularisation parameter without any prior knowledge at each iteration step. In [64], Hardie et al. define an observation model which incorporates knowledge of the optical system and the sensor blur. The high-resolution image is formed by minimising the regularised cost function which is based on the observation model. Farsiu et al [47, 48] propose an alternate fidelity term based on the L_1 norm. They also propose a robust regularisation term which provides robust performance while persevering the edges. Shechtman et al. [167, 168] developed a space-time super-resolution method by combining several input sequences of different space-time resolution. This method allows a reduction of spatial artefacts such as motion blur by increasing the temporal resolution. A steerable space-time regularisation term is introduced in the super-resolution reconstruction which applies smoothness only in directions within the space-time volume where the derivatives are low.

Stochastic Approach

From a statistical perspective, regularisation is incorporated as a-priori knowledge about the solution. Thus, using a maximum a-posteriori (MAP) estimator a much richer class of regularisation functions emerges.

Given the observation model in Equation (4.1) the MAP approach seeks the estimate $\hat{F}$ for which the a-posteriori probability $\Pr\left(F|G(k)\right)$ is maximum:

$$\hat{F} = \arg\max_{F} \Pr(F|G(1), \ldots, G(K)) \tag{4.10}$$

where F is the unknown super-resolution image and $G(K)$ the low-resolution observations. Applying Bayes' rule yields:

$$\hat{F} = \arg\max_{F} \left[\frac{\Pr(G(1), \ldots, G(K)|F) \Pr(F)}{\Pr(G(1), \ldots, G(K))} \right] \tag{4.11}$$

Since the maximum $\hat{F}$ is independent of $G(1), \ldots, G(K)$ [14]:

$$\hat{F} = \arg\max_{F} \left[\Pr(G(1), \ldots, G(K)|F) \Pr(F) \right] \tag{4.12}$$

Taking the logarithmic function, the MAP optimisation problem can be expressed as [14]:

$$F = \arg\max \{\ln \Pr(G(1), \ldots, G(K)|F) + \ln \Pr(F)\} \tag{4.13}$$

The term $\ln \Pr(G(1), \ldots, G(K)|F)$ is the log-likelihood function and $\Pr(F)$ the a-priori density of F. Since Equation (4.1), the log-likelihood function is determined by the pdf of the noise. The specific form of $\ln \Pr(F)$ depends on the prior being used. In [21] a good overview on common image priors is given. Often Markov random field (MRF) priors are used as the prior term $\Pr(F)$. Thus $\Pr(F)$ can be expressed by a Gibbs distribution which has the form [133]:

$$\Pr(F) = \frac{1}{Z} \exp\left(-U(F)\right) = \frac{1}{Z} \exp\left(-\sum_{c \in S} \varphi_c(F)\right) \tag{4.14}$$

where Z is a normalising constant, U is an energy function, φ_c is a potential function that depends only on the pixel values located with in the clique c, and S denotes the set of cliques. By defining φ_c as a function of the image derivative, U measures the cost cause by the irregularities of the solution. Commonly, an image is assumed to be globally smooth which is incorporated into the estimation problem through a Gaussian prior. With the Gaussian prior, the potential function takes quadratic form $\varphi_c(F) = (D^{(n)}F)^2$ where $D^{(n)}$ is the nth order difference. This penalises the high-frequency components severely and as a result the solution becomes oversmoothed. However, if a potential function is modelled which less penalises the large difference in F, an edge-preserving super-resolution image can be obtained.

If the error between frames is assumed to be independent and noise is assumed to an independent identically distributed zero mean Gaussian distribution, the optimisation problem can be expressed as:

$$F = \arg\min \left[\sum_{k=1}^{K} \|G(k) - \mathrm{H}(k)F\|^2 + \alpha \sum_{c \in S} \varphi_c(F)\right] \tag{4.15}$$

where α is the regularisation parameter. It can be shown that Equation (4.8) is equal to a MAP estimate if a Gaussian prior is used in Equation (4.15) [133].

The super-resolution reconstruction from a low-resolution image sequence using a MAP method was proposed by Schultz and Stevenson [159, 160]. They extended their earlier work on Bayesian MAP image interpolation for improved definition to the problem of super-resolution image reconstruction. Assume that K odd low-resolution frames are observed, $G(m_x, m_y, k)$ with $k \in \left\{r - \frac{K-1}{2}, \cdots, r, \cdots, r + \frac{K-1}{2}\right\}$ and $m_x \in \{1, 2, \cdots, M_x\}$, $m_y \in \{1, 2, \cdots, M_y\}$. The objective is to reconstruct a super-resolution image $F(n_x, n_y, r)$ with $n_x \in \{1, 2, \cdots, \varsigma M_x\}$, $n_y \in \{1, 2, \cdots, \varsigma M_y\}$ and $\varsigma \in \mathbb{Z}$, coincident with $G(m_x, m_y, r)$ the center frame in the observed image sequence. The subsampling model for center frame which models the spatial integration of light intensity over the sensor in the detector array is:

$$G(m_x, m_y, r) = \frac{1}{\varsigma^2} \sum_{n_x = \varsigma m_x - \varsigma + 1}^{\varsigma m_x} \sum_{n_x = \varsigma m_y - \varsigma + 1}^{\varsigma m_y} F(n_x, n_y, r) \tag{4.16}$$

This relationship can be expressed in matrix notation as:

$$\mathbf{G}(r) = \mathrm{H}(r)\mathbf{F}(r) \tag{4.17}$$

where $\mathrm{H}(r) \in \mathbb{R}^{M_x M_y \times \varsigma^2 M_x M_y}$ is the subsampling matrix relating the super-resolution image $\mathbf{F}(r)$ with the observed frame $\mathbf{G}(r)$. The remaining observed images $\mathbf{G}(k)$, $k \neq r$, are related to $\mathbf{F}(r)$ through motion-compensated subsampling matrices which model the subsampling of the high-resolution frame and account for object motion between frames:

$$\mathbf{G}(k) = \mathrm{H}(k)\mathbf{F}(r) + \mathbf{u}(k) \tag{4.18}$$

where $\mathrm{H}(k) \in \mathbb{R}^{M_x M_y \times \varsigma^2 M_x M_y}$ is the motion-compensated subsampling which relates the kth low-resolution image to the super-resolution image $\mathbf{F}(r)$. The vector $\mathbf{u}(k)$ contains pixels which are not present in $\mathbf{F}(r)$, but in $\mathbf{F}(k)$ due to object motion. The elements of $\mathbf{u}(k)$ are not known as $\mathbf{F}(k)$ is not known. Only the rows of $\mathrm{H}(k)$ for which elements of $\mathbf{G}(k)$ are observed entirely from motion-compensated elements of $\mathbf{F}(r)$ contain useful information. Using only the rows of useful information, the set of equations can be reduced:

$$\mathbf{G}'(k) = \mathrm{H}'(k)\mathbf{F}(r) \tag{4.19}$$

In practice $\mathrm{H}'(k)$ is estimated from the observed low-resolution frames $\mathbf{G}(k)$ and $\mathbf{G}(r)$ by hierarchical block matching. This results in:

$$\mathbf{G}'(k) = \mathrm{H}'(k)\mathbf{F}(r) + \mathbf{V}(k) \tag{4.20}$$

where $\mathbf{V}(k)$ represent the error in estimation $\mathrm{H}'(k)$.

By assuming a Huber Markov random field for the prior term which is a Gibbs prior similar to (4.14) and independent identically distributed Gaussian density to represent the error in estimating the observation model, the MAP estimate of the high-resolution image given the low-resolution images becomes:

$$\hat{F}(r) = \arg \min_{F(r) \in C} \left\{ \sum_{x,y} \sum_{1}^{4} \varphi_c(D^{(2)}(x,y)F) + \sum_{\substack{k=r-\frac{K-1}{2} \\ k \neq r}}^{r+\frac{K-1}{2}} \lambda(k) \left\| G'(k) - \mathrm{H}'(k)F(r) \right\|^2 \right\} \tag{4.21}$$

subject to

$$C = \left\{ F(r) : G(r) = \mathrm{H}'(r)F(r) \right\} \tag{4.22}$$

The first term is related to the image model. This prior models piece-wise smooth data and thus preserves edges. The likelihood of an edge is controlled by the Huber edge penalty function on the second order difference $D^{(2)}F$. $D^{(2)}F$ has small values in smooth image regions and high values at edges. The Huber edge penalty function is defined as:

$$\varphi_c(x) = \begin{cases} x^2 & \text{if } |x| \leq \alpha \\ 2\alpha |x| - \alpha^2 & \text{if } |x| > \alpha \end{cases} \tag{4.23}$$

where α is a threshold controlling the size of discontinuities modelled by the prior. The second term corresponds to the independent identically distributed Gaussian density used for the observation model error. In (4.21) each frame has an associated confidence parameter $\lambda(k)$ which represents the confidence in $\mathrm{H}'(k)$. To estimate $\hat{F}(r)$ the gradient projection method is used. Note that this method includes a constraint set C and is thus not entirely a MAP method.

Although we do not follow this approach in this research, we find a a common point in the regularisation where discontinuities i.e. image contours are taken into account. Generally speaking, MAP super-resolution methods have been intensively studied and we refer the reader to [63, 103, 69, 7, 205].

The major advantages of the Bayesian methods are the robustness and flexibility in modelling noise characteristics and a-priori knowledge about the solution. Assuming that the noise process is white Gaussian, a MAP estimation with convex energy functions in the priors ensures the uniqueness of the solution. Efficient gradient descend methods can be used to estimate the super-resolution image.

Maximum likelihood (ML) estimation [190, 24] is a special case of MAP estimation where no prior term is used. Since the use of a-priori information is essential for the solution of ill-posed inverse problems, MAP estimation is usually used in preference to ML.

4.1.4 Set Theoretic Approach

Set theoretic methods, especially the method of projection onto convex sets (POCS), utilise a spatial domain model and allow the simple inclusion of a-priori information. Therefore, convex sets $\mathcal{C}_i$ are defined representing desirable super-resolution image characteristics such as data consistency, positivity, smoothness etc. The super-resolution image is supposed to be a member of these sets.

A set $\mathcal{C}_i$ is convex if the following conditions are satisfied [19]:

$$\mathcal{C}_i \neq \emptyset \tag{4.24}$$

$$(\forall \alpha \in]0,1[),\ (\forall (X,Y) \in \mathcal{C}_i \times \mathcal{C}_i) : \alpha X + (1-\alpha)Y \in \mathcal{C}_i \tag{4.25}$$

For example, the positivity constraint on the super-resolution image is represented by [14]:

$$\mathcal{C}_+ = \{F : F(\mathbf{p}) > 0\ \forall \mathbf{p}\} \tag{4.26}$$

If the constraint sets have a non empty intersection, then a solution that belongs to the intersection $\mathcal{C}_s = \cap_{i=1}^{m} \mathcal{C}_i$, can be found by alternating projections onto these convex sets. Therefore, for each convex set a projection operator P_i is defined which projects the estimate of the super-resolution image onto the surface of the convex set $\mathcal{C}_i$. Then, iteratively projecting onto the constraint sets the method converges to a solution onto the surface of the intersection of the constraint sets:

$$F^{i+1} = P_m.P_{m-1} \cdots P_2.P_1.F^i \tag{4.27}$$

where F^0 is an arbitrary starting point.

The POCS formulation of super-resolution reconstruction was first suggested by Stark and Oskoui [178]. Their method was extended by Tekalp et al. [189] to include observation noise. Therefore, a data consistency constrained set (4.1) is defined for each pixel in the low-resolution images $G(\mathbf{m}, k)$:

$$\mathcal{C}_{\delta_0}(\mathbf{m}, k) = \{F(\mathbf{n}) : |\mathbf{r}(\mathbf{m}, k)| \leq \delta_0\} \tag{4.28}$$

with

$$\mathbf{r}(\mathbf{m}, k) = G(\mathbf{m}, k) - \sum_{\mathbf{n}} F(\mathbf{n}) \mathrm{H}(\mathbf{m}, \mathbf{n}, k) \tag{4.29}$$

where $\mathrm{H}(k)$ includes the effects of the sensor blur and δ_0 is a bound reflecting the statistical confidence in the observation. The projection of $F(\mathbf{n})$ onto $\mathcal{C}_{\delta_0}(\mathbf{m}, k)$ is defined as [189]:

$$P(F(\mathbf{n})) = F(\mathbf{n}) + \begin{cases} \frac{(\mathbf{r}(\mathbf{m},k)-\delta_0)}{\sum_{\mathbf{p}} \mathrm{H}(\mathbf{m},\mathbf{p},k)^2} \cdot \mathrm{H}(\mathbf{m}, \mathbf{n}, k) & \mathbf{r}(\mathbf{m}, k) > \delta_0 \\ 0 & -\delta_0 \leq \mathbf{r}(\mathbf{m}, k) \leq \delta_0 \\ \frac{(\mathbf{r}(\mathbf{m},k)+\delta_0)}{\sum_{\mathbf{p}} \mathrm{H}(\mathbf{m},\mathbf{p},k)^2} \cdot \mathrm{H}(\mathbf{m}, \mathbf{n}, k) & \mathbf{r}(\mathbf{m}, k) < -\delta_0 \end{cases} \tag{4.30}$$

Patti et al. [137] developed a POCS super-resolution method to consider space varying blur, non-zero aperture time, non-zero physical dimensions of each individual sensor element, sensor noise and arbitrary sampling lattices. Eren et al. [43] extended this method to the case of multiple moving objects in the scene. To do this, they introduced the concepts of validity map and segmentation map. The validity map disables projections based on observations with inaccurate motion estimates, while the segmentation map enables object-based processing where more accurate motion models can be used to improve the quality of the reconstructed image. In [135], Patti and Altunbasak propose a POCS-based super-resolution method where the discretisation of a continuous image formation model is improved to allow for higher order interpolation methods. They also modify the constraint sets to reduce the ringing artefacts near edges.

An alternative approach to POCS was proposed by Tom and Katsaggelos [191] where an ellipsoid is used to bound the constraint set. The centroid of the ellipsoid is taken as the super-resolution estimate.

The advantage of POCS based methods is that it is simple and uses a flexible spatial domain observation model. It allows the powerful inclusion of a priori information. Drawbacks are the non-uniqueness of the solution, the dependence of the solution on the initial guess, the slow convergence and high computational costs.

4.1.5 Iterative Backprojections

Simulate and correct methods start from a given estimate of the super-resolution reconstruction and a model of the imaging process. The imaging process is simulated using the super-resolution estimate as the input to produce a set of simulated low-resolution observation images. These images are compared with the actual observation, an error is computed and

backprojected onto the super-resolution estimate by inverting the imaging process. This is then used to correct the estimate of the super-resolution image. The process is iterated until convergence or at least until a stopping criterion is fulfilled.

For a given observation model there is freedom in the choice of the mechanism for back-projecting the error to correct the super-resolution estimate, as well as the error function to be minimised. Consider the observation model relating K low-resolution observation images in Equation (4.1). Given F^i an estimate of the super-resolution reconstruction, it is possible to simulate the low-resolution observation images by applying (4.1) to the super-resolution estimate F^i as:

$$G^i(k) = \mathrm{H}(k).F^i \tag{4.31}$$

The projection operator $\mathrm{H}(k)$ comprises blurring, motion and subsampling.

Then, the super-resolution estimate is updated by backprojecting the error between the simulated low-resolution images $G^i(k)$ and the observed low-resolution images $G(k)$ via the backprojection operator H^{BP} [13]:

$$F^{i+1} = F^i + \mathrm{H}^{BP}\left(G(k) - G^i(k)\right) \tag{4.32}$$

$$= F^i + \mathrm{H}^{BP}\left(G(k) - \mathrm{H}(k).F^i\right) \tag{4.33}$$

Typically H^{BP} is designed to approximate the inverse of the projection operator H. Its choice affects the characteristics of the solution when there are possible solutions. Therefore, H^{BP} may be utilised as an additional constrained which represents the desired property of the solution. The different approaches vary in the choice of the projection and the backprojection operators.

In [41, 21] is shown that the configuration of Equation (4.32) is a simple error relaxation algorithm such as steepest descent which minimises a quadratic error. This means that the iterative backprojection method is similar to a ML (or least squares) method without regularisation.

The initial idea of iterative backprojections was formulated by Peleg et al. [141] where globally translated images of a static scene are considered. This method was then taken up by Keren et al. [84] and further developed to a more general global translation and rotation motion model.

Given the set of low-resolution images $G(k)$, the super-resolution image F can be reconstructed from the low-resolution sequence. Assuming that the imaging process of the observed low-resolution images is known, it is possible to simulate the imaging process with the estimate F^i as an approximation to the original scene yielding the simulated low-resolution images $G^i(k)$. Denoting $\mathbf{n}$ a pixel in the super-resolution image and $\mathbf{m}$ the low-resolution pixel, the error between the simulated images at the ith iteration and the original observed images to be minimised in the iterative scheme is defined as [141]:

$$\epsilon^i = \sum_{k=1}^{K} \sum_{\mathbf{m}} \left| G^i(\mathbf{m}, k) - G(\mathbf{m}, k) \right| \tag{4.34}$$

Then, each pixel $\mathbf{n}$ in the current estimate F^i is examined. Assuming that the grey level at the position $\mathbf{n}$ is $F^i(\mathbf{n}) = l$, then the modifications $F^i(\mathbf{n}) = l - 1$ and $F^i(\mathbf{n}) = l + 1$ are

considered. For each of the three pixel values $\{l-1, l, l+1\}$ the corresponding simulated low-resolution images and the corresponding errors (4.34) are computed. Then, the pixel value that minimises the error is assigned to $F^i(\mathbf{n})$. The process is repeated for each pixel in the image over several iterations.

The work of [141, 84] was improved in [74]. Irani and Peleg propose an iterative backprojection method similar to that one used in computer aided tomography. The error functional (4.34) was modified in:

$$\epsilon^i = \sqrt{\sum_{k=1}^{K} \sum_{\mathbf{m}} \|G(\mathbf{m},k) - G^i(\mathbf{m},k)\|^2} \tag{4.35}$$

First, motion is estimated which is assumed to be a combination of translation and rotation. Then, for each super-resolution pixel $F(\mathbf{n})$ all low-resolution pixels determined through the imaging process which depend on its value are identified. These values are used to update the super-resolution pixel. The iterative scheme can be expressed as [133]:

$$F^i(\mathbf{n}) = F^{i-1}(\mathbf{n}) + \sum_{\mathbf{m}\in\Upsilon(\mathbf{m},k)} \mathrm{H}^{BP}(\mathbf{n},\mathbf{m}) \left(G(\mathbf{m},k) - G^i(\mathbf{m},k)\right) \tag{4.36}$$

where $\Upsilon(\mathbf{m},k)$ is receptive field determined by the set $\{\mathbf{m} \in G(k) \,|\, \mathbf{m} \text{ is influenced by } \mathbf{n} \in F\}$ and H^{BP} is the backprojection kernel which determines the contribution of the error $\left(G(\mathbf{m},k) - G^i(\mathbf{m},k)\right)$ to the super-resolution estimate $F^i(\mathbf{n})$.

The same authors extended their work in [75] to improve the resolution of differently moving objects. The method segments an image into region of homogeneous motion, the objects, and tracks them along the sequence. Each object is then super-resolved separately. The imaging process, yielding the observed image sequence $G(k)$ is modelled by:

$$G(\mathbf{m},k) = \mathrm{S}^{-1}\left[\mathrm{B} * \left[\mathrm{T}^{-1}(k).F(\mathbf{n})\right] + V(k)\right] \tag{4.37}$$

where $G(k)$ is the observed object in the kth frame, S^{-1} is the downsampling operator and $\mathrm{T}^{-1}(k)$ the geometric transformation from F to $G(k)$ determined by the 2D motion parameters of the tracked object, B is the blurring operator of the sensor, $*$ is the the convolution operator, and $V(k)$ is an additive noise term. The receptive field Υ of a detector whose output is the pixel $G(\mathbf{m})$ is defined by its center $\mathbf{n}$ and its shape. The shape is determined by the region of support of the blurring operator B and by the geometric transformation T. Similarly the center $\mathbf{n}$ is obtained by $\mathrm{T}(\mathbf{m},k)$.

The algorithm starts with an initial guess of the super-resolution image F^0:

$$F^0 = \frac{1}{K}\sum_{k=1}^{K} \mathrm{T}(k).\mathrm{S}.G(k) \tag{4.38}$$

where S is the upsampling operator, $\mathrm{T}(k)$ is the geometric transformation from $G(k)$ to F inverse to T^{-1}, and K is the number of low-resolution images $G(k)$.

Then, the imaging process is simulated to obtain a set of low-resolution images $G^i(k)$ corresponding to the observed input images $G^i(k)$:

$$G^i(k) = \mathrm{S}^{-1}.\mathrm{B} * \left[\mathrm{T}^{-1}(k).F^i\right] \tag{4.39}$$

The iterative update scheme is expressed by:

$$F^{i+1} = F^i + \frac{1}{K}\sum_{k=1}^{K} \mathrm{T}(k).\mathrm{H}^{BP} * \left[\mathrm{S}.\left[G(k) - G^i(k)\right]\right] \tag{4.40}$$

where H^{BP} is the backprojection kernel determined by B.

Irani and Peleg show that Equation (4.40) converges to the desired solution for 2D affine motion when the following condition is satisfied:

$$\left\|\delta - \mathrm{B} * \mathrm{H}^{BP}\right\| < \frac{1}{\frac{1}{K}\sum_{k=1}^{K}\left\|\mathrm{T}^{-1}(k)\right\|} \tag{4.41}$$

with

$$\left\|\mathrm{T}^{-1}(k)\right\| = \sqrt{\left|\det(\mathbf{A}(k))\right|} \tag{4.42}$$

where δ is the unity pulse function centered at $(0,0)$, and $\mathbf{A}(k)$ is matrix of affine parameters of T^{-1} (see Equation 3.59). The smaller $\left\|\delta - \mathrm{B} * \mathrm{H}^{BP}\right\|$ is, the faster the algorithm converges. Ideally, if $\left\|\delta - \mathrm{B} * \mathrm{H}^{BP}\right\| = 0$, then the algorithm converges in one single iteration. This means that H^{BP} is the inverse kernel of B, which may not exist, or which may numerically be unstable to compute. Permitting $\left\|\delta - \mathrm{B} * \mathrm{H}^{BP}\right\| > 0$, allows H^{BP} other than the exact inverse of B, and therefore increases the numerical instability, but slows down the speed of convergence. It is demonstrated that given the condition (4.41) the algorithm converges at exponential rate regardless of the choice of initial guess.

The works of Irani and Peleg [74, 75] are the basis for several iterative backprojection super-resolution methods. Mann and Picard [106] extended the approach to a perspective motion model, and Zhao and Sawhney [214] to optical flow. Later, Zhao [212] proposes introducing an illumination model to recover a high-resolution object shape under different illumination angles. Dekeyser et al. [35] backwarp a spatially interpolated error image instead of computing the receptive field of the low-resolution pixels and modify the update of the super-resolution estimate which is updated for each frame and not after one pass of the sequence. Zomet and Peleg [215] derive a modified version of iterative backprojections using a specific blur kernel for each low-resolution observation and forward warping in the backprojection stage. The authors further developed their method in [216], where the sum of images in the update was replaced by a median to improve the robustness of the algorithm. Greenspan et al. [53] extend the iterative backprojections from 2D to 3D in order to compute isotropic 3D MRI images from 2D MRI slices.

The advantage of iterative backprojections is that it is easily to understand. This method enforces that the super-resolution reconstruction matches the observed data. Normally, these algorithms are convergent, but they do not necessarily converge to a unique solution and may be dependent on the order of pixel updates or the initial guess. In contrast to POCS and the regularised approach, it is difficult to apply a-priori constraints.

This bibliographical survey is not exhaustive. We can also cite the interpolation from non-uniformly spaced samples approach [3, 189, 197, 169, 124], the hybrid approach [160, 41], or the adaptive filtering approach [138, 40, 42].

4.1.6 Compressed Domain Approach

Super-resolving compressed video provides an additional challenge with respect to traditional super-resolution methods as other types of observations are available. Video compression methods represent images with a sequence of motion vectors and transform coefficients. The motion vectors provide a noisy observation of the temporal relationship within the high-resolution scene. The transform coefficients represent a noisy observation of the high-resolution intensities. This noise results from more sophisticated processing than the transitional processing scenario, as compression techniques discard data according to perceptual significance. Typically, super-resolution methods are based on traditional reconstruction methods such as MAP [28, 54, 58, 57, 163, 109, 165, 132], POCS [55, 134], iterative backprojections [213], least squares estimation [4], regularised least squares estimation [108], and interpolation [107], but novel processing methods are adopted. A review on super-resolution methods for compressed video presented in a Bayesian framework is given in [164].

The use of compressed video requires an extended observation model as motion compensation, block-based DCT, and quantisation may be modelled in addition to the traditional imaging process of the camera. Segall et al. [164] define the observation model for compressed video as follows. Denoting $F(x, y, t)$ the high-resolution time-varying scene in the image plane coordinate system, where x, y and t indicate horizontal, vertical and temporal locations. The scene is filtered and sampled during acquisition to obtain the discrete sequence $G(m_x, m_y, k)$ where k is an integer time index, $1 \leq m_x \leq M_x, 1 \leq m_y \leq M_y$. The frames of the sequence $G(m_x, m_y, k)$ are then compressed with a video compression algorithm resulting in the sequence $G_{\mathrm{c}}(m_x, m_y, k)$. Additionally, motion vectors $\mathbf{d}(m_x, m_y, k, r) = (d_x(m_x, k, r), d_y(m_y, k, r))^{\mathrm{T}}$ are provided that predict a pixel in $G(m_x, m_y, k)$ from the previously transmitted $G(m_x, m_y, r)$. The low-resolution image $G(m_x, m_y, k)$ is related to the high-resolution image $F(n_x, n_y, k)$ by:

$$G(m_x, m_y, k) = \mathrm{B.S}^{-1}.F(n_x, n_y, k) \tag{4.43}$$

where B describes the filtering of the high-resolution image and S^{-1} the downsampling. The operators B and S^{-1} model the acquisition system and are assumed to be known. Note, that the detector noise is not taken into account.

Frames within the high-resolution sequence are also related through time. A translational relationship between the frames is assumed:

$$F(n_x, n_y, k) = F(n_x + d_x(n_x, k, r), n_y + d_y(n_y, k, r), r) + V(n_x, n_y) \tag{4.44}$$

where $V(n_x, n_y)$ account for errors within the model. Noise introduced by the sensor can also be incorporated into this error term.

During compression the frames are divided into blocks that are encoded either directly transform-coded using a DCT and quantised, or predicted from a reference frame and the prediction error is transform-coded and quantised. Hence, the acquired low-resolution

frame $G(k)$ and its compressed observation $G_{\mathrm{c}}(r)$ are related as:

$$G_{\mathrm{c}}(k) = \mathrm{DCT}^{-1}\left[\mathbf{Q}\left[\mathrm{DCT}\left[G(k) - MC(G_{\mathrm{c}}(r), \mathbf{d}(k,r), k)\right]\right] + MC(G_{\mathrm{c}}(r), \mathbf{d}(k,r), k)\right] \quad (4.45)$$

where $\mathbf{Q}$ represents the quantisation procedure, DCT and DCT^{-1} are the forward and inverse transform operations, $MC(G_{\mathrm{c}}(r), \mathbf{d}(k))$ is the motion-compensated prediction of $G(k)$ formed by motion compensating previously decoded frame(s), and $G_{\mathrm{c}}(r)$ and $\mathbf{d}(r)$ denote the set of decoded frames and motion vectors that predict $G_{\mathrm{c}}(k)$.

Combining (4.43), (4.44), and (4.45) the relationship between the high-resolution frames and the low-resolution observations is:

$$G_{\mathrm{c}}(m_x, m_y, k) = \mathrm{B.S}^{-1}.F(n_x + d_x(n_x, k, r), n_y + d_y(n_y, k, r), r) + V_e(m_x, m_y, k) \quad (4.46)$$

where $V_e(m_x, m_y, k)$ includes the errors introduced during compression, registration, and acquisition.

In the literature, two types of models for the quantisation noise have been proposed. The first is formulated in the transform domain following the fact that quantisation errors are bounded by the quantisation scale factor as:

$$-\frac{S(i)}{2} \leq (\mathrm{DCT}[G_{\mathrm{c}}(k)])(i) - (\mathrm{DCT}[G(k)])(i) \leq \frac{S(i)}{2} \quad (4.47)$$

where $(\mathrm{DCT}[G_{\mathrm{c}}(k)])(i)$ denotes the ith transform coefficient of the compressed frame, $(\mathrm{DCT}[G(k)])(i)$ the ith transform coefficient of the acquired frame, and $S(i)$ is the quantisation scale factor for the coefficient i. It seems reasonable that the recovered high-resolution image has transform coefficients within the same interval. This quantisation constraint is used e.g. by Altunbasak et al. [5], Gunturk et al. [54], Patti and Altunbasak [134].

The second model for quantisation noise is constructed in the spatial domain where the quantisation noise is modelled by a Gaussian distribution. This approximation for quantisation noise appears e.g. in the work of Chen and Schultz [28], Gunturk et al. [57], Mateos et al. [108, 109], Segall et al. [163, 165].

The iterative backprojection method of Zhao [212] used a traditional observation model of the imaging process, but proposes a dynamic masking method in order to limit the influence of regions to the super-resolution process that are highly degraded due to the compression error. In [107], Martins and Forchhammer proposed an interpolation technique which takes into account the encoded picture type and the quantisation step size of the motion-compensated low-resolution pixels.

Some of these methods are formulated in the spatial domain which requires the full decoding of the compressed sequence before applying the super-resolution algorithm [28, 4, 57, 213, 107, 108, 132, 163, 165]. Other methods are directly formulated in the DCT domain relating the super-resolution image with the quantised DCT coefficients [134, 55, 54, 56, 58, 5]. Thus, DCT coefficients are extracted from the compressed flow as low-resolution observations.

Most of these methods consider the MPEG motion compensation vectors as too inaccurate for further use. Except the methods presented in [28, 108, 109, 165]. Chen and Schultz [28] used the encoded motion vectors as an initial estimate in a block-matching algorithm. The so estimated motion vectors are constrained to be within a region surrounding the actual subpixel displacement. In [108], Mateos et al. combined the upsampled encoded motion vectors with the motion vectors estimated from the high resolution image. The objective is to improve the estimated motion vectors by forcing them to be close to the one provided by the encoder. Later, the same authors [109] propose a Bayesian framework to simultaneously estimate the high-resolution frames from compressed low-resolution video and the motion relating the high- and low-resolution frames using the motion vectors provided by the encoder. To do this, the information on the upsampled encoded motion vectors is included in the prior for estimating the high resolution motion vectors field. Segall et al. [165] estimate motion using the encoded motion vectors in a Bayesian framework. They model the accuracy of the encoded motion vectors based on the encoded error residual.

4.1.7 Blur Estimation

As showed in Section 4.1.1 three kinds of imaging operators are considered: warping, blurring, and downsampling. The downsampling operator is straightforward and we realised it by a bilinear interpolation. We presented a method to compute the warping operator in the previous chapter (see Section 3.2.2). Thus, the last challenge is the estimation of the blur operators.

Typically, a blur function in the spatial domain is described by the *point spread function* (PSF) [93]. This function describes the impulse response to a point light source at the sensor of the camera. The PSF can be spatially varying. The difficulty in solving the restoration problem with a spatially varying blur commonly motivates the use of a stationary model for the blur. This leads to the following discrete model for a linear degradation caused by blur and additive noise:

$$\begin{aligned}\tilde{I}(x,y) &= \sum_{m=1}^{M}\sum_{n=1}^{N} \mathrm{B}(x-m, y-n) I(m,n) + V(x,y) \\ &= \mathrm{B}(x,y) * I(x,y) + V(x,y) \end{aligned} \tag{4.48}$$

where $I(x,y)$ is the ideal image of the size $M \times N$, $\tilde{I}(x,y)$ is the degraded image, $\mathrm{B}(x,y)$ is the PSF, $*$ is the 2D convolution operator and $V(x,y)$ represents the additive noise. The noise is usually assumed to be a zero mean Gaussian distributed, as this effectively models noise in many different imaging scenarios. The use of linear techniques for solving the restoration problem is facilitated by using this shift-invariant model (4.48). We will focus on the spatially varying case in the next chapter.

If B is known or estimated, then the direct way to restore the ideal image by linear techniques would be to synthesise B^{-1} and convolve with the left hand side of (4.48). Obviously, B^{-1} in not directly feasible in spatial domain and also the noise would be amplified. Furthermore, such a convolution operation is computationally demanding. In order to reduce

the complexity of calculations it is often preferred to work in the frequency domain since the convolution transforms to a multiplication of spectra:

$$\tilde{\mathfrak{I}}(u,v) = \mathfrak{B}(u,v)\mathfrak{I}(u,v) + \mathfrak{v}(u,v) \tag{4.49}$$

where $\tilde{\mathfrak{I}}, \mathfrak{I}, \mathfrak{B}, \mathfrak{v}$ are the the Fourier transforms of $\tilde{I}, I, \mathrm{B}, V$, respectively, and u, v are the coordinates in the transform domain.

The frequency response of the PSF, $\mathfrak{B}$, is called the *optical transfer function* (OTF). It can be decomposed into the amplitude part, $|\mathfrak{B}|$, which is the *modulation transfer function* (MTF) and the phase part which is the *phase transfer function* (PTF). The most frequently used models to represent shift-invariant image degradations are [8]:

- *Linear motion blur* is a common result of camera panning or fast object motion. It is represented by a 1D uniform local averaging of neighbouring pixels in the direction of motion:

$$\mathrm{B}(\nu) = \begin{cases} \frac{1}{L} & \text{if } -\frac{L}{2} \leq \nu \leq \frac{L}{2} \\ 0 & \text{otherwise} \end{cases} \tag{4.50}$$

  where ν is the directional variable, and L is the length of the blur.

- *Atmospheric turbulence blur* is common in remote sensing and aerial imaging. The blur due to log-term exposure though the atmosphere can be modelled by a 2D Gaussian:

$$\mathrm{B}(x,y) = C \exp\left(-\frac{1}{2}\left(\frac{x^2+y^2}{\sigma^2}\right)\right) \tag{4.51}$$

  where C is a normalising constant ensuring that the blur is of unit volume, and σ^2 is the variance that determines the severity of the blur. A 2D Gaussian blur can also be used to model the PSF of the camera sensor [75].

- *Uniform out-of-focus blur* is primarily due to effects at the cameras aperture that result in a spreading of a point of incoming light across a circle of confusion. A complete model of the camera's focusing system depends on many parameters such as the focal length, the camera aperture size and length, the distance between the object and the camera, the wavelength of incoming light, and the effects due to diffraction. The knowledge of these parameters is often not available. When the blur due to poor focusing is large, the PSF can be approximated as a uniform intensity distribution within a circular disk of the radius R:

$$\mathrm{B}(x,y) = \begin{cases} \frac{1}{\pi R^2} & \text{if } \sqrt{x^2+y^2} \leq R \\ 0 & \text{otherwise} \end{cases} \tag{4.52}$$

- *Uniform 2D blur* is a more severe form of degradation that approximates an out-of-focus blur:

$$\mathrm{B}(x,y) = \begin{cases} \frac{1}{L^2} & \text{if } -\frac{L}{2} \leq x,y \leq \frac{L}{2} \\ 0 & \text{otherwise} \end{cases} \tag{4.53}$$

Most of the super-resolution methods presented above are of restricted scope as they suppose that the PSF is known, but often no knowledge about the imaging system is available. It is clear that these methods are not suitable for real image processing situations. *Blind super-resolution* addresses this problem, but it is still an open challenge [47]. The objective of blind super-resolution is to estimate the unknown PSF parameters from the measured data. Many single-frame blind deconvolution algorithms have been suggested [96, 97], but only few have been incorporated in super-resolution methods. There remains a need for more research to provide a super-resolution method along with a more general blur estimation algorithm from aliased images [47]. Nguyen et al. [126] propose a blind super-resolution method where the PSF and regularisation parameters are computed using the generalised cross-validation method. In [68], He et al. proposed a MAP framework for joint estimation of the blur and the super-resolution image. Both are iteratively estimated in a cyclic procedure using two different regularisation parameters, one for the blur operator and one for the super-resolution image. In [69], He et al. extend this work to parametric blur models. The best fit blur model and parameters are determined based on a learning set of different blur types.

In this PhD thesis we are interested in restoring motion blur caused by the motion of the camera. The particular problem of motion blur in video is that it varies from frame to frame i.e. the blur direction and its size can vary in each image as they depends on the velocity of camera motion, the direction of the camera motion, and the aperture time of the camera. There are two main approaches to blind deconvolution [96, 97]:

- Identifying the PSF separately from the true image, in order to use it later with one of the known classical image restoration methods. Estimating the PSF and the true image are disjoint procedures. This approach leads to computationally simple algorithms.

- Incorporating the identification procedure with the restoration algorithm. This merge involves simultaneously estimating the PSF and the true image, which leads to the development of more complex algorithms.

Methods of the first class tend to be computationally simpler and are therefore in our interest. Recently several deblurring methods taking into account motion blur have been presented. For instance, Rav-Acha and Peleg [148] compute the motion direction by performing one iteration of the algorithm for each angle and the angle which gives the strongest deblurring effect is the angle of the motion blur. Then, they estimate simultaneously the motion parameters and the size of the blur kernel using a multiresolution Gaussian pyramid. In [209], Yitzhaky and Kopeika measure the motion direction as the angle where the minimum of the total intensity of the image derivative is attained. In order to determine the blur extent, they compute the autocorrelation function of the spatial image derivative in motion direction. Then, the blur extent corresponds to the distance of the minimum of the autocorrelation function and its zero-shift center.

Contrary to these methods, in a super-resolution framework where motion has to be estimated for the alignment and warping of the frames the direction of the blur can be obtained from this estimated motion. This will be our approach.

4.2 Super-Resolution Mosaic Construction

In this section, we present our contribution to super-resolution mosaicing. The objective here is to increase the resolution and to improve the visual quality of DC-resolution mosaics. The method for their construction has been presented in Chapter 3. Therefore, we apply a super-resolution algorithm to these mosaics. We consider the global scheme illustrated in Figure 4.3 for the super-resolution mosaic construction. Taking into account our new goal, i.e. improving the resolution of mosaics, this scheme differs from that one presented in the previous chapter (Figure 3.7). First, in this chapter we do not consider moving objects as they require special processing. This will be the focus of the next chapter. Consequently, the moving object extraction and postprocessing steps are omitted. As we explained above super-resolution methods incorporate restoration techniques. This requires the estimation of the blur in the input image sequence. Hence, blur estimation is an additional step in Figure 4.3. Blending is now accomplished in the super-resolution step. The steps of data extraction, registration and illumination correction have already been described in Chapter 3. Hence, in the following of this chapter we will address the remaining open issues for the super-resolution mosaic construction: the estimation of the blur, and the super-resolution algorithm.

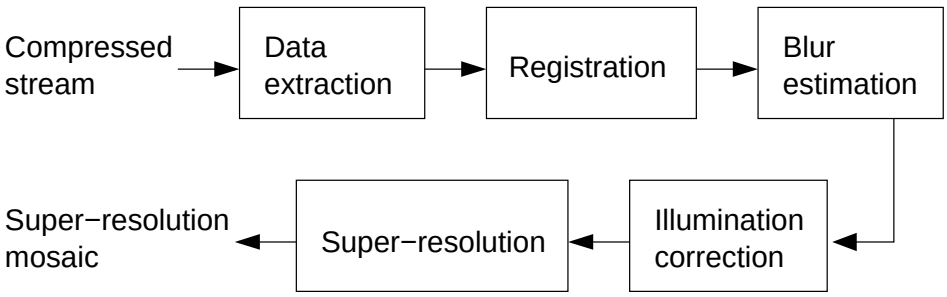


Figure 4.3: Global scheme of the super-resolution mosaic construction.

We chose the iterative backprojection algorithm from Irani and Peleg [75] as reference method because of its simplicity and fast convergence. We described this method in Section 4.1.5. This method was formulated for moving objects i.e. regions of homogeneous motion. Here, we assume only one region of homogeneous motion which is the entire image. We first derive our observation model from that one proposed in [75] and then reformulate the iterative backprojection algorithm.

4.2.1 Modelling of the Imaging Process

Let $\mathbf{n} = (n_x, n_y)$ be a pixel in the target super-resolution image F and $\mathbf{m} = (m_x, m_y)$ a pixel in the available LR image G. In [75], Irani and Peleg model the imaging process as:

$$G(k) = \mathrm{S}^{-1}\mathrm{B} * \left[\mathrm{T}^{-1}(k)\left[F + V(k)\right]\right] \tag{4.54}$$

where S^{-1} is the downsampling operator, B is the PSF of the sensor, $*$ is the convolution operator, $\mathrm{T}^{-1}(k)$ the geometric transformation from the super-resolution image to the kth

low-resolution image, and V the additive noise. Irani and Peleg assumed a sensor PSF which is constant for all observations. In contrast to this, we assume linear motion blur which can vary in each image of the sequence. Thus, we reformulate the observation model as:

$$G(k) = \mathrm{S}^{-1}.\mathrm{B}(k) * [\mathrm{T}(k).[F + V(k)]] \tag{4.55}$$

where $\mathrm{B}(k)$ is the linear motion PSF of kth frame. Note that F refers in our case to a mosaic. Having specified our observation model, we now can derive the super-resolution algorithm.

4.2.2 Iterative Backprojections

Based on the observation model (4.55), the low-resolution sequence is simulated at each iteration i using the current guess of the super-resolution mosaic F^i:

$$G^i(k) = \mathrm{S}^{-1}.\mathrm{B}(k) * \left[\mathrm{T}^{-1}(k).F^i(k)\right] \tag{4.56}$$

The result is a set of simulated low-resolution images $G^i(k)$ corresponding to the sequence of the original low-resolution images $G(k)$. Then, the difference between the simulated low-resolution images $G^i(k)$ and the input images $G(k)$ is used to update the super-resolution mosaic:

$$F^{i+1} = F^i + \mu(K) \sum_{k=1}^{K} \mathrm{T}(k).\mathrm{R}(k) * \left[\mathrm{S}.\mathrm{A}(k).\left[G(k) - G^i(k)\right]\right] \tag{4.57}$$

where $\mathrm{A}(k)$ is a regularisation operator, $\mathrm{R}(k)$ is the restoration filter defined by $\mathrm{B}(k)$, S is the upsampling operator by the factor ς (inverse to S^{-1}), and $\mu(\mathbf{p}, K) = \frac{1}{|\mathbf{p}|}$ with $|\mathbf{p}|$ as the number of available pixels at position $\mathbf{p}$. We realised the upsampling and downsampling operators, S and S^{-1}, by a bilinear interpolation. Here, we warp a spatially interpolated error image $E(k) = \mathrm{S}.\mathrm{A}(k).\left[G(k) - G^i(k)\right]$ by $\mathrm{T}(k)$ in the process of backprojection similarly to Dekeyser et al. [35] instead of computing the receptive field Υ as proposed by Irani and Peleg (see Equation (4.36)).

The low-resolution observations we use in this work are DC images. They are highly undersampled and thus suffer strongly from aliasing. Therefore, even if the geometric transformations T are accurately estimated, it is sometimes not possible to exactly superimpose edges and textures in the mosaic. This can cause artefacts in the mosaic which are amplified during the iterations of the super-resolution method. Therefore, we add the regularisation operators $\mathrm{A}(k)$ in the backprojection process in order to attenuate the amplification of such artefacts. This is different to [75] (see Equation (4.40)).

We choose the initial guess F^0 similarly to [75] as:

$$F^0 = \mu(K) \sum_{k=1}^{K} \mathrm{T}(k).\mathrm{S}.G(k) \tag{4.58}$$

In this work, we are dealing with RGB color images. Therefore, we apply the equations above to each color component.

In order to determine the maximum number of iterations, we define a stopping criterion based on the error functional ϵ^i (4.35) which is minimised by the iterative backprojections.

In fact, we use the normalised error $\bar{\epsilon}^i$ between the low-resolution observations and its simulated versions at the ith iteration:

$$\begin{aligned}\bar{\epsilon}^i &= \frac{1}{K \cdot M}(\epsilon^i)^2 \\ &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{M}\sum_{\mathbf{m}}\|(G(\mathbf{m},k) - G^i(\mathbf{m},k))\|^2\right)\end{aligned} \tag{4.59}$$

where M is the number of pixels in $G(k)$, and K is the number of frames participating in the mosaic. Note, we use the vector norm $\|\cdot\|$ in the computation of $\bar{\epsilon}^i$, as the R, G and B components are considered in the computation. When $\bar{\epsilon}^i$ approaches stability or increases, the maximum number of iterations has been reached. The value $\bar{\epsilon}^i$ may increase after several iterations for several reasons. On the one hand, it can be due to an inaccurate motion compensation coming from inaccurate motion models or aliasing. On the other hand, it can be due to ringing artefacts induced in the restoration as we will show below.

4.2.3 Blur Estimation

The origin of blur in DC frames is twofold. On the one hand, the blur comes from full-resolution frames due to the well studied phenomenon of motion during the exposure time. To estimate it several methods have been proposed in the literature [59]. On the other hand, the DC frames represent a low-pass version of the full-resolution frames in the video sequence in which the low pass is done locally inside a block of 8×8 pixels. Therefore, the overall blur is a result of these two phenomena. To this end, we propose in this section several blur models to describe the real overall blur function. Moreover, we present a method to estimate the blur parameters from the image sequence itself.

Blur Models

In this work, we focus on three different blur models. We consider:

- *Isotropic Gaussian blur* which models most of the natural blurs.
- *Anisotropic Gaussian blur in motion direction* which models blur due to random motion.
- *Linear Motion blur* which models motion due to fast object motion or to the fast panning of the camera.

We will present in the following, for each blur model, the mathematical expression of the PSF and its frequency response, the OTF, as we will use the OTF in the restoration which is accomplished in the frequency domain. Furthermore, we show that for these blur models, only the amplitude part of the OTF, called the MTF, can be used.

Isotropic Gaussian Blur: This blur models most of the natural blurs. The PSF of such a blur $\mathrm{B}_{\mathrm{Gauss}}$ can be modelled by a 2D Gaussian function:

$$\mathrm{B}_{\mathrm{Gauss2D}}(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2}\left(\frac{x^2+y^2}{\sigma^2}\right)\right) \tag{4.60}$$

The Fourier transform of a Gaussian function is also a Gaussian [203]. Thus, the OTF consists only in an amplitude part which is the MTF. Then, the corresponding MTF $\mathfrak{B}_{\mathrm{Gauss}}$ is:

$$\mathfrak{B}_{\mathrm{Gauss2D}}(u,v,\sigma_{\mathbf{f}}) = \frac{1}{2\pi\sigma_{\mathbf{f}}^2}\exp\left(-\frac{1}{2}\left(\frac{u^2+v^2}{\sigma_{\mathbf{f}}^2}\right)\right) \tag{4.61}$$

where $\sigma_{\mathbf{f}}$ is inversely related to σ. This relation depends on the normalisation factor and will be derived below.

Anisotropic Gaussian Blur in Motion Direction: Here, we describe the mathematical expression corresponding to random motion. The PSF of such a blur $\mathrm{B}_{\mathrm{Gauss}}$ can be modelled by a 2D Gaussian function as:

$$\begin{aligned}\mathrm{B}_{\mathrm{Gauss}}(x,y,\sigma_x,\sigma_y) &= \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right)\\ &= \frac{1}{\sqrt{2\pi}\sigma_x}\exp\left(-\frac{x^2}{2\sigma_x^2}\right)\cdot\frac{1}{\sqrt{2\pi}\sigma_y}\exp\left(-\frac{y^2}{2\sigma_y^2}\right)\\ &= \mathrm{B}_{\mathrm{Gauss}}(x,\sigma_x)\cdot\mathrm{B}_{\mathrm{Gauss}}(y,\sigma_y)\end{aligned} \tag{4.62}$$

As we can notice, the Gaussian PSF can be separated in a horizontal and vertical component, $\mathrm{B}_{\mathrm{Gauss}}(x,\sigma_x)$ and $\mathrm{B}_{\mathrm{Gauss}}(y,\sigma_y)$. Thus, the blurred image is:

$$\tilde{F} = \mathrm{B}_{\mathrm{Gauss}}(x,\sigma_x) * (\mathrm{B}_{\mathrm{Gauss}}(y,\sigma_y) * F) \tag{4.63}$$

For the same reason than above we can only consider the MTF. Thus, the corresponding MTF $\mathfrak{B}_{\mathrm{Gauss}}$ is:

$$\begin{aligned}\mathfrak{B}_{\mathrm{Gauss}}(u,v,\sigma_u,\sigma_v) &= \frac{1}{2\pi\sigma_u\sigma_v}\exp\left(-\frac{1}{2}\left(\frac{u^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right)\right)\\ &= \frac{1}{\sqrt{2\pi}\sigma_u}\exp\left(-\frac{u^2}{2\sigma_u^2}\right)\cdot\frac{1}{\sqrt{2\pi}\sigma_v}\exp\left(-\frac{v^2}{2\sigma_v^2}\right)\\ &= \mathfrak{B}_{\mathrm{Gauss}}(u,\sigma_u)\cdot\mathfrak{B}_{\mathrm{Gauss}}(v,\sigma_v)\end{aligned} \tag{4.64}$$

where σ_u and σ_v are inverse related to σ_x and σ_y, respectively. Similarly to the PSF, the Gaussian MTF can be separated in a horizontal and vertical component, $\mathfrak{B}_{\mathrm{Gauss}}(u,\sigma_u)$ and $\mathfrak{B}_{\mathrm{Gauss}}(u,\sigma_u)$.

Linear Motion Blur: Our objective is to restore directional motion blur that comes from the camera motion in video. We assume a constant velocity of the motion since the exposure time is usually very short.

Let v be the velocity of the motion according to the direction $\mathbf{d}$. Then, the box PSF $\mathrm{B}_{\mathrm{box}}$ in motion direction for linear motion is:

$$\mathrm{B}_{\mathrm{box}}(\nu) = \int_{-T/2}^{T/2}\delta(\nu - vt)\,\mathrm{d}t \tag{4.65}$$

where T is the exposure time and ν is the directional variable.

The corresponding MTF $\mathfrak{B}_{\text{sinc}}$ is [59]:

$$\mathfrak{B}_{\text{sinc}}(\mathbf{f}) = \left| \frac{\sin(\pi v \mathbf{f} T)}{\pi v \mathbf{f} T} \right| = |\text{sinc}(\pi b \mathbf{f})| \tag{4.66}$$

with $b = vT$ as the spatial extent of the blur. We compute $\mathfrak{B}_{\text{sinc}}$ only until the first zero-crossing since a detail of a size smaller than the blur radius b is unresolvable and undergoes black-white phase or colour reversal. Thus, $\mathfrak{B}_{\text{sinc}}$ at frequencies higher than $1/b$ does not exist [93].

Now, let us consider a displacement vector $\mathbf{d}$ developed as $\mathbf{d} = \vec{d_x} + \vec{d_y}$ where $\vec{d_x}$ and $\vec{d_x}$ are the components in xOy orthogonal basis. We suppose that this motion is translational. Let us denote by α the angle with the x-axis (see Figure 4.5). Then, the 1D motion blur in $\mathbf{d}$ direction can be represented as a composition of two motion blurs in horizontal and vertical direction:

$$\text{B}_{\text{box}}(x, y) = \int_{-T/2}^{T/2} \delta\left(x - v\cos(\alpha)t, y - v\sin(\alpha)t\right) \text{d}t \tag{4.67}$$

Here $v\cos(\alpha)$ and $v\sin(\alpha)$ respectively represent the horizontal and vertical components, v_x and v_y, of the velocity v (orthogonal projection). From Equation (4.67) follows:

$$\begin{aligned} \text{B}_{\text{box}}(x, y) &= \int_{-T/2}^{T/2} \delta\left(x - v_x t\right) dt \cdot \int_{-T/2}^{T/2} \delta\left(y - v_y t\right) dt \\ &= \text{B}_{\text{box}}(x) \cdot \text{B}_{\text{box}}(y) \end{aligned} \tag{4.68}$$

For the case of linear motion degradation we can replace the OTF by the MTF. The reason is that here the phase part of the OTF (the PTF) is a linear function and therefore it causes no degradation, but only translation. Thus, only the amplitude part can be taken into account. As the PSF is separable in horizontal and vertical direction, we can do similarly for the MTF:

$$\begin{aligned} \mathfrak{B}_{\text{sinc}}(u, v) &= \left|\text{sinc}\left(\pi b_x u\right)\right| \cdot \left|\text{sinc}\left(\pi b_y v\right)\right| \\ &= \mathfrak{B}_{\text{sinc}}(u) \cdot \mathfrak{B}_{\text{sinc}}(v) \end{aligned} \tag{4.69}$$

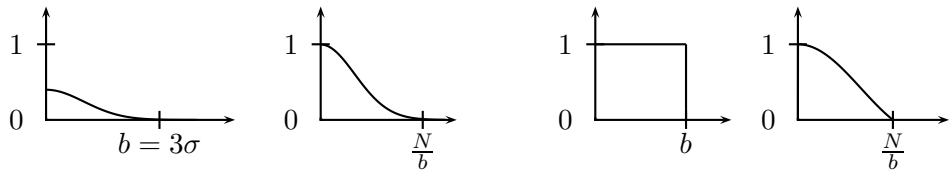
where $b_x = v_x T$ and $b_y = v_y T$. The sinc MTF in motion direction can thus be represented by the combination of a horizontal and a vertical sinc function.

Normalisation

Assuming an image spectrum of the size $2N \times 2N$. (We use rectangular spectra where the size $2N \times 2N$ is obtained by padding which will be precised later). Then, we need to normalize the horizontal and vertical components of the MTFs presented above (see Equations (4.61), (4.64) and (4.69)) with respect to the maximum available frequency $2N$. Here, we treat only the horizontal component of the MTF for the frequencies $[0, N-1]$. The second part is obtained by mirroring the frequencies $[0, N-1]$ to the frequencies $[N, 2N-1]$. The vertical component is obtained similarly and is then transposed.

Let us denote the blur extent in the pixel domain b in motion direction, and b_x, b_y its components in x and y-direction, respectively. In order to obtain σ for the isotropic Gaussian blur from b (and σ_x, σ_y for the anisotropic Gaussian blur from b_x, b_y), we propose to apply the 3σ-property. According to this property, the values within the interval $\pm 3\sigma$ represent 99.73% of the Gaussian distribution. We assume that the values of $\mathrm{B}_{\mathrm{Gauss}}$ outside of the interval $\pm 3\sigma$ are so small that they do not contribute to the blurring process and could be neglected due to the discretisation. Thus, we derive $\sigma = b/3$ (and $\sigma_x = b_x/3, \sigma_y = b_y/3$).

Figures 4.4(a) and (b) show, respectively, the Gaussian in the spatial domain and our normalisation in the frequency domain. Using the assumption of the 3σ-property, the first value equal zero of the PSF appears at $b = 3\sigma$. From this relation we can estimate $\sigma_{\mathbf{f}}$.



(a) (b) (c) (d)

Figure 4.4: The normalisation of the MTFs: (a) The PSF $\mathrm{B}_{\mathrm{Gauss}}$ and (b) the corresponding MTF $\mathfrak{B}_{\mathrm{Gauss}}$ normalised with respect to the size of the spectrum $2N$. (c) The PSF $\mathrm{B}_{\mathrm{box}}$ and (d) the corresponding MTF $\mathfrak{B}_{\mathrm{sinc}}$ normalised with respect to the size of the spectrum $2N$.

Assuming that:

$$b = 3\sigma \tag{4.70}$$

we wish the size of $\mathfrak{B}_{\mathrm{Gauss}}$ to correspond to the frequency N for $b = 1$. As the sigma value in the frequency domain $\sigma_{\mathbf{f}}$ is inversely related to σ:

$$\sigma_{\mathbf{f}} = \frac{C_\sigma}{\sigma} \tag{4.71}$$

we can introduce the normalisation by the relation:

$$\begin{aligned} 3\sigma_{\mathbf{f}} &= N \\ \sigma_{\mathbf{f}} &= \frac{N}{3} \end{aligned} \tag{4.72}$$

By fixing $b = 1$, from (4.70)(4.71) and (4.72) we obtain:

$$\sigma_{\mathbf{f}} = \frac{N}{9\sigma} \tag{4.73}$$

Thus, the first value equal zero in the frequency domain corresponds according to our normalisation to N/b. In order to not modify the frequency $(0, 0)$ in the blurring or restoration process, which would cause a luminance change of the image, we normalise the magnitude of the Gaussian. We divide each value of $\mathfrak{B}_{\mathrm{Gauss}}$ by $\mathfrak{B}_{\mathrm{Gauss}}(0)$.

We normalise the sinc MTF similarly which is shown in Figures 4.4(c) and (d). For a box of the size b, the first zero of $\mathfrak{B}_{\text{sinc}}$ appears at the frequency N/b i.e. for $b = 1$ the size of the sinc corresponds to the frequency N.

Parameter Estimation

The blur can vary in each image, since it depends on the motion direction, on the velocity, and on the exposure time. Therefore, we estimate the blur extent from the image itself. In [59], Hadar et al. proposed the estimation of the blur for translational motion along the x-axis based on the edge response. The authors assumed an underlying ideal edge model as a step function U in the horizontal direction:

$$U(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.74}$$

and measured the response of the imaging system in the image plane. In this work we propose to consider natural edges in the video which are not vertical and measure the edge response in the arbitrary direction of motion. Chiang and Boult [30] also considered a local blur estimation of a Gaussian blur parameter, but in gradient direction.

We first locate the edges by computing the first derivative ∇G of the low-resolution image G using the Sobel operator. Then, a non-maxima suppression and a threshold $\lambda_{\mathcal{E}}$ are applied to the gradient magnitude in order to extract significant edges. We obtain the following set of pixels:

$$\mathcal{E}_{\nabla} = \{\mathbf{m}|\ \mathrm{N}\left(||\nabla G(\mathbf{m})||\right) > \lambda_{\mathcal{E}}\} \tag{4.75}$$

where N is the function of the non-maxima suppression. We chose the threshold $\lambda_{\mathcal{E}}$ with respect to the root mean square of the gradient magnitude $||\nabla G||$:

$$\lambda_{\mathcal{E}} = C_{\mathcal{E}} \cdot \sqrt{\frac{1}{(M_x - 1)(M_y - 1)} \sum_{\substack{1<m_x<M_x-1 \\ 1<m_y<M_y-1}} ||\nabla G(m_x, m_y)||^2} \tag{4.76}$$

where $M_x \times M_y$ is the size of the low-resolution image, and $C_{\mathcal{E}}$ a constant. We determined it experimentally as $C_{\mathcal{E}} = 2$.

We estimate the blur in the motion direction and consider translational motion. This can be justified as locally a motion vector can be interpreted as translational motion. The motion direction $\mathbf{d}$ is known from the motion compensation vector for the geometric transformations T. Thus, our algorithm does not need additional computation in order to determine the direction of the blur.

We measure the edge response only on the edges which are orthogonal to the motion direction. Hence, the complex 2D case of arbitrary oriented motion and non-vertical edges can be considered as a classical model of a horizontal motion and vertical edge described by Equation (4.65) in motion direction. Therefore, we compute the angle between the gradient $\nabla G(\mathbf{m})$ and the motion vector $\mathbf{d}(\mathbf{m})$ and retain only the pixels for which the angle is smaller

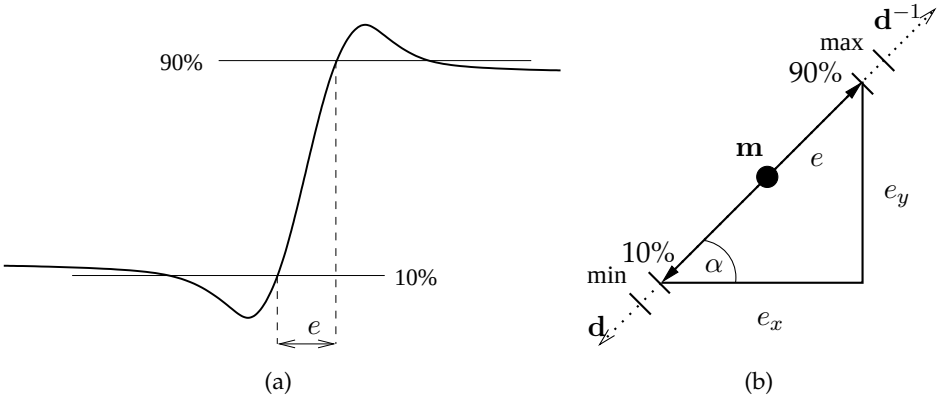


Figure 4.5: (a) Edge response in motion direction of a blurred edge, (b) Computation of the edge response in horizontal and vertical direction.

than the threshold $\lambda_\angle$. We inverted $\mathbf{d}(\mathbf{m})$, if $\angle(\mathbf{d}(\mathbf{m}), \nabla G(\mathbf{m})) > 90^0$ for an easier computation. Then, $\mathbf{d}(\mathbf{m})$ points always in the direction of the local maximum. The resulting pixel set is:

$$\mathcal{E}_{\nabla,\perp} = \left\{ \mathbf{m} \in \mathcal{E}_\nabla | \left| \arccos\left(\frac{\mathbf{d}(\mathbf{m}) \cdot \nabla G(\mathbf{m})}{||\mathbf{d}(\mathbf{m})||\,||\nabla G(\mathbf{m})||} \right) \right| < \lambda_\angle \right\} \tag{4.77}$$

To choose the threshold $\lambda_\angle$, we can start with a small angle value of $\pm 5°$. Nevertheless, if the image does not contain a sufficient amount of edges orthogonal to the motion direction, we increase the threshold. The maximal value we worked with was $\pm 45°$.

For each point of a significant contour we consider a local estimation of the edge response. The edges in a blurred image can be considered as a 1D section in motion direction with form as illustrated in Figure 4.5(a). They are limited by a local minimum and a local maximum determining the rise of the edge. In natural images the local minimum and maximum are caused by artefacts due to the acquisition process, known as Gibbs ringing phenomenon [51]. In order to not take account for these artefacts in our computation, we take the 10% to 90%-distance of the local minimum and maximum [176] as the width of the edge response.

Let $\mathbf{m}$ denote the position in the low-resolution image G which indicates the center of the edge, situated between the local minimum and the local maximum. A discrete straight line segment on a pixel grid according to the direction $\mathbf{d}$ is considered, centered on $\mathbf{m}$. The local maximum and minimum are searched along the segment in the low-resolution image G. Then, the 10% and 90%-limits can be determined using the same method.

The edge response in motion direction $e(\mathbf{m}),\ \mathbf{m} \in \mathcal{E}_{\nabla,\perp}$, is computed as the euclidean distance of the 10% and 90%-limits. Its value can be used for the isotropic Gaussian blur, but for other motion models we need its x and y-components, $e_x(\mathbf{m})$ and $e_y(\mathbf{m})$. They can be

determined using trigonometric triangle rules (see Figure 4.5(b)):

$$\cos(\alpha(\mathbf{m})) = \frac{\mathbf{d}(\mathbf{m}) \cdot (1,0)^T}{||\mathbf{d}(\mathbf{m})||} \tag{4.78}$$

$$\sin(\alpha(\mathbf{m})) = \frac{\mathbf{d}(\mathbf{m}) \cdot (0,1)^T}{||\mathbf{d}(\mathbf{m}))||} \tag{4.79}$$

$$e_x(\mathbf{m}) = |e(\mathbf{m}) \cdot \cos(\alpha(\mathbf{m}))| \tag{4.80}$$

$$e_y(\mathbf{m}) = |e(\mathbf{m}) \cdot \sin(\alpha(\mathbf{m}))| \tag{4.81}$$

As we suppose that this blur is induced by global camera motion, we compute only one "global" value of blur parameters per image. Hence, we compute the edge response for the image as the average of all estimated values:

$$e = \sum_{\mathbf{m} \in \mathcal{E}_{\nabla,\perp}} e(\mathbf{m}) \tag{4.82}$$

$$e_x = \sum_{\mathbf{m} \in \mathcal{E}_{\nabla,\perp}} e_x(\mathbf{m}) \tag{4.83}$$

$$e_y = \sum_{\mathbf{m} \in \mathcal{E}_{\nabla,\perp}} e_y(\mathbf{m}) \tag{4.84}$$

A spatial convolution filter of the size l causes an edge response of the size $2l$. Hence:

$$b = \frac{e}{2} \tag{4.85}$$

$$b_x = \frac{e_x}{2} \tag{4.86}$$

$$b_y = \frac{e_y}{2} \tag{4.87}$$

We have now defined the blur extend in the available data, i.e. the low-resolution frame $G(k)$. Nevertheless, in our observation model (4.55) the blur B is defined for the super-resolution image corresponding to G which necessitates the knowledge of the edge response in the super-resolution image. Here, we suppose that the edge response increases proportionally to the upsampling factor ς of the super-resolution mosaic and define the blur extend is the super-resolution frame as:

$$\hat{b} = \frac{e}{2} \cdot \varsigma \tag{4.88}$$

$$\hat{b}_x = \frac{e_x}{2} \cdot \varsigma \tag{4.89}$$

$$\hat{b}_y = \frac{e_y}{2} \cdot \varsigma \tag{4.90}$$

For the sake of simplicity, we use in the following of this chapter the notations b, b_x, b_y for the blur size in the super-resolution domain.

4.2.4 Restoration in the Frequency Domain

Having determined the blur function and its parameters, we would like to use it to compensate the degradation from motion by synthesising the restoration filter R. The restoration can be done in the pixel domain according to (4.57) with R which is the impulse response of the inverse filter of the PSF B. Contrarily, we can synthesise its transfer function in the Fourier domain using the MTF $\mathfrak{R}$ which is the magnitude part of the Fourier transform of the PSF B. In this section we consider standard restoration filters such as the inverse filter, the pseudo-inverse filter and the optimum filter which is the Wiener filter for the restoration filter $\mathfrak{R}$.

According to the properties of the Fourier transform the convolution in Equation (4.57) transforms into a multiplication in frequency domain. Then, the restoration of the difference image $\mathrm{S.A}(k)\left[G(k) - G^i(k)\right]$ becomes:

$$\mathrm{R}(k) * \mathrm{S.A}(k)\left[G(k) - G^i(k)\right] = \mathrm{FT}^{-1}\left(\mathrm{FT}\left(\mathrm{S.A}(k)\left[G(k) - G^i(k)\right]\right) \cdot \mathfrak{R}(k)\right) \tag{4.91}$$

where FT is the Fourier transform and $\mathfrak{R}(k)$ is the OTF of the restoration filter $\mathrm{R}(k)$.

From the above equation it is clear that the image to be restored has also to be transformed in the frequency domain by a Fourier transform. Implementations of the Fourier transform are typically optimised for a images of the size $2^j \times 2^j, j \in \mathbb{N}$. Another aspect is that the Fourier transform assumes a periodic signal. Thus, artefacts can appear on the image borders when applying an inverse Fourier transform. Therefore, we pad the images by a mirror to the nearest size $2^j \times 2^j$ as illustrated in Figure 4.6. The padded image is then used in the restoration and afterwards cropped to the original images size.

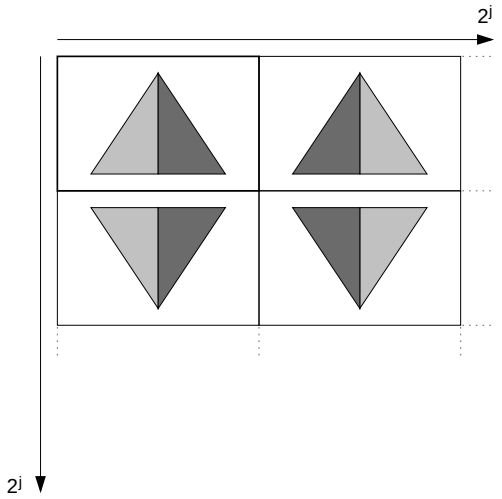


Figure 4.6: Padding of an image to the size $2^j \times 2^j$ by a mirror.

Pseudo-Inverse Filter

When the MTF is available, the simplest way to restore an image is inverse filtering. When synthesising an inverse filter transfer function, the problem is the division by very small magnitude values which leads to numerical instability.

Instead of dividing by very small magnitude values of the MTF we suggest to replace the MTF by a constant value. Thus, a pseudo-inverse filter is synthesised. The pseudo-inverse filter has good results for cases where the noise power in the image is very low (high signal to noise (SNR) values).

For the MTF of the isotropic Gaussian (4.61), we can directly synthesise a 2D restoration filter as:

$$\mathfrak{R}_{\text{pinv}}(u,v,b) = \begin{cases} \frac{1}{\mathfrak{B}(u,v,b)} & \text{if } |\mathfrak{B}(u,v,b)| > \lambda_{\text{pinv}} \\ \lambda_{\text{pinv}} & \text{otherwise} \end{cases} \tag{4.92}$$

where $\mathfrak{R}_{\text{pinv}}$ is the pseudo-inverse filter and λ_{pinv} a threshold.

This filter is discontinuous. Another solution would be to use $1/\lambda_{\text{pinv}}$ if $|\mathfrak{R}(u,v,k)| \leq \lambda_{\text{pinv}}$ and the pseudo-inverse filter would be continuous. Nevertheless, the threshold λ_{pinv} is usually very small. Thus, using the value $1/\lambda_{\text{pinv}}$ would cause a strong amplification of high frequency noise. Hence, we retain (4.92) despite its discontinuity as pseudo-inverse filter.

For the anisotropic Gaussian blur and the linear motion blur model, we separated the MTFs in a horizontal and vertical component (see Equations (4.64) and (4.69)). Hence, we do similarly for the restoration filter. We compute two 1D restoration filters $\mathfrak{R}(u), \mathfrak{R}(v)$ using $\mathfrak{B}(x), \mathfrak{B}(y)$, respectively. First, we restore the lines using $\mathfrak{R}(u)$ and then the columns using $\mathfrak{R}(v)$ as:

$$\mathfrak{R}_{\text{pinv}}(u,v,b) = \mathfrak{R}_{\text{pinv}}(u,b_x) \cdot \mathfrak{R}_{\text{pinv}}(v,b_y) \tag{4.93}$$

Wiener Filter

We introduce here the Wiener filter because it is the optimum filter for the case of linear degradation and additive noise which we suppose for our DC frame formation. The Wiener filter $\mathfrak{R}_{\text{Wiener}}$ is defined as [145]:

$$\mathfrak{R}_{\text{Wiener}}(u,v,b) = \frac{\mathfrak{B}^*(u,v,b)}{|\mathfrak{B}(u,v,b)|^2 + 1/\text{SNR}} \tag{4.94}$$

where $\mathfrak{B}^*$ is the complex conjugate of $\mathfrak{B}$.

For the same reason as above, the restoration is performed separately along the rows and the columns of the image for the anisotropic Gaussian blur and the linear motion blur model. Thus, the image restoration in the update process (4.57) is then accomplished according to (4.91) with:

$$\mathfrak{R}_{\text{Wiener}}(u,v,b) = \mathfrak{R}_{\text{Wiener}}(u,b_x) \cdot \mathfrak{R}_{\text{Wiener}}(v,b_y) \tag{4.95}$$

SNR Estimation: Video frames suffer from noise due to the imaging system. This noise influences the restoration step. The SNR generally varies for different image sequences since it strongly depends on the imaging system. Therefore, we need to quickly estimate the noise from the available video data. Since we only have low-resolution frames in our disposal, we will estimate the SNR from them. We consider edges as a useful signal. Thus, the SNR for the image G can be computed as the ratio [145]:

$$\text{SNR} = \frac{h^2}{\sigma_{\text{N}}^2} \tag{4.96}$$

where h is the edge height and σ_{N}^2 the variance of the noise.

Here, we take h as the minimal gradient magnitude $||\nabla G||$ on the contours we considered significant (see Equation (4.75)). To compute the variance of the noise, the usual approach consists in determining a flat area and computing the variance with respect to the mean value. We propose to approximate this variance by the mean energy of the gradient on the complement to the set of significant contours $\bar{\mathcal{E}}_\nabla$:

$$\bar{\mathcal{E}}_\nabla = \{\mathbf{m} \,|\, \|\nabla G(\mathbf{m}, k)\| < \lambda_{\mathcal{E}}\} \tag{4.97}$$

Thus:

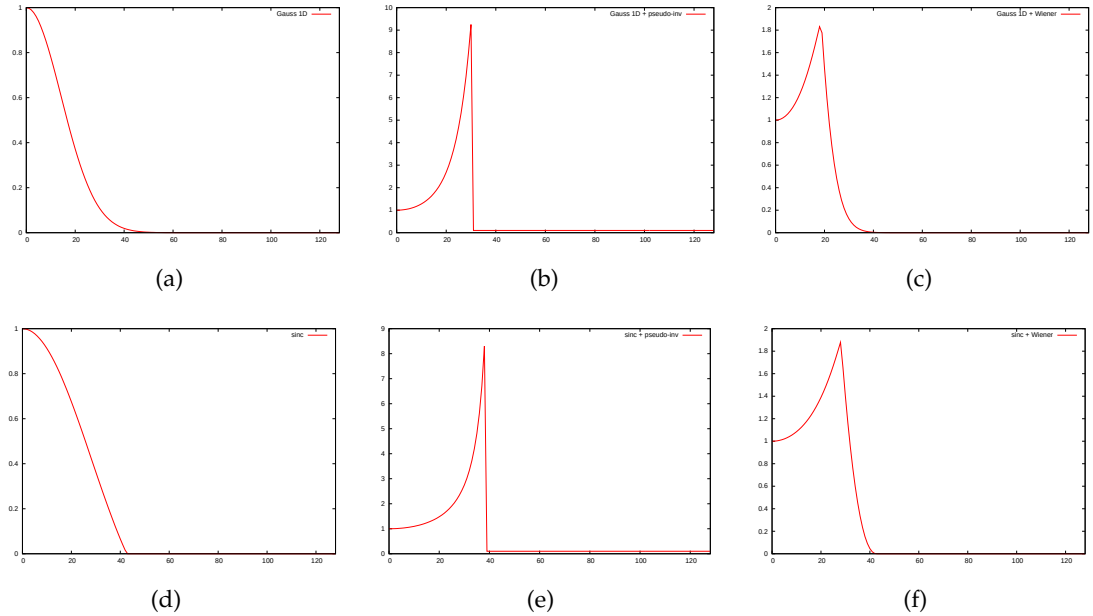
$$\sigma_{\text{N}}^2 = \frac{1}{\left|\bar{\mathcal{E}}_\nabla\right|} \sum_{\mathbf{m} \in \bar{\mathcal{E}}_\nabla} \|\nabla G(\mathbf{m}, k)\|^2 \tag{4.98}$$

where $\left|\bar{\mathcal{E}}_\nabla\right|$ is the cardinality of the set $\bar{\mathcal{E}}_\nabla$. Such an approach allows avoiding the segmentation of frames into flat areas which is a costly process.

Figure 4.7 shows a comparison of the restoration filters. For the sake of simplicity we only show the 1D case. In all examples the same blur size $b = 1.5$ has been used. We observe that the Gaussian MTF (Figure 4.7(a)) is narrower than the sinc MTF (Figure 4.7(d)). Hence, the restoration filters (Figures 4.7(b) and 4.7(c)) using this MTF are narrower than the restoration filters using the sinc MTF (Figures 4.7(e) and 4.7(f)) and less frequencies will be restored. The pseudo-inverse filter has a hard cut-off which is here smoother for the Wiener filter, and amplifies more the frequency before the cut-off. Therefore, the pseudo-inverse filter is likely to introduce more ringing artefacts than the Wiener filter. In fact, the smoothness of the cut-off of the Wiener filter depends on the SNR. It is a well known fact that the Wiener filter performs better in the case of noise, and when the SNR is high it approaches the inverse filter. The disadvantage of the Wiener filter is its complexity compared to the pseudo-inverse filter, because we need to calculate the SNR in the image by the Equations (4.96)–(4.98) which is a computationally tedious task. The use of the Wiener filter instead of the pseudo-inverse filter can thus be justified if the noise is sufficiently strong. In Section 4.3, we will discuss this choice for DC frames taking into account the nature of noise.

Validation of the Blur Estimation and Restoration

We created a synthetic sequence in order to validate our blur estimation and restoration method. The motion of the synthetic sequence shown in Figure 4.8 is translational along



(a) (b) (c)

(d) (e) (f)

Figure 4.7: Comparison of the restoration filters: (a) The 1D Gaussian MTF, (b) pseudo-inverse filter of (a), (c) the Wiener filter of (a), (d) the sinc MTF, (e) the pseudo-inverse filter of (d), (f) the Wiener filter of (d).

the x-axis. It was degraded with motion blur using the `fspecial` function of the MATLAB software [1]. This function convolves the image with a box function of the size l in the direction of a given angle. The size of the box function l corresponds to twice the blur size b. Since no Gibbs ringing artefacts appear in the original sequence, we did not restrain the edge response to the 10%–90% limits, but took instead the minimum and maximum values to determine the blur extent. Table 4.1 shows the results we obtained with our method. We can see that in the case of an odd length l of the box, our estimation is exact and otherwise we get an aberration of a half pixel which is due to the discretisation of the convolution kernel. In case of an even length l, an odd kernel of the size $l + 1$ is computed with half the weights on the kernel borders. Using $b > 3$ in this experiment results in smooth edges which cannot be detected as the gradient magnitudes drop below the threshold in Equation (4.76).

Figure 4.8: The synthetic test sequence.

l	b
2	1.5
3	1.5
4	2.5
5	2.5

Table 4.1: Results of the blur extent estimation on the synthetic sequence with l as the size of the box filter and b the estimated blur size.

Figures 4.9(a)–(d) show the super-resolution mosaics for the sequence of Figure 4.8 using different combinations of the blur models with the restoration filters. We can observe Gibbs ringing artefacts in all super-resolution mosaics. This is due to the normalisation of the filters. If $b > 1$ then $\mathfrak{B}_{\text{sinc}}$ contains zeros in the frequencies higher than N/b and $\mathfrak{B}_{\text{Gauss}}$ contains values near zero in the same frequencies. The ringing artefacts are strongest for the isotropic Gaussian blur model, and they are stronger for the anisotropic Gaussian blur model than for the linear motion blur model since with this normalisation $\mathfrak{B}_{\text{Gauss}}$ is narrower than $\mathfrak{B}_{\text{sinc}}$ and thus suppresses more of the high frequencies as illustrated in Figure 4.7. The ringing artefacts for the pseudo-inverse filter are stronger than for the Wiener filter due to an amplification of the middle frequencies and the discontinuity of the filter caused by the thresholding (4.92).

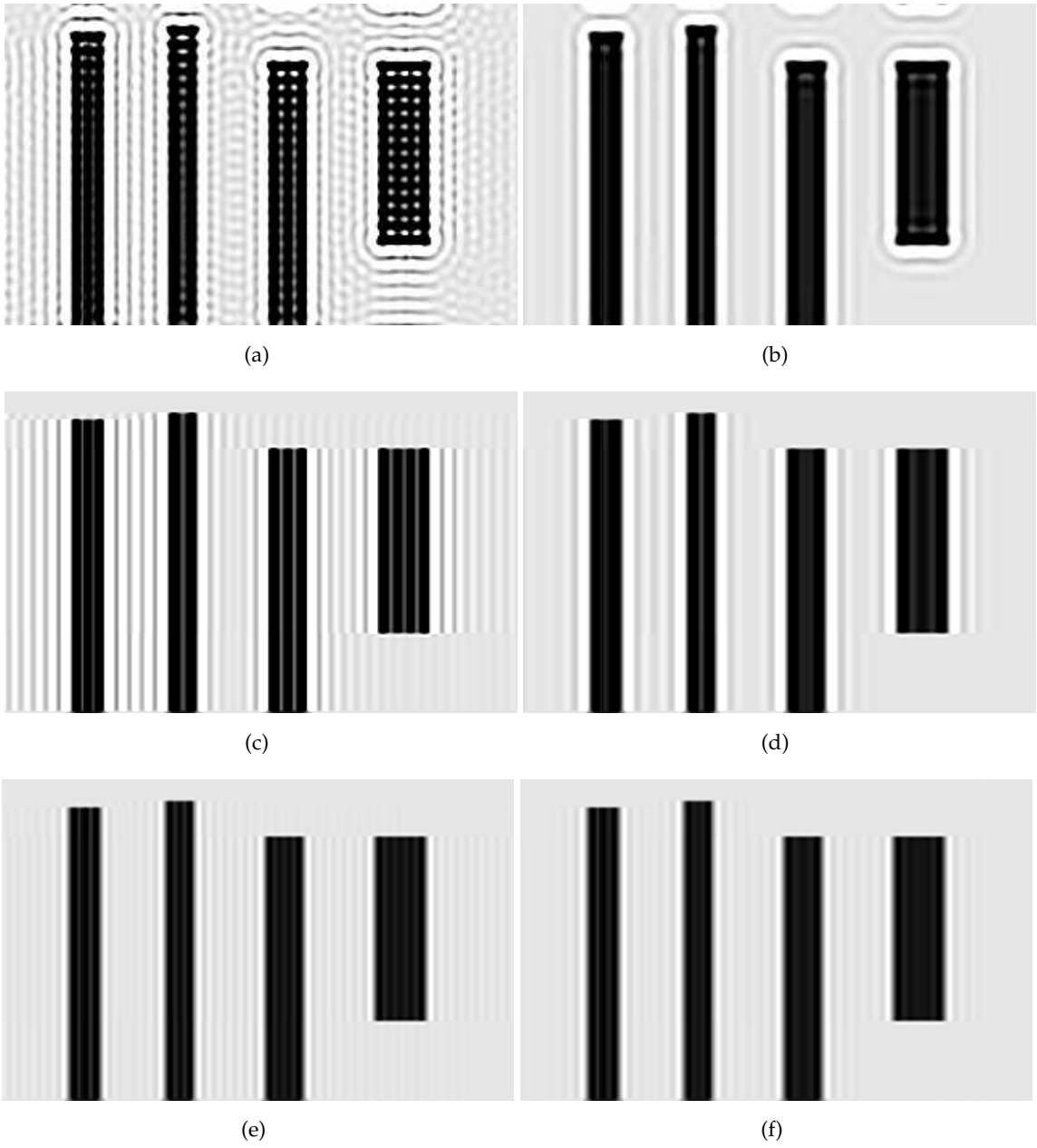
If we normalise the filters in another way e.g. assuming that $b = \sigma$ instead of $b = 3\sigma$ for the Gaussian filter, we obtain the super-resolution mosaic shown in Figure 4.10 for the pseudo-inverse filter. Using this normalisation for $b = 3$ the size of $\mathfrak{B}_{\text{Gauss}}$ corresponds to N. Hence, almost no ringing artefacts appear in the super-resolution mosaic, but it is still blurred after 10 iterations.

Our results coincide with those presented in [23]. Therein, Capel and Zisserman study the effects of a poorly estimated PSF. If the PSF is too low-pass, then the super-resolution image develops ringing artefacts. If the PSF is too high-pass the super-resolution estimate is blurry.

4.2.5 Regularisation

The process of restoration in general increases the high frequency content of the image. In case of DC frames it is specifically hold, because we observe the blocking effect due to the independent averaging of each block. Thus, we expect to get a degradation process due to aliasing. In order to compensate this process we need to use an antialiasing filter. Therefore, with regard to the super-resolution process described by Equation (4.57), the aliasing effect in each DC frame will result in a strong motion compensation error. This is due to the fact that the geometric transformation T will not lead to an exact superposition of the details in high frequency areas such as textures and edges. Thus, the difference image $G(k) - G^i(k)$ will be of high magnitude. Adding this elevated error image to the super-resolution mosaic will cause artefacts in the super-resolution mosaic amplified at each iteration step.

Thus, we propose the following regularisation operator A. It is based on the weight-



(a) (b)

(c) (d)

(e) (f)

Figure 4.9: Super-resolution mosaics for the synthetic sequence after 10 iterations using $\varsigma = 2$ and $b = 3$: (a) $\mathfrak{B}_{\text{Gauss2D}}$ restored by the pseudo-inverse filter (b) $\mathfrak{B}_{\text{Gauss2D}}$ restored by the Wiener filter, (c) $\mathfrak{B}_{\text{Gauss}}$ restored by the pseudo-inverse filter (d) $\mathfrak{B}_{\text{Gauss}}$ restored by the Wiener filter, (e) $\mathfrak{B}_{\text{sinc}}$ restored by the pseudo-inverse filter, (f) $\mathfrak{B}_{\text{sinc}}$ restored by the Wiener filter.

Figure 4.10: The super-resolution mosaic after 10 iterations with $\varsigma = 2$ for $\mathfrak{B}_{\text{Gauss}}$ restored by the pseudo-inverse filter using the normalisation $b = \sigma$.

ing of the error $G(k) - G^i(k)$ by the gradient magnitude of the low-resolution image $G(k)$. We determine for each pixel in the simulated image $G^i(k)$ if it corresponds to an edge or a texture in the low-resolution image $G(k)$. In such pixels the weights have to penalise the contribution of the pixel values.

In order to exactly relate the pixels in the simulated image $G^i(k)$ to the gradient magnitude of the image $G(k)$, the image of the gradient magnitude undergoes the super-resolution process. We calculate the magnitude of the Roberts gradient $\|\nabla_{\mathrm{r}} F\|$ for the super-resolution mosaic with respect to the low-resolution image $G(k)$. Thanks to the geometric transformation $\mathrm{T}(k)$, the corresponding position $\hat{\mathbf{n}}$ at higher resolution in a frame of the pixel $\mathbf{m}$ in the mosaic is known. If we divide it by the upsampling factor ς, we obtain the corresponding decimal position $\hat{\mathbf{n}}/\varsigma$ in the low-resolution image G. Then, we can calculate the magnitude of the Roberts gradients on the luminance component of $F(\hat{\mathbf{n}}/\varsigma)$, $\|\nabla_{\mathrm{r}} \mathbf{Y}_F(\hat{\mathbf{n}}/\varsigma)\|$, according to Equation (3.70). The gradient magnitude is then compensated according to T and downsampled by S^{-1}. We compute the matrix of the weights for antialiasing as the inverse of the compensated and downsampled gradient magnitude. Therefore, we define the following penalty function:

$$\Psi(x, \lambda_{\mathrm{A}}) = \begin{cases} \frac{\lambda_{\mathrm{A}}}{x} & \text{if } x > \lambda_{\mathrm{A}} \\ 1 & \text{otherwise} \end{cases} \tag{4.99}$$

where λ_{A} is a threshold. Figure 4.11 depicts the penalty function.

Then, the regularisation operator is computed as:

$$\mathrm{A}(k) = \Psi(\mathrm{S}^{-1}.\mathrm{T}^{-1}(k).\, \|\nabla_{\mathrm{r}} \mathbf{Y}_F\|\,, \lambda_{\mathrm{A}}) \tag{4.100}$$

In fact, the threshold λ_{A} defines if a certain gradient belongs to an edge or a texture. We chose the its value as 2.5% of the grey level range.

As we stated above, the restoration in the frequency domain causes ringing artefacts on the image borders due to the fact that a periodic signal is assumed. In a first step, we mitigate these artefacts by padding the image using a mirror before the Fourier transform. But, this

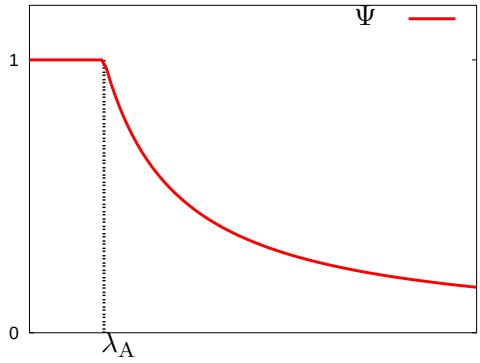


Figure 4.11: The penalty function Ψ.

is not sufficient. Therefore, we set by default the weights of the border rows and columns to zero.

Figure 4.12 shows a comparison of mosaics obtained with and without regularisation. We observe much more artefacts in the cut-out of the super-resolution mosaic after 20 iterations without regularisation (Figure 4.12(a)) than in the cut-out of super-resolution mosaic with regularisation (Figure 4.12(b)).

(a) (b)

Figure 4.12: A cut-out of the super-resolution mosaic of the sequence "Chancre1" CERIMES-SFRS® after 20 iterations with $\varsigma = 2$ using $\mathfrak{B}_{\text{sinc}}$ restored by the pseudo-inverse filter: (a) without regularisation and (b) with regularisation.

4.3 Results

We showed in the last chapter, that when a mosaic is constructed from DC images, the resolution is too low to give a scene overview and the image quality is not satisfying due to the strong degradations of the DC images. Therefore, we developed in this chapter a super-resolution algorithm to increase the resolution and to improve the visual quality of the mosaic. Here, we consider that twice the DC-resolution is an appropriate resolution for this task. Furthermore, we show that increasing DC-resolution mosaics to this target resolution, is less costly than calculating full-resolution mosaics and downsampling them to this target resolution. Hence, full decoding of the compressed stream is not required.

Here, we show the results we obtained on broadcasted MPEG-2 compressed video with

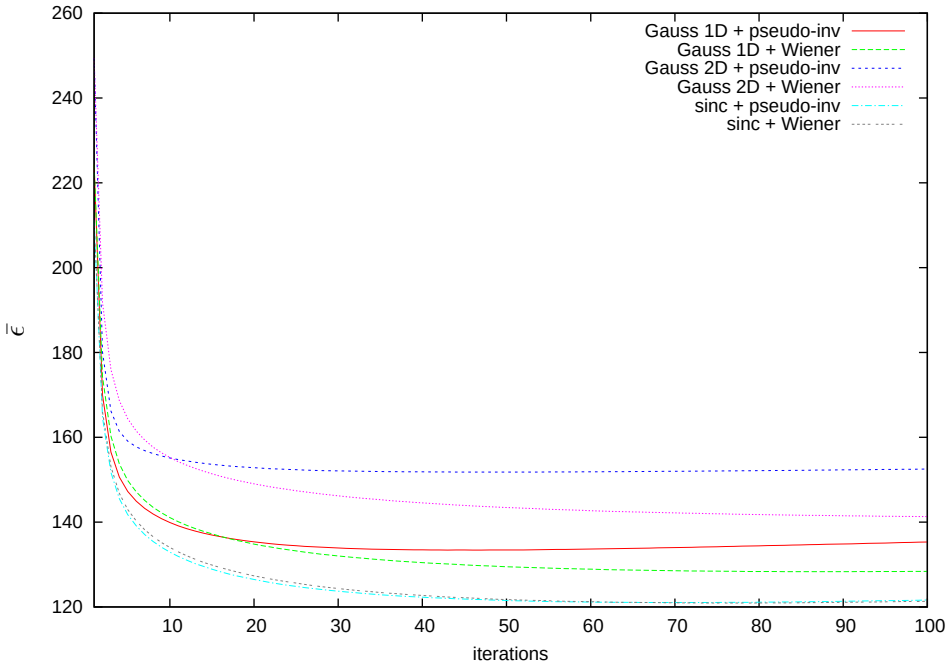
unknown blur parameters. In order to determine the best blur model and its restoration, we tried all combinations of the blur models with the restoration filters. In order to evaluate the performance of the different combinations of blur models and restoration filters, we use the error $\bar{\epsilon}$ (4.59) as a quality measure. All examples are computed with luminance correction and regularisation except if we indicate the contrary. The size of DC images is 90×72 pixels for all test sequences.

Figure 4.13 plots the error $\bar{\epsilon}$ of Equation (4.59) versus the number of iterations for the sequence "Chancre1" (see Figure 3.21(b)) we obtained with and without regularisation. For each blur model, the error $\bar{\epsilon}$ of the pseudo-inverse filter decreases and converges faster than for the Wiener filter. Due to its fine high frequency details such as the leaves, this sequence is susceptible to aliasing artefacts in the super-resolution process. Thus, after a few iterations the super-resolution process is becoming unstable despite the regularisation and the error $\bar{\epsilon}$ increases due to the amplification of aliasing and Gibbs ringing artefacts. In this case the Wiener filter is more robust. Comparing the anisotropic Gaussian MTF $\mathfrak{B}_{\mathrm{Gauss}}$ with the sincMTF $\mathfrak{B}_{\mathrm{sinc}}$, the sinc MTF better minimizes the $\bar{\epsilon}$ and the instability appears after more iterations. The isotropic Gaussian MTF $\mathfrak{B}_{\mathrm{Gauss2D}}$ shows very high values and becomes unstable after very few iterations. Thus, it does not seem to model appropriately this kind of blur.

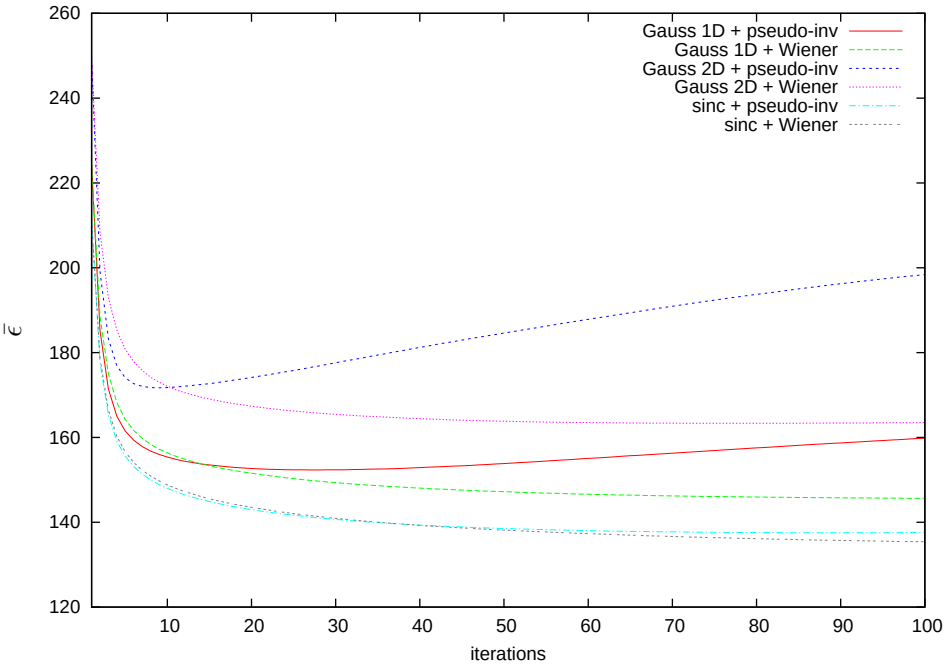
These $\bar{\epsilon}$ values seem relatively high. This can be explained by the fact that there is already a quite high initial error due to interpolations for decimal coordinates in the motion compensation. The more high frequencies are present in the low-resolution image, the higher is the initial error as the high frequencies are lost due to these interpolations. For this sequence with a strong high frequency content, we obtain an initial error $\bar{\epsilon}$ of almost 150. This value represents only the error induced by the motion compensation of the low-resolution images without increasing the resolution. If we consider in addition the error due to inaccurate motion estimation the initial $\bar{\epsilon}$ increases to a value of 261. These values were obtained on the sequence without luminance correction.

We observe in Figure 4.13(b) that the use of the regularisation increases the values of $\bar{\epsilon}$. Nevertheless, according to [150] the MSE and related measures ($\bar{\epsilon}$ is based on the mean square error) are not necessarily an objectives measures for the visual image quality. This is confirmed by our results. For instance, the super-resolution process for the isotropic Gaussian seems very unstable for the pseudo-inverse filter as is increases dramatically after few iterations for the regularised case. But if we compare the regularised after 100 iterations (Figure 4.14(b)) with the non-regularised mosaic (Figure 4.14(a)), we observe much less artefacts. The super-resolution mosaic for $\mathfrak{B}_{\mathrm{sinc}}$ in combination with the pseudo-inverse filter after 20 iterations is shown in Figure 4.15.

Tests on other sequences, "Chancre2" (Figure 4.16) and "Tympanon" (Figure 3.27(a)), confirm the $\bar{\epsilon}$ results as illustrated in Figures 4.17 and 4.18. In all experiments, $\mathfrak{B}_{\mathrm{sinc}}$ minimises our quality measure better. If only a small number of iterations are performed which is our objective, the pseudo-inverse filter is preferable due to a faster decrease and convergence of $\bar{\epsilon}$ and the lower computational complexity. If more iterations are performed, the Wiener filter should be chosen since it is more robust. Figures 4.19 and 4.20 show the cor-



(a)



(b)

Figure 4.13: The error measure $\bar{\epsilon}$ versus the number of iterations for the sequence "Chancre1": (a) without regularisation, (b) with regularisation.

(a) (b)

Figure 4.14: The super-resolution mosaic of the sequence "Chancre1" CERIMES-SFRS® with $\varsigma = 2$ using $\mathfrak{B}_{\text{Gauss}}$ 2D in combination with the pseudo-inverse filter after 100 iterations: (a) without regularisation, (b) with regularisation.

(a) (b)

Figure 4.15: The super-resolution mosaic of the sequence "Chancre1" CERIMES-SFRS® with $\varsigma = 2$ using $\mathfrak{B}_{\text{sinc}}$ in combination with the pseudo-inverse filter and regularisation: (a) The initial mosaic, (b) the super-resolution mosaic after 20 iterations.

Figure 4.16: Sequence "Chancre2" extracted from the documentary "Le chancre coloré du platane" CERIMES-SFRS®.

responding super-resolution mosaics obtained using $\mathfrak{B}_{\text{sinc}}$ in combination with the pseudo-inverse filter and the regularisation.

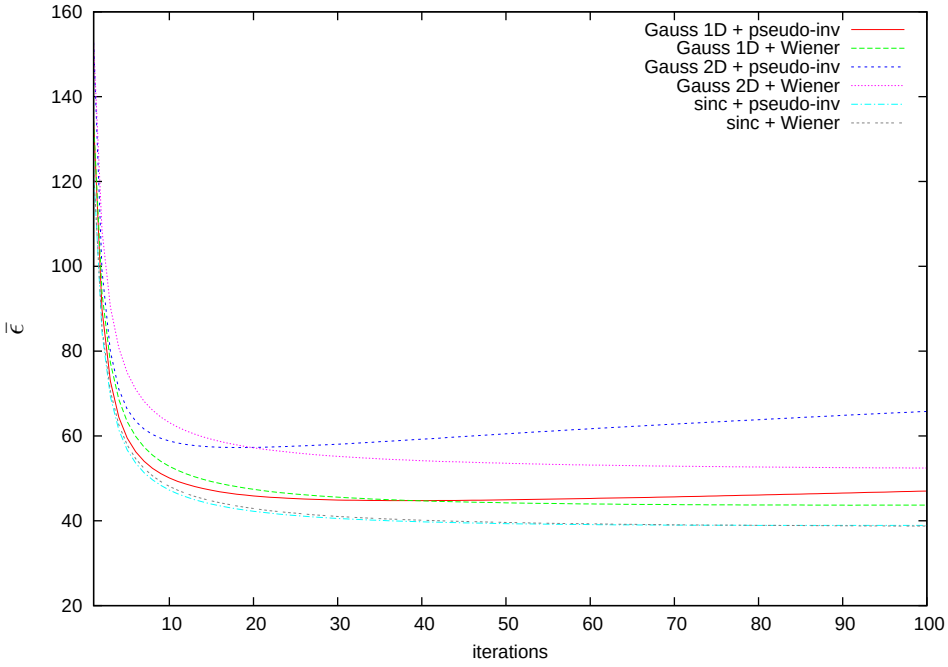


Figure 4.17: The error measure $\bar{\epsilon}$ versus the number of iterations for the sequence "Chancre2".

The regularisation of the mosaic in Figure 4.20, mainly affects the texture in the background and we obtain an attenuation of the artefacts. The overall results are shown in Table 4.2. Therein, the attenuation is computed as the maximum absolute difference of the color values of the mosaic with and without regularisation:

$$\max_{R,G,B} |F(\mathbf{p}) - F_{\text{A}}(\mathbf{p})| \tag{4.101}$$

where F is the mosaic obtained without regularisation and F_{A} the mosaic obtained with regularisation. Figure 4.21 shows the difference image $F - F_{\text{A}}$ for the mosaic of the sequence "Chancre1".

sequence	max attenuation
Chancre1	218
Chancre2	124
Tympanon	228

Table 4.2: The results of the regularisation: maximum level of attenuation of the artefacts in the RGB channels for the mosaics shown in Figures 4.12(b), 4.19(b) and 4.20(b).

Table 4.3 shows the computational times for the mosaics shown in Figures 4.12(b), 4.19(b) and 4.20(b). These times were measured on an Intel Pentium 4 3.00GHz processor using a

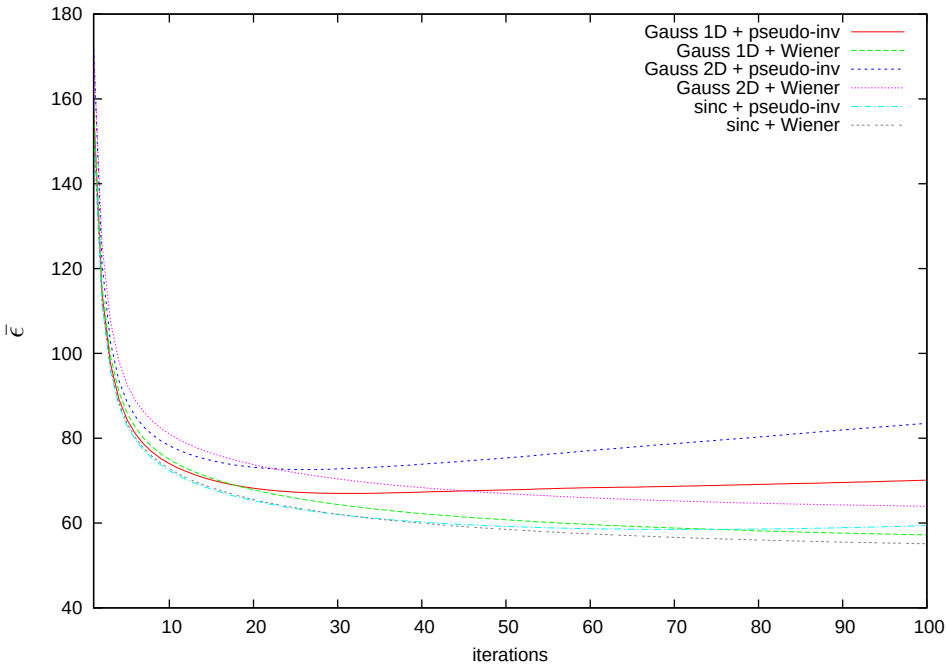


Figure 4.18: The error measure $\bar{\epsilon}$ versus the number of iterations for the sequence "Tympanon".

(a) (b)

Figure 4.19: The super-resolution mosaic of the sequence "Chancre2" CERIMES-SFRS® with $\varsigma = 2$ using $\mathfrak{B}_{\text{sinc}}$ in combination with the pseudo-inverse filter and regularisation: (a) The initial mosaic, (b) the super-resolution mosaic after 20 iterations.

(a) (b)

Figure 4.20: The super-resolution mosaic of the sequence "Tympanon" CERIMES-SFRS® with $\varsigma = 2$ using $\mathfrak{B}_{\text{sinc}}$ in combination with the pseudo-inverse filter and regularisation: (a) The initial mosaic, (b) the super-resolution mosaic after 25 iterations.

Sequence	No. I-frames	No. iterations	(1)	(2)	(3)
Chancre1	22	20	24.733s	10m54.859s	11m19.592s
Chancre2	16	20	27.801s	6m29.652s	6m57.453
Tympanon	12	25	7.166s	6m19.1s	6m27.266s

Table 4.3: The computational times for the mosaics shown in Figures 4.12(b), 4.19(b) and 4.20(b): (1) the computational time for data extraction and motion estimation, (2) the computational time for concatenation of the motion models in the geometric transformations, illumination correction, blur estimation, super-resolution reconstruction, (3) the total computational time.

Figure 4.21: Attenuation by the regularisation operator for the sequence "Chancre1": This image shows the difference between the mosaics obtained without and with regularisation. A medium grey indicates that no difference; the clearer or darker a region the greater is the difference.

non optimised C++ code and the VXL image library [200]. If we compare the computational times with these of construction of DC-resolution images in Table 3.1 we observe an increase. For the sequence "Tympanon" the increase comes out to about 6m. We already claimed in the last chapter that the concatenations to compute the global geometric transformations (3.61) are a costly task in the mosaic construction. In the super-resolution method the geometrical transformations are realised at high resolution. Thus, the concatenations are computed for ς^2 more pixels than before. This causes an important increase in computational time. Additionally, the succeeding forward and inverse Fourier transforms are costly. Moreover, the padding of the image to a size of $2^j \times 2^j$ may produce quite large images and slows down the the restoration process.

We note that according to [101] the increase of resolution by a factor ς requires in the ideal situation ς^2 frames which corresponds to 4 in our case. In practice as shown in Table 4.3 we use a higher number of frames.

4.3.1 Limits of the Method

Along the Chapters 3 and 4, we mentioned drawbacks of the method for the construction of super-resolution mosaics despite its relative simplicity and computational efficiency. The influence of noise in the restoration process has already been discussed. Here, we stress, that the main limit of the method is the capacity to correctly estimate motion from MPEG macroblock optical flow in order to compute the geometric transformations $\mathrm{T}(k)$ in Equation (4.57). This is not the case e.g. for the HD sequence encoded in MPEG-2 (see Figure 4.22(a)) where the camera pans fast. Consequently, most of the macroblocks are intra-coded and the few remaining motion vectors do not describe the true motion. Thus, the motion estimated on these vectors is absolutely erroneous. Hence, for the time being we reestimate the motion using the Motion2D software [113]. Nevertheless, as we presented in Appendix B for HD video new standards such as H.264/MPEG-4 AVC and H.264/MPEG-4 SVC provide a better motion estimation. This can be used in the future of this work. The objective here is to validate our blur estimation and restoration on real data with known blur.

This sequence is strongly degraded by motion blur due to the fast panning of the camera. Figure 4.22 shows one blurred frame of the sequence. Using our blur estimation presented above, we obtained blur sizes b_x up to 3.5 pixels in horizontal direction for the HD sequence. If b_x is estimated in the sequence of DC images it averages out to about 7 times smaller than in the original sequence. Thus our method of computing b_x and b_y (see Equations (4.89) and (4.90)) using a scale factor is confirmed. Figure 4.23 shows the error values $\bar{\epsilon}$ versus the number of iterations for the different combinations of the blur models and restoration filters. Figure 4.24(b) shows the super-resolution mosaic with $\varsigma = 2$ after 10 iterations. This is the best result in terms of the $\bar{\epsilon}$ obtained for $\mathfrak{B}_{\mathrm{sinc}}$ restored by the pseudo-inverse filter. The computational time for this mosaic is 5m29.576s using 14 I-frames. We also showed the mosaic without luminance corrections (Figure 4.24(c)) to demonstrate that our luminance correction can also handle this case.

We also used this sequence to assess the influence of noise in the restoration. In order to test the influence of noise in the original images with respect to the DC images, we added a

Figure 4.22: A motion blurred frame of the HD sequence.

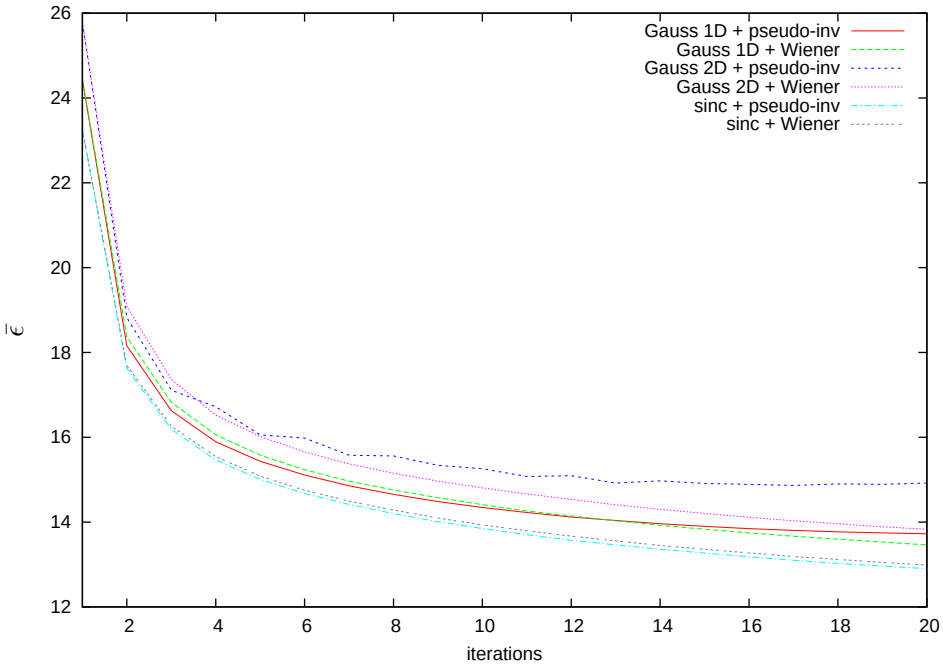
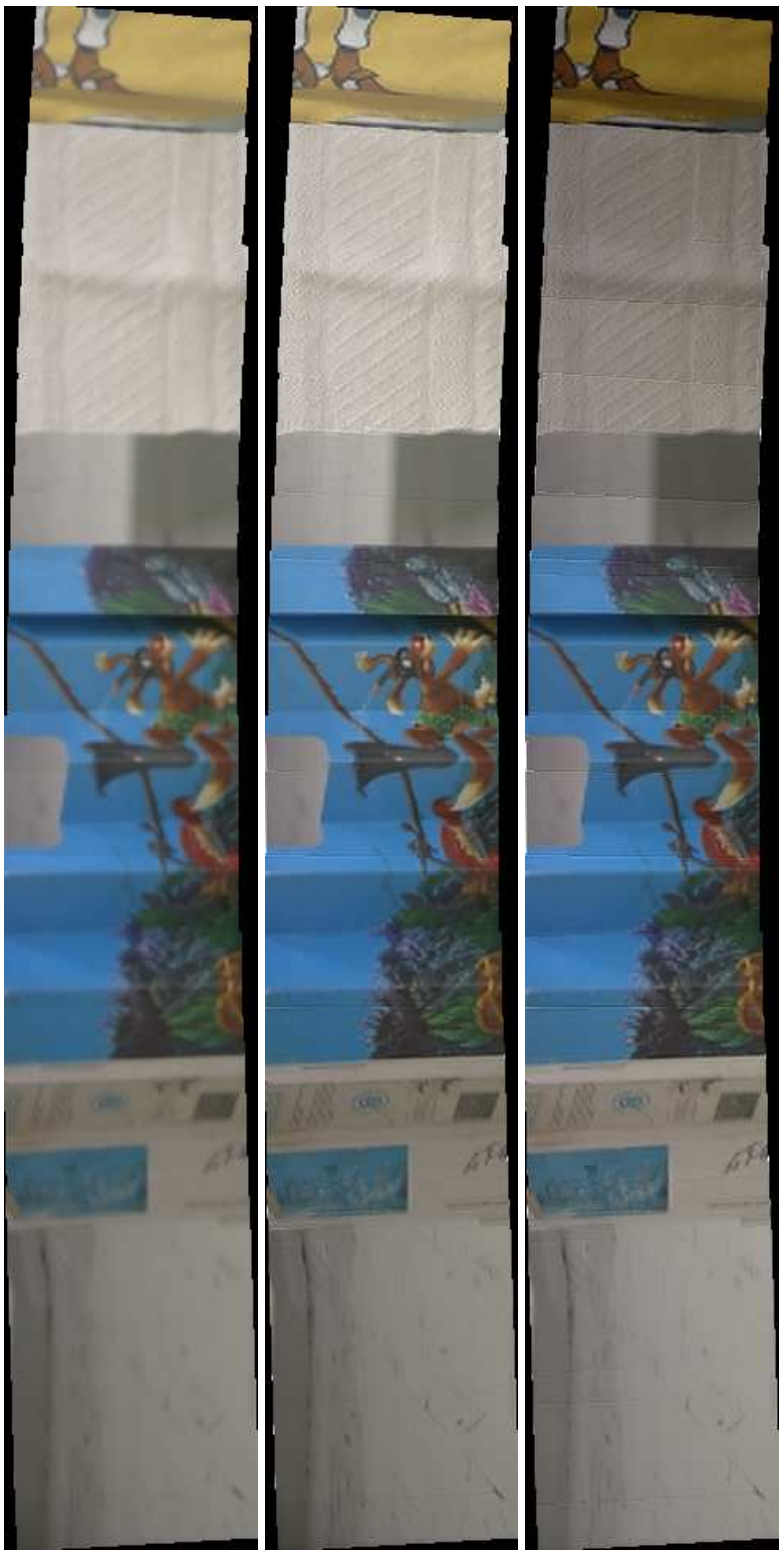


Figure 4.23: The error measure $\bar{\epsilon}$ versus the number of iterations for the HD sequence with $\varsigma = 2$.



(a) (b) (c)

Figure 4.24: (a) The initial super-resolution mosaic for the HD sequence with $\varsigma = 2$: (a) The initial mosaic, (b) the mosaic after 10 iterations using $\mathfrak{B}_{\text{sinc}}$ restored by the pseudo-inverse filter, (c) the mosaic after 10 iterations using $\mathfrak{B}_{\text{sinc}}$ restored by the pseudo-inverse filter without luminance correction.

zero mean Gaussian noise with variance 0.01 to the original sequence. As can be expected, the DC image formation by local averaging induces a low-pass effect. The SNR values calculated on the DC images of the noisy sequence, are the same values as for the DC images of the original sequence, while the SNR in the full-resolution noisy sequence decreases on the average by a factor 0.6 with this noise. This low-pass effect explains the sufficiency of the pseudo-inverse filter restoration in this framework.

4.3.2 Comparison with Mosaicing from Raw Video

Here, we compare the results of the proposed super-resolution method with downsampled full-resolution mosaics in terms of visual quality and computational time for the mosaic construction. We do not consider the computational time for the motion estimation as above since we used the same motion models for the construction of all mosaics, but at different scales. In this test no luminance correction was applied.

In this experiment we used the sequence "Tympanon" shown in Figure 3.27(a). On the one hand, we computed the super-resolution mosaics with $\varsigma = 2$, $\mathfrak{B}_{\mathrm{sinc}}$, pseudo-inverse filter and regularisation for 0, 10 and 25 iterations whereas the first one corresponds to the initial mosaic of Equation (4.58). On the other hand, we decoded the sequence, computed the full-resolution mosaics at temporal I/P and I-resolutions, and sampled them down by a factor 4. We used bilinear interpolation to accomplish this. Consequently, all the resulting mosaics have the same spatial resolution.

The computational times are shown in Table 4.4. As expected the computational times for the full-resolution mosaics are already high for the temporal I- and I/P-resolutions and increase dramatically with the increase in temporal resolution. For this reason, we did not compute the mosaic at the full temporal resolution of the sequence. Here, the costliest operation for the mosaic construction is the computation of the geometrical transformations at full resolution as for each pixel the motion models have to be concatenated. The computational times for the super-resolution mosaics are lower and increase noticeably with the number of iterations. Hence, the costly operations for super-resolution mosaic construction are the successive motion compensations and Fourier transforms during the iterations.

Temporal res	Initial spatial res	No. frames	No. iterations	Mosaic construction
I/P	720×576	60	-	51m16.180s
I	720×576	12	-	10m23.485s
I	90×72	12	25	6m50.832s
I	90×72	12	10	2m46.717s
I	90×72	12	0	0m41.322s

Table 4.4: The computational times for the mosaic construction from the sequence "Tympanon" CERIMES-SFRS®.

Figure 4.25 shows the corresponding mosaics. The downsampled full-resolution mosaics of Figure 4.25(a) and (b) are slightly sharper and provide more fine details than the super-resolution mosaics, but this is at the expense of high computational costs. Both super-

resolution mosaics (Figure 4.25(c) and Figure 4.25(d)) are a little blurred, but nevertheless provide a good scene overview. In comparison to the initial mosaic of the super-resolution algorithm (Figure 4.25(e)), we can see that super-resolution algorithm performs well. Visually there is no difference in detail between the super-resolution mosaic after 25 iterations and the super-resolution mosaic after 10 iterations, but less ringing artefacts appear in the latter. Thus, computational time with respect to the 25 iterations determined by the stopping criterion (4.59) can still be reduced by using a stricter criterion.

4.4 Conclusion

We showed in the previous chapter, that when a mosaic is constructed from DC images, the resolution is too low to give a scene overview and the image quality is not satisfying due to the strong degradations of the DC image. For this reason, we proposed in this chapter a super-resolution algorithm based on iterative backprojections to increase the resolution and to improve its visual quality of the mosaic. The success of the super-resolution algorithm mostly depends on the success of estimating the exact motion between successive images and deriving the correct motion blur in each image.

To this end, we proposed an efficient method to estimate directional blur from a sequence of DC images. Compared to the solutions known from literature, we proposed a new method for the estimation of motion blur based on the direction of global camera motion and significant (not always vertical) contours in the image plane. Based on this blur we considered three degradation filter models: isotropic Gaussian blur, anisotropic Gaussian blur and linear motion blur (sinc). Our success in estimating the blur size and also the linear motion MTF allows us to get good results for the super-resolution image. The assumption of linear motion blurs seems to be justified here because of the short exposure time. We performed restoration in frequency domain by the well-known pseudo-inverse and Wiener filters. In the case of a small number of iterations the pseudo-inverse filter performs better than the Wiener filter as less noise is present in the DC images. However, if several iterations are performed the Wiener filter is more robust against the amplification of artefacts. The results of the simulation show that the assumption of linear motion during the exposure time is correct and by using the sinc MTF in the restoration filter we got the best results in terms of the error measure $\bar{\epsilon}$. The assumption of Gaussian MTFs (isotropic and anisotropic) is usually not correct in case of motion. Therefore, using that function in the restoration filter causes more degraded images compared to the sinc MTF.

We explicitly introduced a regularisation operator in the restoration process. This regularisation operators is based on the image content and on the knowledge of error formation by motion compensation.

The motion estimation based on MPEG motion compensation vectors can fail in some cases. In our example of a HD sequence with a fast camera motion most of the macroblocks are consistently intra-coded and the few remaining motion vectors do not describe the true motion in the scene so that the estimated motion model is inaccurate. However, video compression is developing. For instance, the H.264/MPEG-4 AVC video compression standard

(a) (b) (c)

(d) (e)

Figure 4.25: Mosaics for the sequence "Tympanon" CERIMES-SFRS® without luminance correction: (a) the downsampled full-resolution mosaic at temporal I/P-resolution, (b) the downsampled full-resolution mosaic at temporal I-resolution, (c) the super-resolution mosaic after 25 iterations, (c) the super-resolution mosaic after 10 iterations, (d) the initial super-resolution mosaic.

furnishes an improved and more reliable motion estimation.

We obtain an increase in computational time with respect to the mosaics constructed at DC resolution. Anyhow, these computational times are still interesting for the task of video summarisation. We showed that the computational times can be reduced by choosing a stricter stopping criterion or avoiding restoration in the frequency domain.

The main drawback of this method is that moving objects have not been considered. Moreover, due to restoration in the frequency domain, artefacts can appear in the mosaic. These drawbacks will be addressed in the next chapter where we present a super-resolution method restoring the blur in the spatial domain. This allows to restore local blurs due to moving objects and to decrease the computational costs.

Chapter 5

Super-Resolution in the Region of Interest

In the last chapter, we presented a super-resolution method for the restoration of global blurs. Unfortunately, in video, we observe more complex situations where local blurs appear in each frame. These blurs can be due to the fast motion of objects, or insufficient lighting that reduces the shutter-speed of the auto-exposure cameras. Thus, the objective in this chapter is to restore these local blurs. Therefore, we restore the background and moving objects separately. This is difficult to realise in the Fourier domain. In addition, the successive Fourier transforms are costly and may introduce artefacts due to rounding errors. Thus, we choose to perform the restoration in the spatial domain.

The motion of objects in real video sequences can be very complex. For example consider the sequence shown in Figure 3.27(b) where a person is walking in the forest. Particularly in the case of very low-resolution images, which is our case, the objects are typically represented by only few pixels. For these reasons, its it very difficult to estimate the motion of moving objects and to superimpose them accurately enough for super-resolution. In contrast to [198] where moving objects such as cars with rigid motion are super-resolved or [75] where 2D parametric motion of objects is assumed, our aim is to propose a generic method. Therefore, we develop in this chapter a single image method to restore the blur and to increase the resolution of moving objects and then we derive from this a spatial domain method to super-resolve the background. Figure 5.1 illustrates our restoration scheme where background and moving objects are restored separately. Afterwards, some moving objects are reinserted into the background mosaic.

The remainder of this chapter is organised as follows. First, we give an overview of single image spatial domain restoration methods in Section 5.1. Then, we present our object restoration and super-resolution methods in Section 5.2. Some results are presented in

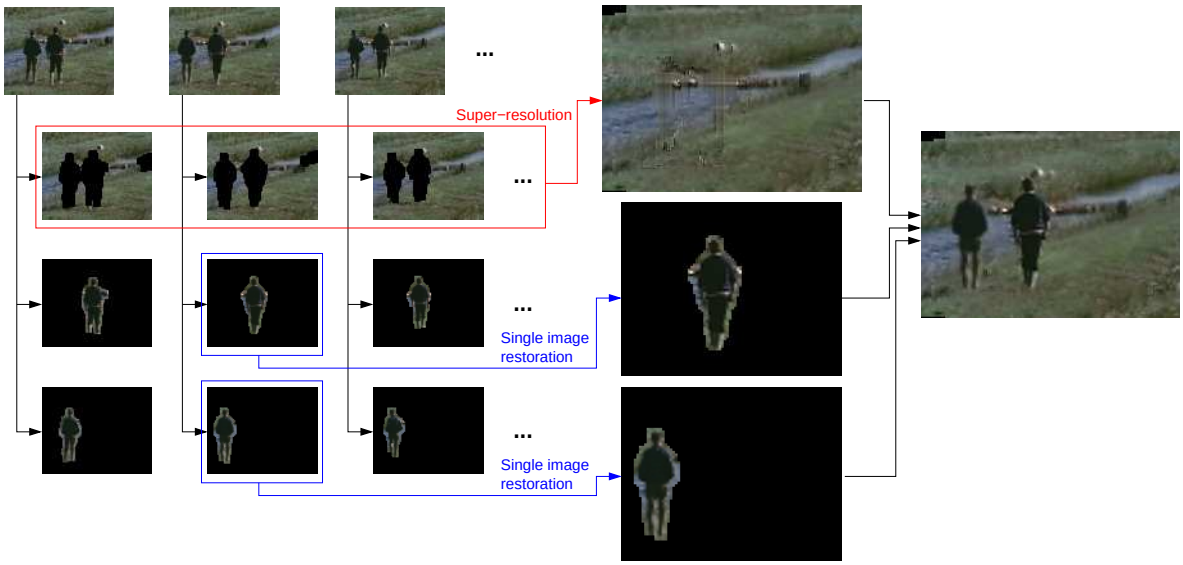


Figure 5.1: Separate restoration of the background and moving objects.

Section 5.3. Finally, we conclude our work in Section 5.4.

5.1 State of the Art

An important advantage of iterative restoration techniques is that there is no need to determine or implement the inverse of a blurring operator [81]. Thus, using this kind of methods the restoration can be accomplished in spatial domain. The modeling of the image degradation is straightforward for single image restoration. The degradation model is described by:

$$\tilde{\mathbf{I}} = \mathrm{B}\mathbf{I} + \mathbf{V} \tag{5.1}$$

where the vectors $\tilde{\mathbf{I}}$,$\mathbf{I}$,$\mathbf{V}$ represent, respectively, the noisy and blurred image, the original image, the noise. The matrix B represents the blur operator. Based on this degradation model, we briefly present in this section the basics of spatial domain restoration techniques. Further information on these techniques can be found in [78, 8, 81].

5.1.1 Basic Iterative Methods

A class of these algorithms can be derived in a very straightforward way. Based on Equation (5.1) with $\mathbf{V} = 0$, the original image can be approximated as [81]:

$$\mathbf{I} = \mathbf{I} + \beta(\tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}) \tag{5.2}$$

where the vectors $\tilde{\mathbf{I}}$,$\mathbf{I}$,$\mathbf{V}$ represent, respectively, the blurred image, the unknown ideal image, the noise, the matrix B represents the known blur operator, and β is a gain parameter.

Applying successive approximations an iterative process to estimate $\mathbf{I}$ can be built [81]:

$$\mathbf{I}^{i+1} = \mathbf{I}^i + \beta(\tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}^i) \tag{5.3}$$

The earliest reference to this iteration was proposed by Van Cittert [31] where the gain parameter β equals 1. This algorithm was modified in [78] where Jansson replaced the gain parameter β with a relaxation parameter that depends on the signal.

Van Cittert stated that the image data $\tilde{\mathbf{I}}$ could be considered as a first approximation of $\mathbf{I}$ [31]:

$$\mathbf{I} = \tilde{\mathbf{I}} + \Delta^1 \tag{5.4}$$

Inserting (5.4) in (5.1) and assuming that $\mathbf{V} = 0$ yields:

$$\tilde{\mathbf{I}} = \mathrm{B}\tilde{\mathbf{I}} + \mathrm{B}\Delta^1 \tag{5.5}$$

Denoting:

$$\mathbf{I}^1 = \mathrm{B}\tilde{\mathbf{I}} \tag{5.6}$$

Then from (5.5) and (5.6), we obtain:

$$\tilde{\mathbf{I}} - \mathbf{I}^1 = \mathrm{B}\Delta^1 \tag{5.7}$$

Assuming that:

$$\Delta^1 = \tilde{\mathbf{I}} - \mathbf{I}^1 \tag{5.8}$$

and inserting in (5.4) yields:

$$\mathbf{I} = 2\tilde{\mathbf{I}} - \mathbf{I}^1 \tag{5.9}$$

Then, the approximation can be refined by:

$$\Delta^1 = \tilde{\mathbf{I}} - \mathbf{I}^1 + \Delta^2 \tag{5.10}$$

Inserting (5.10) in (5.7) yields:

$$\tilde{\mathbf{I}} - \mathbf{I}^1 = \mathrm{B}(\tilde{\mathbf{I}} - \mathbf{I}^1) + \mathrm{B}\Delta^2 \tag{5.11}$$

Assuming that:

$$\Delta^2 = \tilde{\mathbf{I}} - \mathbf{I}^1 - \mathrm{B}\tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}^1 = \tilde{\mathbf{I}} - 2\mathbf{I}^1 + \mathbf{I}^2 \tag{5.12}$$

with:

$$\mathbf{I}^2 = \mathrm{B}\mathbf{I}^1 \tag{5.13}$$

Inserting (5.12) in (5.10) yields:

$$\Delta^1 = 2\tilde{\mathbf{I}} - 3\mathbf{I}^1 + \mathbf{I}^2 \tag{5.14}$$

Then, inserting (5.14) in (5.4), we obtain:

$$\mathbf{I} = \tilde{\mathbf{I}} + 2\tilde{\mathbf{I}} - 3\mathbf{I}^1 + \mathbf{I}^2 = 3\tilde{\mathbf{I}} - 3\mathbf{I}^1 + \mathbf{I}^2 \tag{5.15}$$

Applying further approximations the following iterative scheme is finally obtained [78]:

$$\mathbf{I}^{i+1} = \mathbf{I}^i + (\tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}^i) \tag{5.16}$$

The above iterative scheme minimises the mean-square error [79]:

$$\mathrm{MSE}^i = \left\| \tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}^i \right\|^2 \tag{5.17}$$

Katsaggelos [81] analysed the converge of the general iterative scheme (5.2) based on the contraction theorem which usually serves as a basis for establishing convergence of iterative algorithms. By developing Equation (5.2), we obtain:

$$\mathbf{I} = \beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})\mathbf{I}^i \tag{5.18}$$

According to the contraction theorem the above iteration converges for any initial vector $\mathbf{I}^0$ to a unique fixed point $\mathbf{I}^*$, i.e. a point such that:

$$\beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})\mathbf{I}^* = \mathbf{I}^* \tag{5.19}$$

if $\beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})$ is a contraction. This means that for any $\mathbf{I}_1, \mathbf{I}_2$ the following relation is satisfied:

$$\left\| \left[\beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})\mathbf{I}_1\right] - \left[\beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})\mathbf{I}_2\right] \right\| \leq \eta \left\| \mathbf{I}_1 - \mathbf{I}_2 \right\| \tag{5.20}$$

with $\eta < 1$ and $\|\cdot\|$ denotes any norm. Since $\beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})$ is linear, the sufficient convergence condition results in [81]:

$$\|\mathbf{I} - \beta\mathrm{B}\| < 1 \tag{5.21}$$

Considering the L_2 norm, then condition (5.20) is equivalent [81] to:

$$\max_j |\Sigma_j(\mathbf{I} - \beta\mathrm{B})| < 1 \tag{5.22}$$

where $|\Sigma_j(\mathbf{I} - \beta\mathrm{B})|$ is the absolute value of the jth singular value of $\mathbf{I} - \beta\mathrm{B}$. Finally, the necessary and sufficient condition for iteration (5.2) to converge is that [81]:

$$\max_j |\Lambda_j(\mathbf{I} - \beta\mathrm{B})| < 1 \text{ or } \max_j |1 - \beta\Lambda_j(\mathrm{B})| < 1 \tag{5.23}$$

where $|\Lambda_i(\mathbf{A})|$ is the magnitude of the jth eigenvalue of the matrix $\mathbf{A}$. The problem is that the matrix B is singular, i.e. B has at least one zero eigenvalue, for various typical distortions (e.g. motion blur, defocussing, etc.). Then, there exist no value of β for which the conditions (5.22) and (5.23) are satisfied. In this case $\beta\tilde{\mathbf{I}} + (\mathbf{I} - \beta\mathrm{B})$ may have an infinite number of fixed points. However, the properties of B can be further restricted without loss of generality [81]. If B is a symmetric, semipositive matrix (all of its eigenvalues are non-negative), then the

iteration (5.2) converges to the minimum norm solution of Equation (5.1) if the solution and the projection of $\mathbf{I}_0$ onto the null space of B for $0 < \beta < 2\left\|\mathrm{B}\right\|^{-1}$ exist.

Another approach is a least square approach to the solution of Equation (5.1). There, the following functional is minimised:

$$\min_{\mathbf{I}} \mathrm{MSE}(\mathbf{I}) = \left\|\tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}\right\|^2 \tag{5.24}$$

A necessary condition for $\mathrm{MSE}(\mathbf{I})$ to have a minimum is that its gradient with respect to $\mathbf{I}$ be equal to zero, which results in the equations:

$$\mathrm{B}^{\mathrm{T}}\mathrm{B}\mathbf{I} = \mathrm{B}^{\mathrm{T}}\tilde{\mathbf{I}} \tag{5.25}$$

where the vectors $\mathbf{I}, \tilde{\mathbf{I}}$ represent, respectively, the lexicographically ordered original and blurred image. The matrix B represents the blur.

In [81] it is shown that these equations can be successively approximated according to the iteration:

$$\begin{aligned}\mathbf{I}^{i+1} &= \mathbf{I}^i + \beta\mathrm{B}^{\mathrm{T}}(\tilde{\mathbf{I}} - \mathrm{B}\mathbf{I}^i) \\ &= \beta\mathrm{B}^{\mathrm{T}}\tilde{\mathbf{I}} + (\mathbf{E} - \beta\mathrm{B}^{\mathrm{T}}\mathrm{B})\mathbf{I}^i\end{aligned} \tag{5.26}$$

with:

$$\mathbf{I}^0 = \beta\mathrm{B}^{\mathrm{T}}\tilde{\mathbf{I}}$$

and $\mathbf{E}$ as the identity matrix. This method is similar to the steepest descend algorithm applied to $\mathrm{MSE}(I)$. Thus, other optimisation techniques can be used to minimise the functional (5.25).

If the image formation process is modelled in a continuous infinite dimensional space, B becomes an integral operator and Equation (5.1) becomes a Fredholm integral equation of the first kind. Then the solution of Equation (5.1) is always an ill-posed problem. This means that the unique least-squares solution of Equation (5.1) does not depend continuously on the data, or that the noise in the data results in an unbounded perturbation (noise) in the solution, or that the generalised inverse of B is unbounded [81]. The integral operator B has a countably infinite number of singular values that can be ordered with their limit approaching zero. Since the finite dimensional discrete problem of image restoration results from the discretisation of an ill-posed continuous problem, the matrix B has (in addition to possibly a number of zero singular values) a cluster of very small singular values. The finer the discretisation, the larger the size of B, the closer the limit if the singular values are approximated. Therefore, although the finite dimensional inverse problem is well posed in the least squares sense, the ill-posedness of the continuous problem translates it into an illconditioned matrix B.

The problem of noise amplification can be further explained by using the singular value decomposition of the matrix B. Then, the minimum least-squares solution I^+ of Equation (5.1) can be written as [81]:

$$I^+ = \sum_{j=1}^{r} \frac{1}{\Sigma_j}\langle \mathbf{x}_j, \tilde{\mathbf{I}}\rangle \mathbf{y}_j + \sum_{j=1}^{r} \frac{1}{\Sigma_j}\langle \mathbf{x}_j, \mathbf{V}\rangle \mathbf{y}_j \tag{5.27}$$

where Σ_r are the non-zero singular values of B, with $\Sigma_1 \geq \Sigma_2 \geq \cdots \geq \Sigma_r > \Sigma_{r+1} = \Sigma_{r+2} = ... = 0$, r is the rank of B, $\mathbf{x}_j, \mathbf{y}_j$ are, respectively, the eigenvectors of $\mathrm{B}\mathrm{B}^{\mathrm{T}}$ and $\mathrm{B}^{\mathrm{T}}\mathrm{B}$, and $\langle \cdot, \cdot \rangle$ denotes the inner product. Since B is an ill-conditioned matrix, some of its singular values will be close to zero, thus some of the weights $1/\Sigma_j$ are very large numbers. Then, if the inner product $\langle \mathbf{x}_j, \mathbf{V} \rangle$ is not zero, the noise is amplified.

These methods have weakness due to the following reasons. The noise in Equation (5.1) was ignored in the restoration algorithms presented above. In addition, the solution of Equation 5.1 may become an ill-posed problem as shown above. Therefore, several methods have been proposed to constrain the solution space.

5.1.2 Basic Constrained Methods

A priori knowledge about the solution can be incorporated into the restoration process e.g. bandlimitness of the image or positivity. A convenient way to express such a priori knowledge is to define a constraint operator $\mathbf{C}$ such that [81] :

$$\mathbf{I} = \mathbf{C}\mathbf{I} \tag{5.28}$$

if and only if $\mathbf{I}$ satisfies the constraint. Using such a representation for the constraint e.g. the iteration (5.26) can be written as [81]:

$$\mathbf{I}^{i+1} = (\beta \mathrm{B}^{\mathrm{T}} \tilde{\mathbf{I}} + (\mathbf{E} - \beta \mathrm{B}^{\mathrm{T}} \mathrm{B}))\mathbf{C}\mathbf{I} \tag{5.29}$$

In general $\mathbf{C}$ represents the concatenation of several constraint operators, which are applied at each iteration.

5.1.3 Other Methods

As already stated in the last chapter POCS and regularised methods are a more powerful mean to incorporate a priori knowledge. Thus, they are also popular methods for single image restoration. For example POCS restoration has been proposed in [210, 166, 207, 19] and regularised methods in [112, 82]. The principle of these methods was already presented in the last chapter. As POCS and regularised super-resolution algorithms are in general derived from single image restoration algorithms and the formulation of the problem in the last chapter is more generic (see Sections 4.1.3 and 4.1.4). The fact of considering an image sequence of the size one will result in single image restoration.

5.2 Super-Resolution Mosaic Construction

In this section, we consider the global scheme of Figure 5.2 for the construction of super-resolution mosaics. Similar to the methods presented in Chapters 3 and 4, the mosaic construction starts with the data extraction from the MPEG-1/2 compressed stream. Then, registration is performed and moving objects are extracted. These steps have been presented in Chapter 3. From the moving object detection, label maps are obtained defining the regions

of homogeneous motion in the image. We use them to estimate locally the blur. Therefore, we enhance the blur estimation of Section 4.2.3. At the same time, these mask are used in the luminance correction to exclude moving objects. There is no need to correct illumination of objects as we only insert the objects of the reference frame in the background mosaic and the background regions of other frames are corrected with respect to the reference image. This avoids seams in the background mosaic and adjusts the illumination of the background mosaic to that in the original frame. Thus, there is no illumination change between the mosaic and the inserted objects. The illumination correction was presented in Section 3.2.4. In a next step the background mosaic and the moving objects are restored separately. To do this, we developed a new technique method to super-resolve the background mosaic and a single frame restoration method to increase the resolution of the moving objects. Afterwards, the background mosaic and the moving objects are combined and postprocessing is applied to remove eventual artefacts as described in Section 3.2.6.

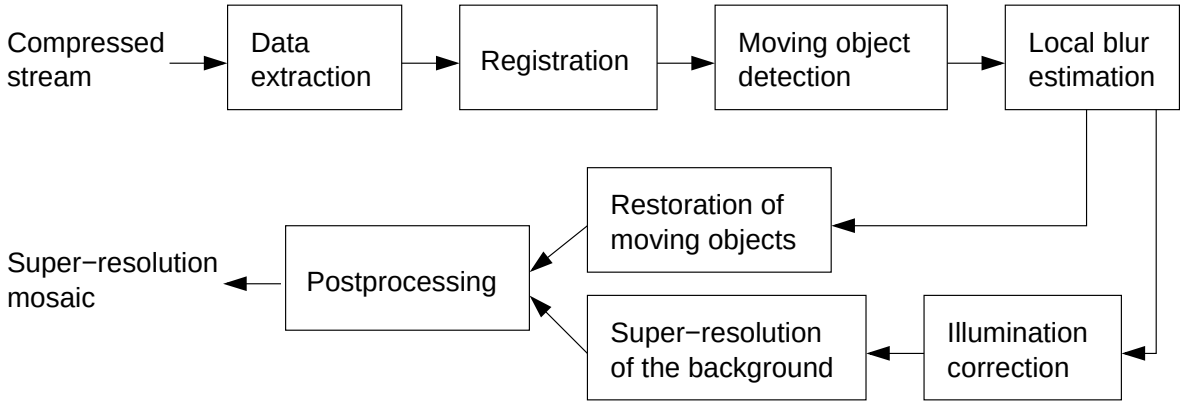


Figure 5.2: Global scheme of the mosaic construction.

Most of these step for the mosaic construction were already presented in Chapters 3 and 4. Thus, we focus in the following on the single frame restoration and super-resolution methods.

5.2.1 Iterative Spatial Domain Restoration Scheme

The restoration and super-resolution methods we propose in this section are based on the iterative restoration method of [31, 78] presented above (see Equation (5.2)). In order to derive from this a super-resolution method for local blurs, we reformulate the degradation model of Equation (5.1) for an arbitrarily shaped region of interest (ROI) considering additionally the local motion and downsampling. Hence, the ROI in low-resolution image G_{roi} can be modelled as:

$$G_{\mathrm{roi}} = \mathrm{S}^{-1}.\mathrm{B}_{\mathrm{roi}} * [\mathrm{T}_{\mathrm{roi}}.F_{\mathrm{roi}} + \mathbf{V}] \tag{5.30}$$

where F_{roi} is the region of interest in the high-resolution frame, $\mathrm{B}_{\mathrm{roi}}$ is the PSF of the ROI, $\mathrm{T}_{\mathrm{roi}}$ is the geometric transformation describing the local motion of the ROI, S^{-1} the down-

sampling operator, $*$ the convolution operator, and $\mathbf{V}$ the noise.

Based on this degradation model, we can derive, from Equation (5.2), a single image iteration scheme to restore and increase the resolution of a region of interest:

$$F^i_{\mathrm{roi}} = F^{i-1}_{\mathrm{roi}} + \mathrm{T}^{-1}_{\mathrm{roi}}.\mathrm{S}\left(G_{\mathrm{roi}} - \mathrm{S}^{-1}\left[\mathrm{B}_{\mathrm{roi}} * \left(\mathrm{T}_{\mathrm{roi}}.F^{i-1}\right)\right]\right) \tag{5.31}$$

where F^i_{roi} is the region of interest in the super-resolution image at the kth iteration and S the upsampling operator. The proof is shown in Appendix A.1.

Let us now consider the case of a video sequence of K low-resolution images where each frame represents the same ROI $G_{\mathrm{roi}}(k), 1 \leq k \leq K$. Then, the following iterative super-resolution algorithm can be derived from Equation (5.31) (see Appendix A.2):

$$M^i_{\mathrm{roi}} = M^{i-1}_{\mathrm{roi}} + \mu(K)\sum_{k=1}^{K} \mathrm{T}^{-1}_{\mathrm{roi}}(k).\mathrm{S}\left(G_{\mathrm{roi}}(k) - \mathrm{S}^{-1}\left[\mathrm{B}_{\mathrm{roi}}(k) * \left(\mathrm{T}_{\mathrm{roi}}(k).M^{i-1}_{\mathrm{roi}}\right)\right]\right) \tag{5.32}$$

where M^i_{roi} is the ROI in the super-resolution mosaic at iteration i and $\mu(\mathbf{p}, K) = \frac{1}{|\mathbf{p}|}$ with $|\mathbf{p}|$ as the number of available pixels at position $\mathbf{p}$. We can notice that this equation is quite similar to the super-resolution method presented in the last chapter (see Equation 4.57), but the restoration operator is absent. Hence, if we can identify the blurring operator, the restoration is straight forward and does not require the synthesis of a restoration filter.

Another problem of the super-resolution method presented in the last chapter is that the blurring and restoration operator are defined for the super-resolution image but the PSF of the super-resolution image is unknown. Our solution was to estimate the blur in a low-resolution image and derive from it the blur for the corresponding super-resolution image. Our aim is now to directly use the low-resolution PSF instead of making an assumption on it for the super-resolution. Thus, it can be shown that (see Appendix A.3):

$$(\mathrm{S}^{-1}.\mathrm{B}) * (\mathrm{S}^{-1}.F) = \frac{1}{\varsigma}\mathrm{S}^{-1}(\mathrm{B} * F) \tag{5.33}$$

This relation means that convolving the low-resolution image with the low-resolution PSF equals $1/\varsigma$ times the downsampled blurred super-resolution image. Thus, $1/\varsigma$ corresponds to a normalisation factor and in case of a normalised convolution mask, which is the case in image processing, this factor can be neglected.

Using this relation we can rewrite the Equations (5.31) and (5.32):

$$F^i_{\mathrm{roi}} = F^{i-1}_{\mathrm{roi}} + \mathrm{T}^{-1}_{\mathrm{roi}}.\mathrm{S}\left(G_{\mathrm{roi}} - \left[(\mathrm{S}^{-1}.\mathrm{B}_{\mathrm{roi}}) * \left(\mathrm{S}^{-1}.\mathrm{T}_{\mathrm{roi}}.F^{i-1}_{\mathrm{roi}}\right)\right]\right) \tag{5.34}$$

$$M^i_{\mathrm{roi}} = M^{i-1}_{\mathrm{roi}} + \mu(K)\sum_{k=1}^{K} \mathrm{T}^{-1}_{\mathrm{roi}}(k).\mathrm{S}\left(G_{\mathrm{roi}}(k) - \left[\left(\mathrm{S}^{-1}.\mathrm{B}_{\mathrm{roi}}(k)\right) * \left(\mathrm{S}^{-1}.\mathrm{T}_{\mathrm{roi}}(k).M^{i-1}_{\mathrm{roi}}\right)\right]\right) \tag{5.35}$$

Hence, the blur is now defined in the low-resolution domain and there is no need anymore to assume a linear relation between the blur in the super-resolution image and the blur in the low-resolution image as in Equations (4.89) and (4.90). Moreover, we obtain a speedup of the super-resolution algorithm as the convolution is now performed in the low-resolution domain instead of in the super-resolution domain. Thus, the spatial domain restoration

(5.34) for a single frame and (5.35) for a mosaic become a good alternative to frequency domain restoration. Note, that $\mathrm{T}_{\mathrm{roi}}^{-1}$ in Equation (5.34) is not used for motion compensation, but for the estimation of the local blur $\mathrm{B}_{\mathrm{roi}}$ as we will show below. The super-resolution algorithm (5.35) can also be used to super-resolve moving objects if motion can be accurately estimated for the object e.g. in the case of rigid object motion.

As we are working with DC images, there is also the problem of aliasing. That is, images can not be exactly superimposed on edges and textures in the super-resolution method. Thus, a strong motion compensation error $G_{\mathrm{roi}}(k) - \left[\left(\mathrm{S}^{-1}\mathrm{B}_{\mathrm{roi}}(k)\right) * \left(\mathrm{S}^{-1}\mathrm{T}_{\mathrm{roi}}(k)M_{\mathrm{roi}}^{i-1}\right)\right]$ results in these regions. This error amplifies along the iterations and causes spurious artefacts in the super-resolution mosaic. Hence, we incorporate the regularisation operator A we developed in Section (4.2.5) in the super-resolution method (5.35):

$$M_{\mathrm{roi}}^{i} = M_{\mathrm{roi}}^{i-1} + \mu(K)\sum_{k=1}^{K}\mathrm{T}_{\mathrm{roi}}^{-1}(k).\mathrm{S}.\mathrm{A}(k)\left(G_{\mathrm{roi}}(k) - \left[\left(\mathrm{S}^{-1}.\mathrm{B}_{\mathrm{roi}}(k)\right) * \left(\mathrm{S}^{-1}.\mathrm{T}_{\mathrm{roi}}(k).M_{\mathrm{roi}}^{i-1}\right)\right]\right) \tag{5.36}$$

Additionally to the advantages of using directly the blur operator and lower computation cost, the restoration of the frame can be locally adapted. Each ROI can be restored with its own blur operator. We use this property to restore moving objects in video and the background separately. This need is straightforward. In the general case, the motion of the camera and the local motion of the objects differ. Thus the blur influence is not the same. In Section (3.2.3) we presented a method to segment moving objects. After the segmentation process, we obtain the label masks O_{l} (see Equation (3.65)). Based on these masks, we define the characteristic function O_{roi}:

$$O_{\mathrm{roi}}(\mathbf{p}, l) = \begin{cases} 1 & \text{if } O_{\mathrm{l}}(\mathbf{p}) = l \\ 0 & \text{otherwise} \end{cases} \tag{5.37}$$

where l is the label of the object. We use this function to define the object and the background ROIs in the low-resolution image:

$$G_{\mathrm{roi}} = O_{\mathrm{roi}}(l).G \tag{5.38}$$

Additionally, we need to determine the object and background ROIs for the high-resolution image F. To this end, we apply a nearest neighbour upsampling operator S' to the characteristic function O_{R}. Thus, the ROIs in the high-resolution image are:

$$F_{\mathrm{roi}} = (\mathrm{S}'.O_{\mathrm{roi}}(l)).F \tag{5.39}$$

To avoid confusion, we denote the objects by roi and the complement set of pixels, the background, by $\overline{\mathrm{roi}}$. In order to determine the maximum number of iterations we have to define a stopping criterion. For the super-resolution method (5.36) we use the error between the original sequence and the "simulated" sequence similar to (4.59). Thus, the error criterion is:

$$\bar{\epsilon}_{\overline{\mathrm{roi}}}^{i} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{M}\sum_{\mathbf{m}}G_{\overline{\mathrm{roi}}}(\mathbf{m},k) - \left[\left(\mathrm{S}^{-1}.\mathrm{B}_{\overline{\mathrm{roi}}}(k)\right) * \left(\mathrm{S}^{-1}.\mathrm{T}_{\overline{\mathrm{roi}}}(k).M_{\overline{\mathrm{roi}}}^{i-1}\right)\right](\mathbf{m})\right) \tag{5.40}$$

The maximum number of iterations is achieved if $\bar{\epsilon}_{\overline{\text{roi}}}$ converges or increases. Here, we assume that the maximum number of iterations is necessary to restore the background mosaic is also sufficient to restore the moving objects. However, it is possible to define a error criterion for the moving objects:

$$\epsilon^i_{\text{roi}} = \frac{1}{M} \sum_{\mathbf{m}} G_{\text{roi}}(\mathbf{m}) - \left[(\mathrm{S}^{-1}.\mathrm{B}_{\text{roi}}) * \left(\mathrm{S}^{-1}.\mathrm{T}_{\text{roi}}.F^{i-1}_{\text{roi}} \right) \right] (\mathbf{m}) \tag{5.41}$$

We will use these error measures to analyse the convergence of the restoration method.

5.2.2 Estimation of Local Blurs

Image blurs can be grouped in two categories, namely spatially invariant or spatially variant blurs. In the class of spatially invariant blurs, the blur model does not vary as a function of position in the observed scene. We assumed this kind of blur model in the last chapter. A blur is called spatially variant if the PSF is a function of position in the observed scene.

At present only few methods have been presented in literature for the estimation and restoration of spatially variant blurs. In the work of Trussel and Fogel [193], the image scene is segmented into regions where the blur is approximated locally to be spatially invariant and the Landweber iterative algorithm is used for restoration. The method presented by Tull and Katsaggelos [196] allows the blur to vary at each pixel location defined by the estimated displacement vector field. The blur size for a pixel is directly derived to be proportional to the displacement vector assuming that the frame rate and acquisition time are known. Restoration is then performed by a regularised iterative algorithm. In [34], Dai et al. presented a method to track motion-blurred targets in video. Local motion blurs are identified by a learning-based scheme and the direction of the motion blur is estimated using steerable filters. Once the blur direction is estimated, a new motion-blurred template is synthesised for a given blur length. Then, the blur size is chosen by a line search minimising a similarity criterion between the best match of the tracker and the synthesised template. Nagy and O'Leary [122] propose a fast restoration algorithm for spatially variant blur. Piece-wise constant blur functions are assumed which are stitched together by interpolation to obtain a continuous blur function and restore the image globally. The algorithm proposed by Bascle et al. [10] estimates the motion blur parameters from the displacement of tracked regions whereas 2D affine motion of the region is considered. Har-Noy and Nguyen [62] propose a block-wise region of interest filtering to restore motion blur. A motion vector is computed for each block in the frame. If then the scale gradient magnitude metric, indicating the blur strength, is greater than a threshold the block is restored.

On the contrary to [34], we already know the regions of local blurs thanks to the object masks O_{l}. We assume like in [193] that the blur in the segmented regions can be locally approximated by a spatially invariant blur model as presented in Section 4.2.3. We estimate the blur size based on the method proposed in Section 4.2.3. Similarly to [196] the motion vector dictates the direction of the blur. For the blur estimation, we distinguish two cases,

namely the blur estimation for the background and the blur estimation for a moving object. Thus, we propose a different adaptation of our blur estimation method for both cases.

In order to estimate the blur parameters of the image background, we only take into account the pixels belonging to the background $G_{\overline{\text{roi}}}$ (see Equation 5.38). for the extraction of significant edges (see Equation 4.75). Furthermore, the estimation of the edge response remains the same using the estimated motion of the camera $\mathrm{T}(k)$. The result is a locally constant PSF of the background induced by the camera motion.

In our case of very low-resolution images, objects are typically represented by only few pixels. Therefore, an edge detection inside the object is not reasonable and we directly use the boundary of the segmented region O_{roi} as significant edges. As we only need a rough guess of the object motion, we used the six-parameter affine model from Equation (3.23) to determine the motion vector at each pixel of the object boundary. Like above the estimation of the edge response and the blur size remain the same.

5.2.3 Computation of the Convolution Kernels

Instead of convolving the ROI with a 1D convolution kernel in motion direction which is a complex and costly, e.g. in [196] bilinear interpolation is used to compute the convolution in motion direction, we propose here to compute a 2D convolution kernel using the x and y-components of the blur size, b_x and b_y (see Equations (4.86)) and (4.87)) and compute then a traditional convolution.

We denote $\mathbf{K}$ as the 2D convolution kernel of the size $K_x \times K_y$. In the case of the Gaussian blur model, we determine the kernel size with respect to the 3σ-property (see Section 4.2.3). For the isotropic Gaussian blur model it is:

$$K_x = K_y = 2 \cdot \lceil 3\sigma \rceil + 1 = 2 \cdot \lceil b \rceil + 1 \tag{5.42}$$

and for the anisotropic Gaussian blur model:

$$K_x = 2 \cdot \lceil 3\sigma_x \rceil + 1 = 2 \cdot \lceil b_x \rceil + 1 \tag{5.43}$$

$$K_y = 2 \cdot \lceil 3\sigma_y \rceil + 1 = 2 \cdot \lceil b_y \rceil + 1 \tag{5.44}$$

where $\lceil \cdot \rceil$ is the ceil operator. Note that we have here the multiplication factor 2 as we did not define b, b_x and b_y symmetrically (see Equations (4.85)–(4.87)).

K_x does not necessarily equal K_y in case of the anisotropic model. Nevertheless, we fix 3×3 as the minimum size of the kernel in both cases. We prefer a slightly larger convolution kernel, therefore also the ceil operator, to avoid a hard cut-off of the convolution kernel. Then, having determined the size of the kernel $\mathbf{K}$, its values $\mathbf{K}(x, y)$ are computed by Equation (4.60) for isotropic Gaussian and by Equation (4.62) for the anisotropic Gaussian. $\mathbf{K}(x, y)$ is normalised afterwards.

In case of linear motion blur the kernel size is computed as:

$$K_x = 2 \cdot \text{round}(b_x) + 1 \tag{5.45}$$

$$K_y = 2 \cdot \text{round}(b_y) + 1 \tag{5.46}$$

$$\tag{5.47}$$

We choose 3×1 or 1×3 as the minimum kernel size depending on whether $b_x > b_y$ or not.

In order to take into account even blur sizes, we set the borders of the kernel to 0.5 as:

$$\mathbf{K}(x,y) = \begin{cases} 0.5 & \text{if round}(b_x) \text{ even and } x = 0 \text{ or } x = K_x - 1 \\ 0.5 & \text{if round}(b_y) \text{ even and } y = 0 \text{ or } y = K_y - 1 \\ 1 & \text{otherwise} \end{cases} \tag{5.48}$$

Then, $\mathbf{K}(x,y)$ is normalised.

It happens that $b_x < 1$ and $b_y < 1$ in our computations since b_x and b_y are, respectively, computed as the average of several estimated values (see Equations (4.83) and (4.84)) Thus, fixing the minimal the kernel size to 3×1 or 1×3 for the linear motion blur means that we make the critical assumption that there is a minimal blur of one pixel in horizontal or vertical direction in the ROI. Using this blur model we can not treat blur sizes smaller than one pixel due to the discretisation as the linear motion blur is a constant function. This is different for the Gaussian blur model. As the Gaussian is a continuous declining function, we can handle blur sizes smaller than 1. This only results in a value near 1 in the center of the convolution kernel and in small values near 0 at the borders.

5.2.4 Convolution of the Region of Interest

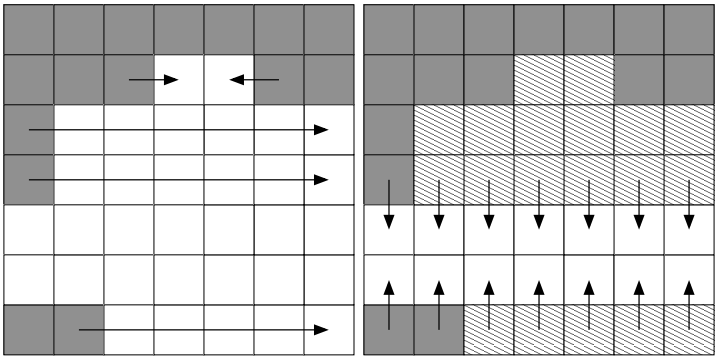
The segmented objects can be of arbitrary form with irregular boundaries. If the convolution kernel acts inside the object where all underlying pixels are determined, the convolution can be performed in the usual way, but the convolution on or near object boundaries needs a special processing. In this case a varying number of pixels underlying the convolution mask might be undetermined as they do not belong to the object. Therefore, we chose MPEG-4-like padding for boundary macroblocks [152] to extrapolate the object in the region of undetermined pixels underlying the convolution mask.

This method first extrapolates pixels of the object boundary horizontally and then vertically as shown in Figure 5.3:

- Undetermined pixels at the object boundary are extrapolated horizontally to fill the undetermined positions in the same row. If a row is bordered by an object pixel at only one side, the value of the nearest object pixel is copied to all undetermined pixel positions. If a row is bordered by two object pixels, the undetermined pixel positions are filled with the mean value of the two neighbouring object pixels. An example of the horizontal padding is shown in Figure 5.3(a).

- If there are still undetermined pixels, object pixels including those filled by the first stage are extrapolated vertically to fill the remaining undetermined pixel positions.

Columns of undetermined pixels with one object pixel neighbour are filled with the value of that pixel and columns with two object pixel neighbours are filled with the mean values of the object border pixels at the top and bottom column. An example of the vertical padding is shown in Figure 5.3(b).



(a) (b)

Figure 5.3: Padding for a convolution kernel of the size 7×7: (a) Horizontal padding, (b) vertical padding.

5.3 Results

In this section, we present the results obtained with the method proposed above. We create a super-resolved background mosaic using the algorithm of (5.36), restore some of the foreground objects by the algorithm (5.34) and combine them with the background mosaic. The resulting mosaic gives an appropriate scene overview for video summarisation. in contrast to Chapter 3 the mosaic here are of higher resolution and contrary to Chapter 4 we are able to process moving objects. Here, we assume also that a zoom factor $\varsigma = 2$ is sufficient for an appropriate visualisation of the mosaics. All sequences presented below were processed according to the global scheme of Figure 5.2. If the processing differs from this scheme, we mention this explicitly.

We first evaluate the convergence of our algorithm using the error measures (5.40) and (5.41) and analyse the results in terms of visual quality and computational times. Then, we compare this method with our frequency domain restoration method of Chapter 3.

Figure 5.5(a) shows the graphs of error measures for the three different PSFs for the restoration of the background mosaic of the sequence "Comportements2". The sequence of DC images is shown in Figure 5.4. The anisotropic Gaussian blur model $\mathrm{B}_{\mathrm{Gauss}}$ minimises the best the error measure $\bar{\epsilon}_{\overline{\mathrm{roi}}}$. The isotropic Gaussian blur model $\mathrm{B}_{\mathrm{Gauss2D}}$ is close to the anisotropic Gaussian, but the result of the linear motion blur PSF $\mathrm{B}_{\mathrm{box}}$ is not satisfying. This is due to the discretisation of the convolution kernel. There, we made the assumption that at least one pixel blur appears in horizontal or vertical direction. As there is only small blur in this sequence ($b < 1$), the blur model is not appropriate and ringing artefacts appear in

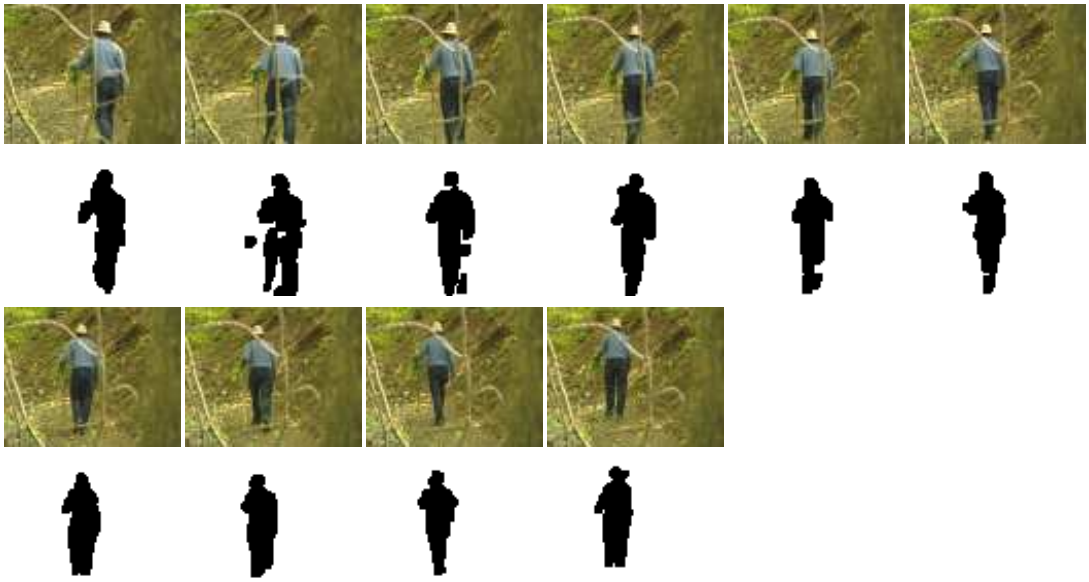
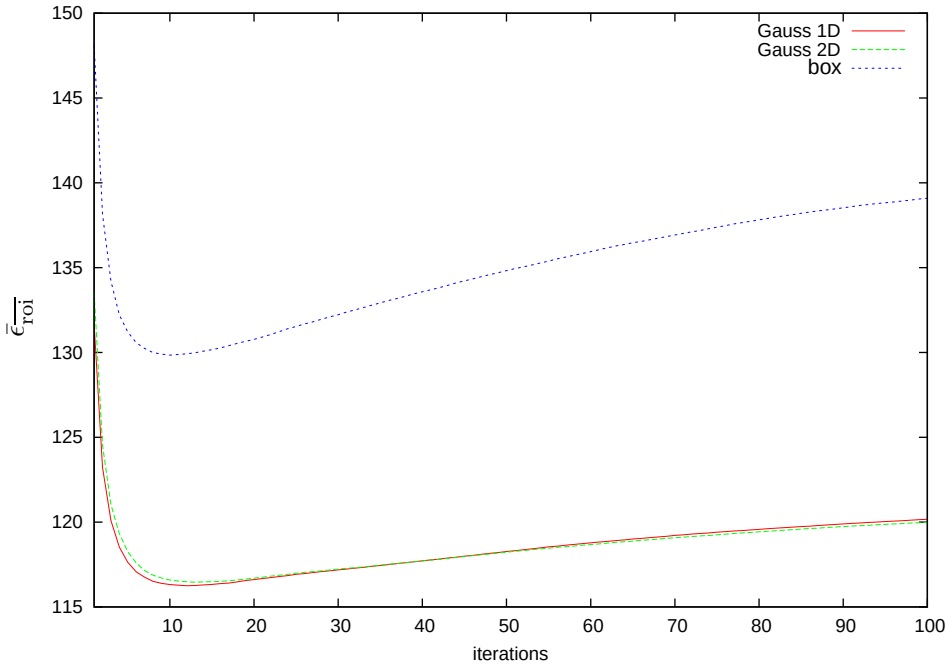


Figure 5.4: Sequence "Comportements2" extracted from the documentary "Comportements alimentaires des hommes préhistoiriques" CERIMES-SFRS® and the corresponding dilated masks $O_{\mathrm{b}}(k)$.

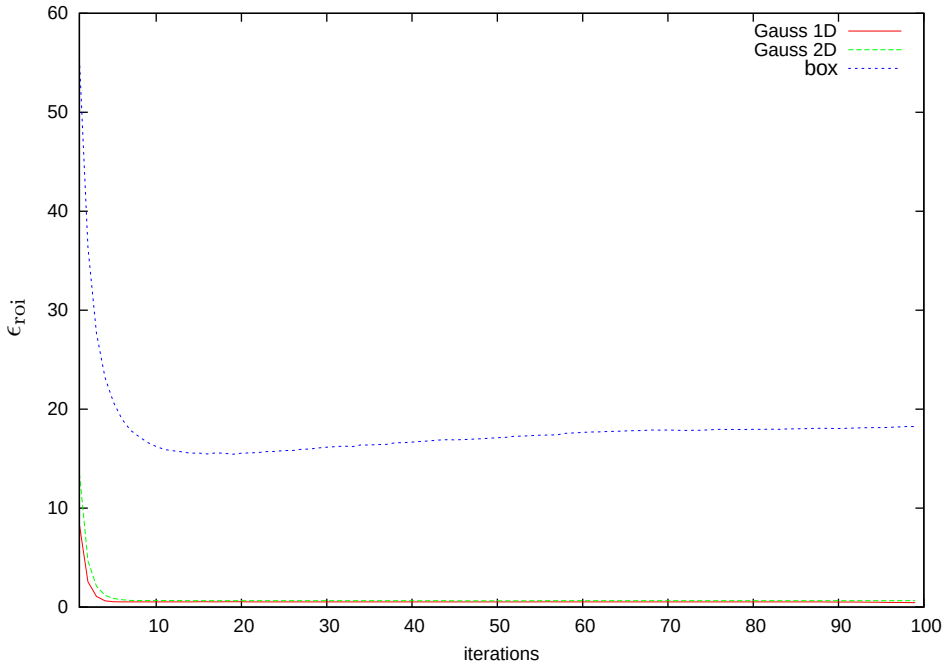
the mosaic. Nevertheless, for all three PSFs the restoration process becomes instable after several iterations and the error $\bar{\epsilon}_{\overline{\mathrm{roi}}}$ increases.

The error measure ϵ_{roi} of the restoration of the object is shown in Figure 5.5(b). We observe the same behaviour of the PSFs: the anisotropic Gaussian performs the best, the isotropic is close, the linear motion blur is not satisfying as it causes ringing artefacts for the same reason than above. Contrary to the restoration of the background mosaic convergence and then stability of the Gaussian PSFs is much better for the object restoration. Figure 5.6 shows the mosaic for isotropic Gaussian PSF (Figure 5.6(b)) and the linear motion PSF (Figure 5.6(a)) after 100 iterations. As expected in the mosaic restored using the linear motion PSF strong ringing artefacts appear. The mosaic determined by the stopping criterion is obtained after 12 iterations for isotropic Gaussian is shown in Figure 5.7(b). We can see that with respect to the initial mosaic (Figure 5.7(a)) (initial background mosaic combined with bilinearly interpolated object) the restored mosaic (Figure 5.7(b)) is much less blurred and more high frequency details appear.

These results are confirmed by other test sequences. For instance, we show the results of the sequence "Comportements1" in Figures 5.8 and 5.9. By processing the sequence '"Hiragasy", we meet a particular problem. We have the same behaviour of the PSFs in Figure 5.10, but the point of increase of $\bar{\epsilon}_{\overline{\mathrm{roi}}}$ is already after 2 iterations (see Figure 5.10(a). This resulting mosaic is shown in Figure 5.11(b). Only less blur was restored during the 2 iterations and we wish to apply further iterations. The background mosaic after 12 iterations is shown in Figure 5.12(a). We observe strong artefacts on the borders of the object masks which causes the increase of $\bar{\epsilon}_{\overline{\mathrm{roi}}}$. However, these artefacts have been removed in the postprocessing step and



(a)



(b)

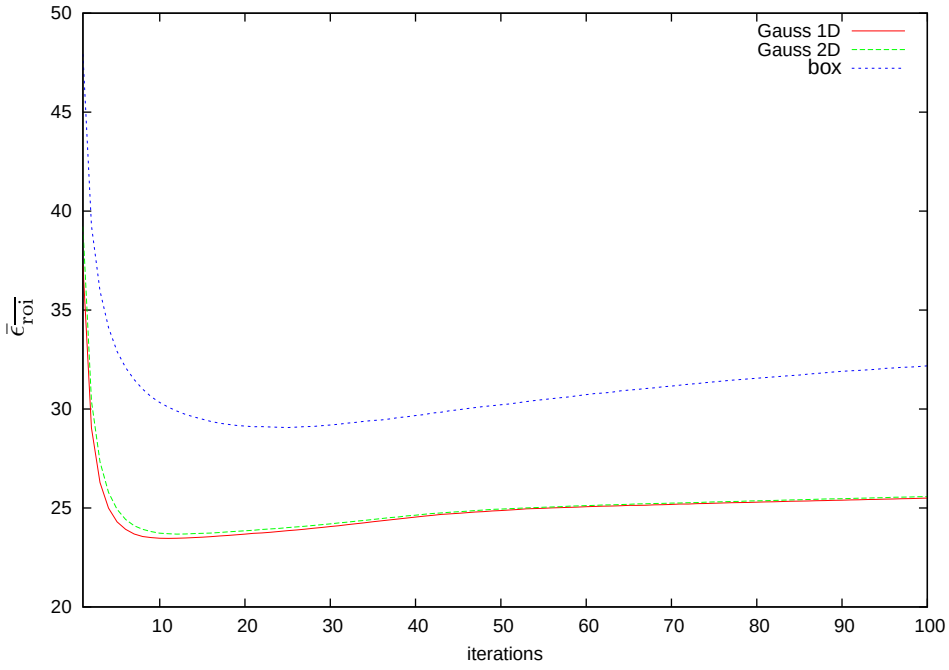
Figure 5.5: The error measure versus the number of iterations for sequence "Comportements2": (a) For the background mosaic, (b) for the moving object.

(a) (b)

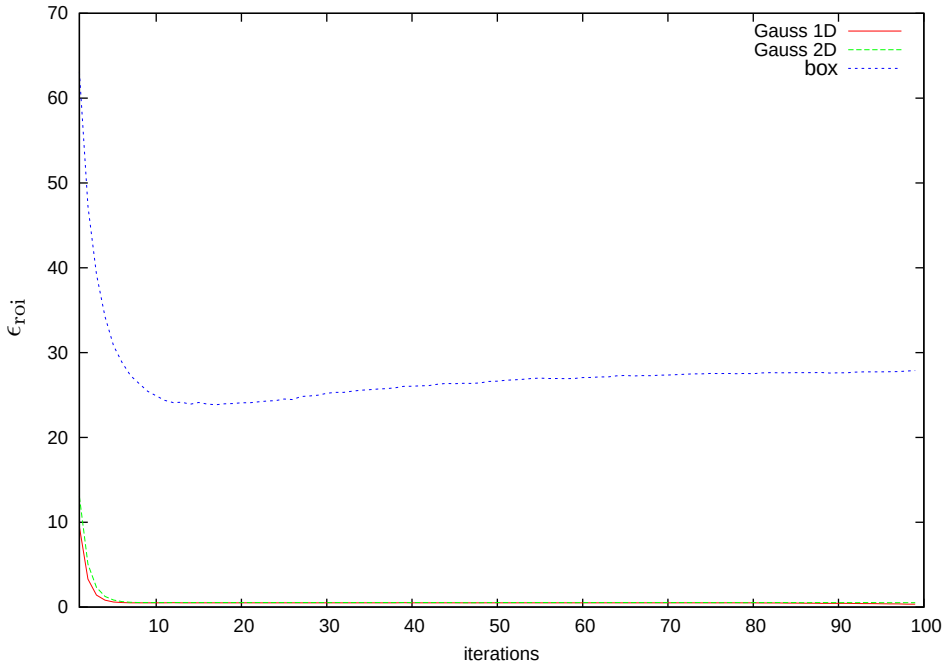
Figure 5.6: Super-resolution mosaics for the sequence "Comportements2" after 100 iterations: (a) Linear motion blur PSF, (b) isotropic Gaussian blur PSF.

(a) (b)

Figure 5.7: Super-resolution mosaic for the sequence "Comportements2" for the isotropic Gaussian blur PSF: (a) The initial mosaic (initial background mosaic combined with the bilinearly interpolated object), (b) the mosaic after 12 iterations.

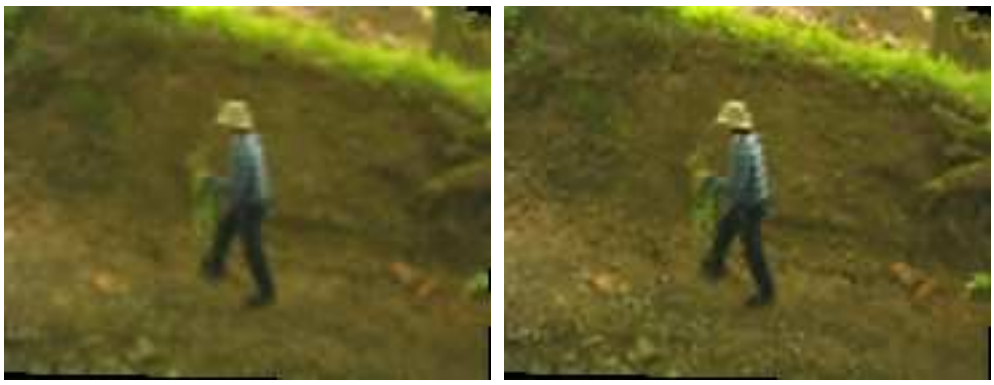


(a)



(b)

Figure 5.8: The error measure versus the number of iterations for sequence "Comportements1": (a) For the background mosaic, (b) for the moving object.



(a) (b)

Figure 5.9: Super-resolution mosaic for the sequence "Comportements1" for the isotropic Gaussian blur PSF: (a) The initial mosaic (initial background mosaic combined with a bilinearly interpolated object), (b) the mosaic after 5 iterations.

a less blurred mosaic results (Figure 5.12(b)). Hence, in cases where the error $\bar{\epsilon}_{\overline{\text{roi}}}$ increases too early additional iterations can be enforced to restore the background mosaic.

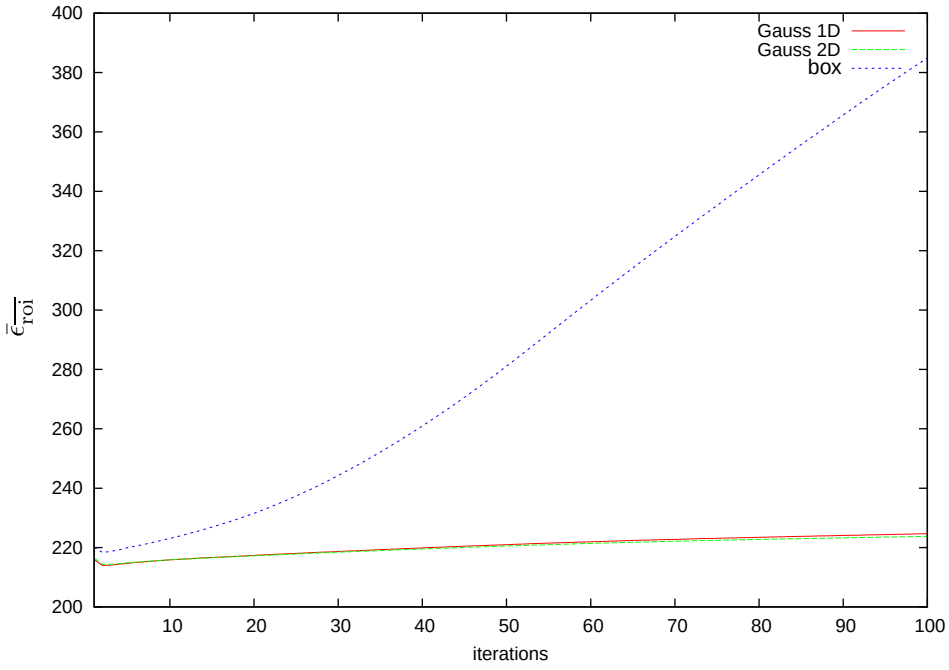
Table 5.1 shows the computational times of the mosaic in Figures 5.7(b), 5.9(b), and 5.12(b). They were obtained on an Intel Pentium 4 3.00GHz processor using a non optimised C++ code and the VXL image library [200]. These computational times are very interesting, as there is no important increase with respect to these for the construction of DC-resolution mosaics. A comparison of computational times is shown in Table 5.2. For the sequence "Comportements1" the additional computational time is only about 9s to increase the resolution by a factor 2, and about 37s for the sequence "Hiragasy". This additional computational time is not linear as it depends on the motion compensations. The larger the camera motion, the larger is the mosaic and the costlier the motion compensation.

Sequence	No. I-frames	No. iterations	(1)	(2)	(3)
Comportements1	6	12	6.161s	12.807s	18.968s
Comportements2	10	12	9.421s	25.387s	34.808s
Hiragasy	12	12	8.326s	47.555s	55.881s

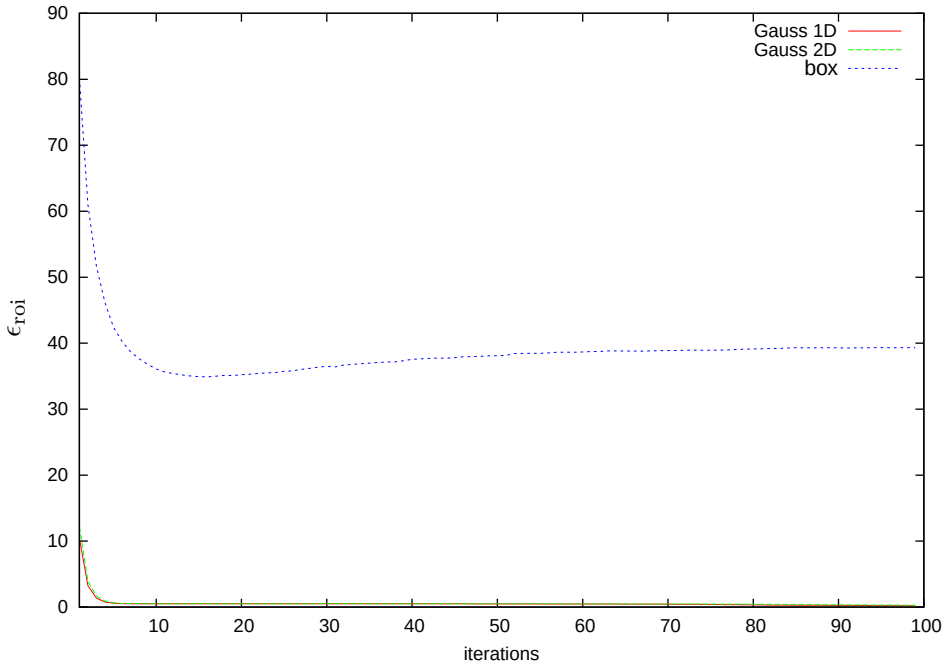
Table 5.1: Computational times for the super-resolution mosaics of Figures 5.7(b), 5.9(b), and 5.12(b): (1) The computational time for data extraction, motion estimation and object extraction, (2) the computational time for concatenation of the motion models in the geometric transformations, illumination correction, blending and postprocessing, (3) the total computational time.

5.3.1 Comparison with the Frequency Domain Restoration

In order to compare the spatial domain restoration method with the frequency domain restoration method presented in the last chapter, we chose two sequences without moving



(a)



(b)

Figure 5.10: The error measure versus the number of iterations for sequence "Hiragasy": (a) For the background mosaic, (b) the moving object.

(a)

(b)

Figure 5.11: Super-resolution mosaic for the sequence "Hiragasy" for the isotropic Gaussian blur PSF: (a) The initial mosaic (initial background mosaic combined with the bilinearly interpolated object), (b) the mosaic after 2 iterations.

(a)

(b)

Figure 5.12: Super-resolution mosaic for the sequence “Hiragasy” for the isotropic Gaussian blur PSF after 12 iterations: (a) The background mosaic before postprocessing, (b) the combined mosaic after postprocessing.

Sequence	(1)	(2)
Comportements1	9.275s	18.968s
Hiragasy	18.348s	55.881s

Table 5.2: Comparison of computational times: (1) The total computational time for the DC-resolution mosaic, (2) the total computational time for the super-resolution mosaic.

objects as the frequency domain restoration was not elaborated for local object restoration. Hence, we compare the restoration of the background mosaic in terms of error measures, visual quality and computational time. Here, we assume that the error measures $\bar{\epsilon}$ (4.59) and $\bar{\epsilon}_{\overline{\text{roi}}}$ (5.40) are comparable as both express the error between the the original sequence and a sequence of simulated images using the estimate of the mosaic at the current iteration.

Figure 5.13 shows a comparison of the error measures for the sequence "Chancre1". The minimal error values are for both restoration schemes around 140 where as the error of the spatial domain restoration method (Figure 5.13(b)) converges faster. The mosaics for both methods after 20 iterations are shown in Figure 5.14. The mosaic restored in the frequency domain (Figure 5.13(a)) seems a little bit sharper than the mosaic restored in the spatial domain (Figure 5.13(b)), but in the same time spurious artefacts appear.

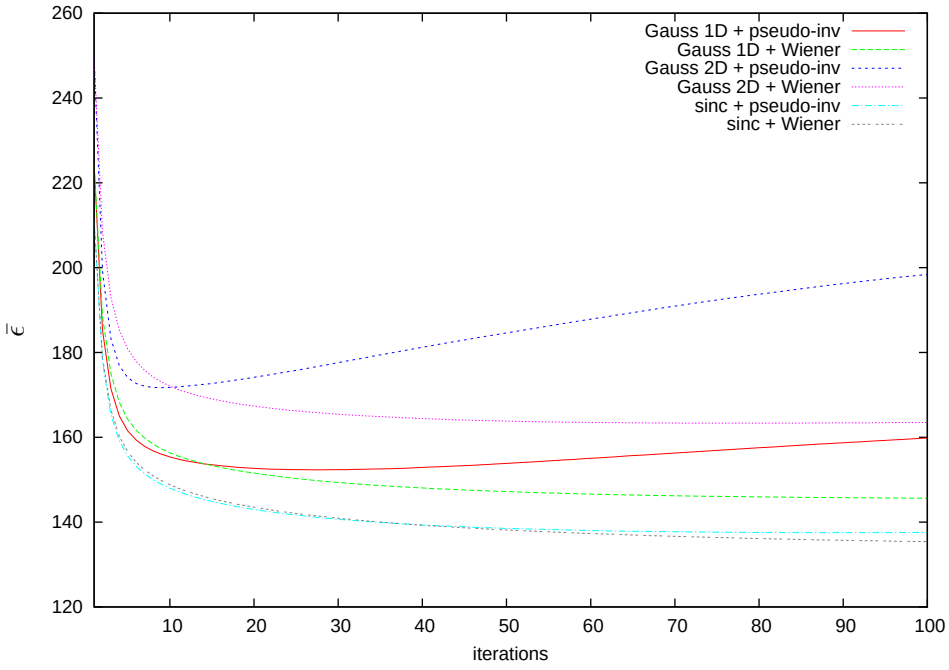
The second sequence we chose for our comparison is the "Chancre3" sequence shown in Figure 5.15. For this sequence the frequency domain restoration results in lower error values than the spatial domain restoration (see Figure 5.16). Like in the example above, the mosaic restored in frequency domain is a little bit sharper and spurious artefacts appear (Figure 5.17(a)) which is not the case for the mosaic restored in the spatial domain (Figure 5.17(b)).

The time figures obtained on an Intel Pentium 4 3.00GHz processor using a non optimised C++ code and the VXL image library [200] are strongly in favour of the spatial domain restoration method (see Table 5.3). Using the spatial domain restoration, we obtain an important gain in computational time almost 10m for both examples. This is due to the fact that padding and successive Fourier transforms are omitted. Moreover, the convolution is now performed at the low resolution instead of at the high resolution which saves additionally computational time.

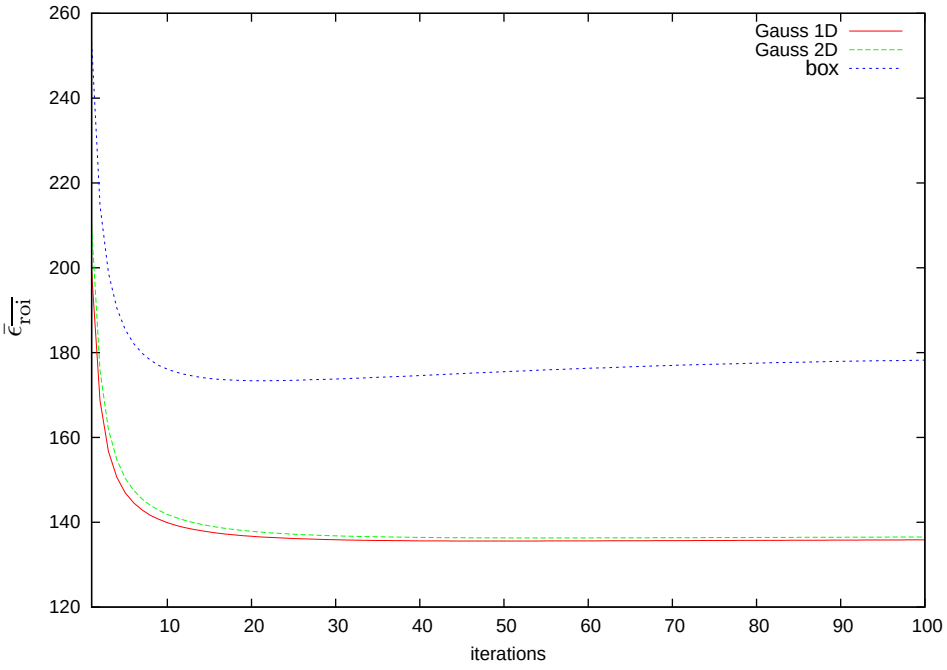
5.4 Conclusion

We presented in this chapter a new super-resolution method which performs restoration in the spatial domain. This method allows taking into account moving objects. Based on a segmentation of moving objects, the background and moving objects are super-resolved separately. As it is often impossible to superimpose moving objects accurately enough for super-resolution, we proposed a new single frame method to increase the resolution and restore the moving objects. Consequently, we enhanced the blur estimation method presented in the last chapter to estimate local blurs in motion direction.

We tested several blur PSFs in our restoration scheme. The best results were obtained for



(a)



(b)

Figure 5.13: The error measure versus the number of iterations for sequence "Chancre1": (a) Frequency domain restoration, (b) spatial domain restoration.

(a) (b)

Figure 5.14: The super-resolution mosaics of the sequence "Chancre1" after 20 iterations: (a) Frequency domain restoration with $\mathfrak{B}_{\mathrm{sinc}}$ and pseudo-inverse filter, (b) spatial domain restoration with $\mathrm{B}_{\mathrm{Gauss}}$.

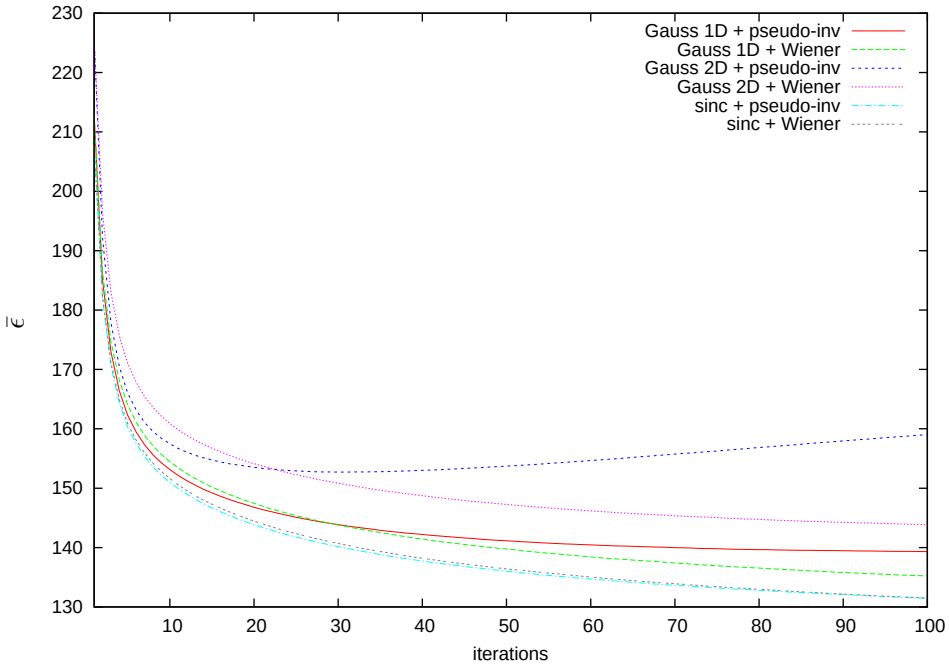
Method	No. I-frames	No. iterations	(1)	(2)	(3)
Chancre1					
Frequency domain	22	20	24.733s	10m54.859s	11m19.592s
Spatial domain	22	20	24.733s	4m28.798s	4m53.531s
Chancre3					
Frequency domain	22	20	19.875s	10m54.926s	11m14.801s
Spatial domain	22	20	19.875s	3m50.631s	4m10.506s

Table 5.3: Computational times for mosaics in Figures 5.14 and 5.17: (1) The computational time for data extraction, motion estimation, (2) the computational time for concatenation of the motion models in the geometric transformations, illumination correction, and blending, (3) the total computational time.

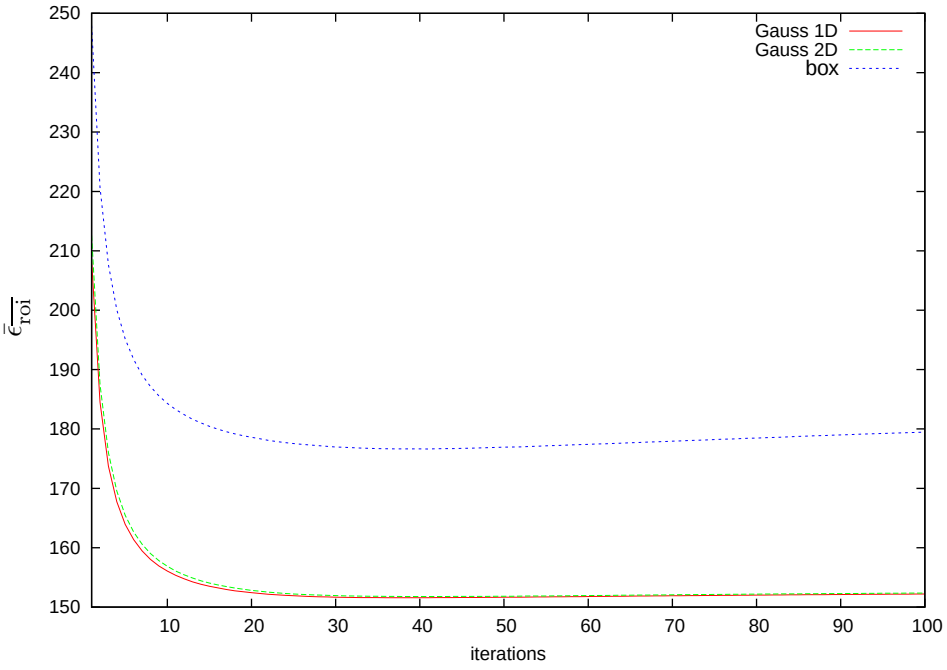
Figure 5.15: The sequence "Chancre3" extracted from the documentary "Comportements alimentaires des hommes préhistoiriques" CERIMES-SFRS®.

the anisotropic Gaussian blur whereas the results for the isotropic Gaussian blur were quite close. Here, the linear motion blur PSF does not seem an appropriate blur model due to the discretisation of the convolution kernel.

This method addresses the shortcomings of the mosaicing methods proposed in Chapters 3 and 4. Contrary to the method of Chapter 3, the resulting mosaics have a higher resolution with only a small increase in computational time. Contrary to the method of Chapter 4, this method allows the processing of local blurs as the restoration is performed in spatial domain. The mosaics restored by the frequency domain method of Chapter 4 seem slightly sharper, but less artefacts appear in the mosaics obtained by the method proposed in this chapter. In comparison to the method of Chapter 4, we obtain an important gain in computational time as successive Fourier transforms are omitted and convolution is performed at low resolution.



(a)



(b)

Figure 5.16: The error measure versus the number of iterations for sequence "Chancre3": (a) Frequency domain restoration, (b) spatial domain restoration.

(a) (b)

Figure 5.17: The super-resolution mosaics of the sequence "Chancre3" after 20 iterations: (a) Frequency domain restoration with $\mathfrak{B}_{\mathrm{sinc}}$ and pseudo-inverse filter, (b) spatial domain restoration with $\mathrm{B}_{\mathrm{Gauss}}$.

Part II

Applications

Chapter 6

Introduction

We presented in Part 1 a complete framework for mosaic construction from MPEG-2 compressed video. The purpose was to create a video summary in a fast way where each shot is represented by a mosaic image. This can then be used for a fast browsing and indexing of the compressed video.

However, several further application areas of the mosaicing and super-resolution methods exist and this is the focus of this part. Our objective is to reuse the methods developed in Part 1 and apply them in other domains. In particular, we address related indexing tasks, namely shot boundary detection and dominant camera motion characterisation, and error concealment.

The detection of shot boundaries is a fundamental task for video indexing. For instance, we assumed for the mosaic construction that shot boundaries have already been detected. Lots of shot boundary detection methods for compressed video have been presented in literature. For instance, Bescós [12] proposes a effective real-time shot boundary detection method based on the application of a combined set of metrics, like the sum of the squared luminance difference and the chrominance likelihood ratio, computed at DC images. In [18], Ćalić and Izquierdo analyse the statistics of the features extracted from the MPEG compressed stream such as the macroblock type for real-time shot boundary detection. Contrarily, the shot boundary detection on I-frames is not the objective per se, but the illustration how intermediary results of the mosaicing method can be used for this task in order to propose an integral framework for video indexing.

Furthermore, this shot boundary detection method was adopted in a system combined with a shot boundary detection on P-frames [38] for the TREC Video Retrieval Evaluation (TRECVID)[1] campaign 2004. Later, we enhanced this system for the TRECVID campaigns 2005 and 2006 and the ARGOS campaign 2006 [2]. Nevertheless, we do not describe further

[1] http://www-nlpir.nist.gov/projects/trecvid/

[2] http://www.irit.fr/recherches/SAMOVA/MEMBERS/JOLY/argos/

developments as this is out of scope of this thesis.

Camera motion is defined as a descriptor of a video segment in the MPEG-7 standard. It expresses which type(s) of camera motion is/are present in the segment among all possible camera motion types. To this end, Bouthemy et al. [16] proposed method for raw video which computes affine global motion and then thresholds likelihoods of motion parameters to robustly characterise camera motion. Based on this, we develop a method to characterise camera motion for compressed video using the affine global motion computed for image alignment in the mosaicing method. We present some result obtained in the TRECVID campaign 2005. Moreover, this methods was used with some modifications in the ARGOS campaign 2006 and within the BBC rushes exploitations tasks for TRECVID 2006. For the same reason than above, we do not describe further developments in this thesis.

When digital video data is transmitted over an error-prone network, parts of the data might be altered or lost during transmission due to channel noise, congestion or other network errors. As a result, the video images suffer from visual errors. Different approaches have been proposed to conceal such errors e.g. by exploiting temporal redundancy in video. Without being exhaustive in the subject, we propose applying our super-resolution mosaicing method at the decoder side to conceal I-frames. A mosaic is constructed during the decoding of a shot. Then, when a transmission error occurs in the current I-frame, the missing information can be replaced by the corresponding regions in the mosaic. Zhao [213] already considered super-resolution for error concealment. However this approach is different from ours, as the frame is concealed by interpolating neighbouring frames and then a super-resolution method is applied afterwards to improve the resolution of the frame.

The remainder of this part is organised as follows: Chapter 7 shows how intermediary results of our mosaicing method can be reused for related indexing tasks. We propose a method for the shot boundary detection on I-frames and method for the characterisation of camera motion. In Chapter 8 we present our method for error concealment using super-resolution mosaicing. In addition, we first study and simulate the transmission errors for MPEG-2 compressed video and compare our results with a spatial error concealment method.

Chapter 7

Related Indexing Tasks

In this chapter, we show how various components of the mosaicing process we developed in this thesis can be applied to video indexing problems. Remaining in the same framework of MPEG-1/2 compressed video, we present here our contribution to shot boundary detection and camera motion characterisation. We will also present some results of the TREC Video Retrieval Evaluation (TRECVID) campaign [175] where we benchmarked our tools.

7.1 Shot Boundary Detection

A shot in post-produced video is a sequence of frames continuously shot by the same camera. Thus, a shot boundary is a discontinuity in the video, so the frames before and after the shot boundary are typically dissimilar. Hence, to identify shot boundaries, the definition of similarity between video frames is of primary importance. In the context of the present research work, we are specifically interested in the detection of shot boundaries in the compressed domain. Therefore, we will briefly present available methods.

Adams et al. [2] use sampled, three-dimensional color histograms in the RGB color space to compare pairs of frames for real-time shot boundary detection. The method of Sugano et al. [181] uses the inter-frame luminance difference and the chrominance histogram matching of DC images. Shen [170] presents a shot cut detection algorithm using Haussdorf distance histograms. The Haussdorf distance is obtained by comparing edge points of successive frames, wherein the edge information is directly extracted from compressed frames. Ćalić and Izquierdo [18] analyse the statistics of the features extracted from the MPEG compressed stream such as the macroblock type for real-time shot boundary detection. Haoran et al. [61] consider the dissimilarity between frames with respect to the type of macroblocks and incorporate camera motion with the aim of removing false detections. The real-time method of Lefèvre et al. [99] acts on a frame-to-frame difference of DC images. In order to avoid

illumination change effects, the colour space is changed from RGB to HSV, where H and S are used to compare two successive frames. Bescós [12] achieves a real-time operation using an algorithm based on the application of a combined set of metrics, like the sum of the squared luminance difference and the chrominance likelihood ratio, computed at DC images. Ewerth and Freisleben [44] propose an adequate frame difference normalisation for the shot boundary detection. In [46], Farag and Abdel-Wahab use a neural network to detect shot boundaries in a DC image sequence. Summarising these methods, we state that a lot of research has been done using partial information extracted from compressed streams such as DC images.

In this work, we are interested in a fast and approximate segmentation and interpretation of multimedia content, which we call "rough indexing" paradigm [105]. Rough indexing means to segment and interpret the video content in a fast, approximate and user oriented way. When the target application of video content analysis requires a fine decision, it is not only necessary to detect e.g. a shot boundary or an object of interest, but also to qualify the transition effects (progressive or abrupt) between shots or to determine precisely the object shape in raw video. Such an analysis is very much required for professional applications. On the contrary to the multimedia professional, common users are not interested in details such as the type of transition between shots in the video production process. For content interpretation, they need some high level and very approximate indexes which can be deduced from signal features. Relatively to the shot boundary detection task, it means that only the fact of a change should be detected and not its character.

In this section we show how the registration process of our mosaicing method (see Section 3.2.2), together with the similarity measure we introduce, allows for the shot boundary detection on I-frames in MPEG-1/2 compressed video. The method we propose for the detection of shot changes on I-frames is combined with a shot change detection on P-frames [38].

Instead of working with original I-frames and decoding the whole image data, we work in the context of rough indexing paradigm and extract only the luminance DC image for each I-frame. We discussed the extraction of DC images in Chapter 2. Then, the principle of our method is to compare DC images of successive I-frames as shown in Figure 7.1. If their content is similar (we compare the pixels of both images), then they are a part of the same shot. Otherwise, a shot boundary appears. However, if motion occurs between successive I-frames, then successive I-frames show a slightly different cut-out of the scene content. Therefore, we have to take this motion into account, i.e. we compensate the DC image of the previous I-frame by motion in order to superimpose it with the DC image of current I-frame. This requires the computation of the global motion trajectory of the whole sequence as it is the case in the registration process of mosaicing method (Section 3.2.2). Global motion defines the motion of the main scene content and is principally due to the movement of the camera or the change of focus. Therefore, the global motion parameters for P-frames are calculated by the method presented in Section 3.2.2. This motion information is not sufficient to calculate the whole trajectory. Since motion parameters are needed for I-frames as well, which do not contain motion information, we extrapolate the motion parameters of the

P-frames in a GOP by a linear regression (see Section 3.2.2). This permits to calculate the motion parameters for the current I-frame. Thanks to this global motion information, the geometric transformation of the DC image of the current I-frame and the DC image of the previous I-frame can be computed by concatenating and scaling the global motion models between them (see Section 3.2.2). Comparing the corresponding pixels of the DC images, two consecutive I-frames are matched and their similarity is measured. These values are then used for shot boundary detection on I-frames.

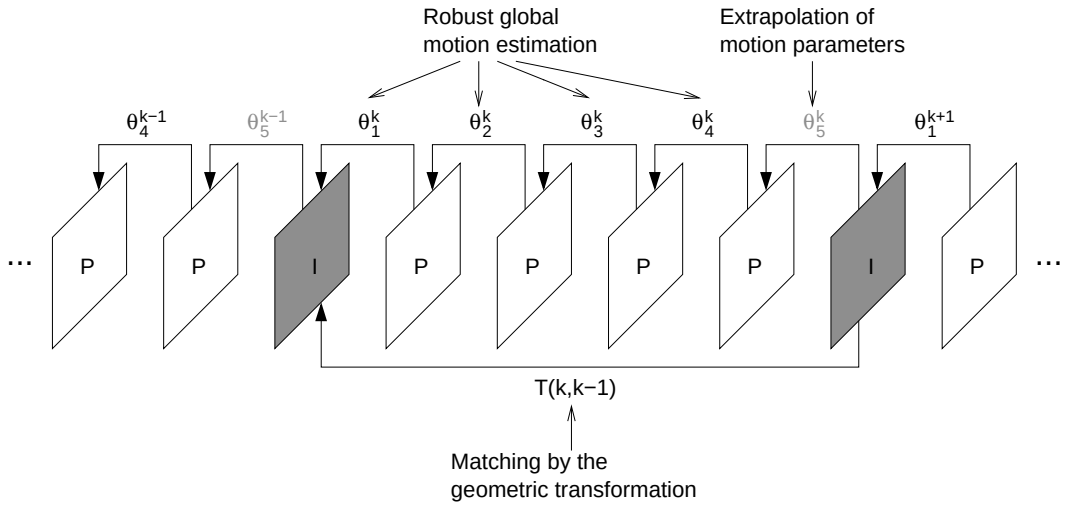


Figure 7.1: Overview of the shot boundary detection on I-frames.

7.1.1 Matching of I-Frames

In order to measure the similarity of consecutive I-frames in video taking into account the motion between these frames, we have now to compensate the DC image of the $k-1$th I-frame, $DC(k-1)$, onto the DC image of kth I-frame, $DC(k)$. Therefore, we compute the geometric transformation $\mathrm{T}(k,k-1)$ (see Equation (3.61)). In the computation of this geometric transformation, when compensating recursively blocks of the kth I-frame onto the P-frames in a GOP, the situation can be encountered that a block is compensated onto a position in a P-frame which corresponds to a motion outlier (see Section 3.2.2). In this case further projection is not possible since motion information for this position is not valid. Thus pixels belonging to such blocks will not be further considered for the I-frame matching. On the contrary, a block whose new position is outside of an P-frame is not rejected because after a few translations it may be found again inside the $k-1$th I-frame. This is illustrated in Figure 7.2.

The motion-compensated DC image of the $k-1$th I-frame, $\widehat{DC}(k-1)$, and $DC(k)$ are compared in order to measure the similarity of two successive I-frames. $\widehat{DC}(k-1)$ is obtained by applying the geometric transformation $\mathrm{T}(k,k-1)$, but the coordinates of the motion-compensated pixels can be decimal. This can be due to either an estimated decimal motion vector or the scaling of the global motion models. As a solution, we propose a bilinear

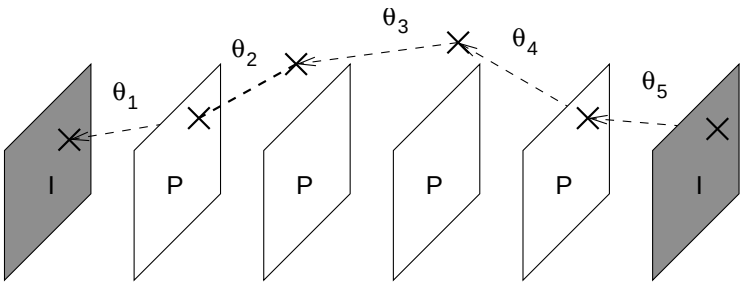


Figure 7.2: Projection of an block: After 2 translations the block is outside the frame, but after 2 further translations it return inside the frame and can finally be found in the previous I-frame.

interpolation to calculate the luminance value at the decimal coordinates according to Equation (2.14).

Figures 7.3 and 7.4 respectively show one scheme of the I-frame motion-compensation in the DC image domain. Figure 7.3 visualises a GOP without shot boundary whereas in Figure 7.4 a cut appears on the kth I-frame. The white boundary at the top of $\widehat{DC}(k-1)$ is caused by a tilt up. Thus, this part of the image is not present in $DC(k-1)$. The white pixels covering the people in $\widehat{DC}(k-1)$ correspond to outliers. In this scene, the camera tracks the people climbing the hill, but it does not move at the same velocity than the people. Thus, the motion vectors of the people differ from the motion vectors of the background and they are rejected during the robust motion estimation (see Section 3.2.2). The white boundary in Figure 7.4 at the right of the projection image is due to a pan right.

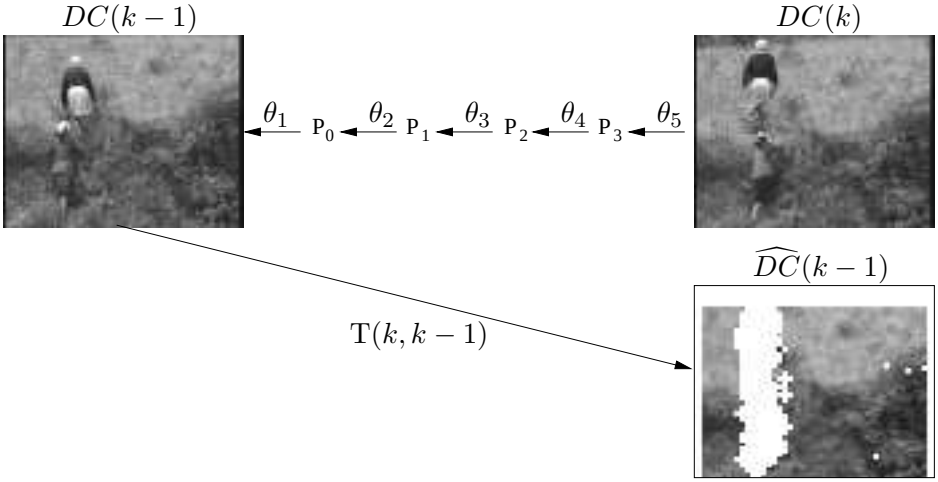


Figure 7.3: Warping of the frame $DC(k-1)$ onto $DC(k)$ without cut extracted from the documentary "Hiragasy" CERIMES-SFRS®.

Due to the rejection of a block in the case it meets a motion outlier when recursively projecting it from the current I-frame to the previous I-frames, we obtain an outlier mask, e.g. the white pixels covering the people in Figure 7.4, which is similar to the object masks

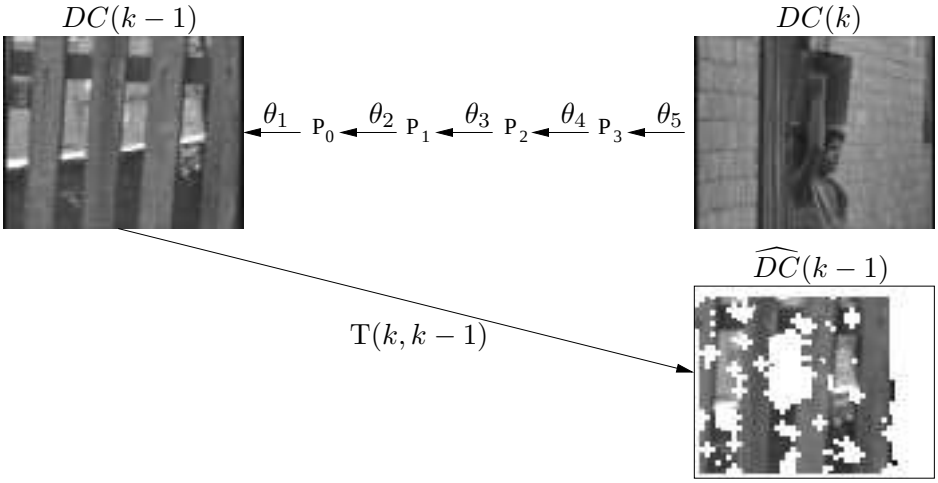


Figure 7.4: Warping of the frame $DC(k-1)$ onto $DC(k)$ with shot cut extracted from the documentary "Hiragasy" CERIMES-SFRS®.

(3.66) we used in the mosaic construction. The computation of the object masks uses also the motion outliers in P-frames, but includes a color segmentation of the DC images of I-frames and are thus more accurate. Hence, the object masks could be used instead, but their computation is more expensive.

Comparing the motion-compensated image $\widehat{DC}(k-1)$ and the image $DC(k)$ in Figure 7.4 they are visually different. This is opposite to Figure 7.3 where the images $\widehat{DC}(k-1)$ and $DC(k)$ seem to be quite similar disregarding the outlier pixels. Then, we will present a method to measure the similarity.

7.1.2 Similarity Measure for Matching DC Images of I-Frames

Here, we address the key issue in the whole method, after matching the DC images of I-frames we have to decide whether they are similar or not. The usual way to measure similarity between two images consists in computing the MSE (3.46) between $\widehat{DC}(k-1)$ and the image $DC(k)$ [174].

Under the assumption of ideal motion and the absence of noise, this value should be zero for the same content of frames. Nevertheless, the computation of this measure for DC frames is strongly corrupted by high frequency noise observed in DC frames due to subsampling effects on textured areas or on contrasted edges as we stated in Chapter 3. This is why we introduce a new similarity measure which privileges the contribution of flat areas and limits the contribution of textured areas. To this end, we weight the MSE by the inverse DC image gradient. The objective is to compare flat regions and not edges or textured areas.

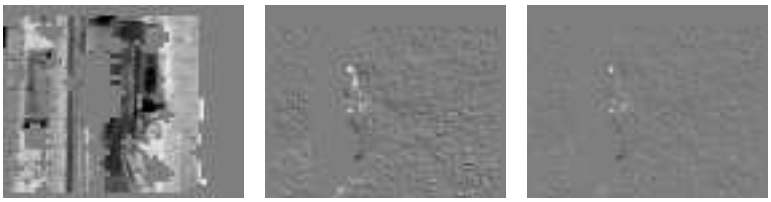
For the sake of low computational costs and due to the low resolution of DC frames we use the Roberts gradient operator as we did in the regularisation operator for the mosaic restoration (4.100). The gradient is computed just on the pixels of the frame $DC(k-1)$, contributing to the interpolated pixel value (see Equation (3.70)). The weighted MSE (WMSE)

associated to the pair of I-frames $(k, k-1)$ is:

$$\text{WMSE}(k) = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{p} \in \mathcal{V}} \Psi(\|\nabla DC(\hat{\mathbf{p}}, k-1)\|)^2 \cdot (DC(\mathbf{p}, k) - \mathrm{T}(k, k-1).DC(\mathbf{p}, k-1))^2 \quad (7.1)$$

where $\mathcal{V}$ is the set of the valid pixels and $|\mathcal{V}|$ its cardinality. Here, valid pixels are motion inlier pixels and pixels corresponding to blocks whose motion-compensated position is inside the frame $DC(k-1)$ only. Ψ is the penalty function of the regularisation operator of Equation (4.99) computed on the magnitude of the Roberts gradient $\|\nabla_{\mathrm{r}} DC(\hat{\mathbf{p}}, k-1)\|$ (3.70) with $\hat{\mathbf{p}}$ as the motion compensated position of $\mathbf{p}$ in $DC(k-1)$. The influence of the penalty function Ψ is determined by the threshold λ_{A} (see Equation 4.99). The smaller the threshold is, the less of contrast is accepted in the definition of a flat area. When the threshold equals 1, only constant zones in the image $DC(k-1)$ are exactly in MSE sense compared with their corresponding zones in the image $DC(k)$.

Figure 7.5 shows the error images for the examples of Figures 7.3 and 7.4. A medium grey indicates that there is no difference between the compared pixels. The more the pixel values differ from a medium grey, the stronger is the error magnitude. Thus, the error image (Figure 7.5(a)) obtained by comparing $\widehat{DC}(k-1)$ and $DC(k)$ from Figure 7.4 is rich in contrast. The error image of Figure 7.5(b) of the sequence without shot boundary from Figure 7.3 is more uniform. Anyhow, the textured region of the background is still quite rich in contrast. On this account, the error image 7.5(c) has been weighted by the penalty function (4.100) with $\lambda_{\mathrm{A}} = 12.75$. This threshold corresponds to 5% of the maximum absolute error which is 255 in our case. The resulting error image is much more smoother.



(a) (b) (c)

Figure 7.5: Visualisation of the error between the images $\widehat{DC}(k-1)$ and $DC(k)$: (a) The error corresponding to Figure 7.4, (b) the error corresponding to Figure 7.3, (c) the weighted error corresponding to Figure 7.3 with $\lambda_{\mathrm{A}} = 12.75$.

7.1.3 Shot Boundary Detection Method for I-Frames

Now, we will apply this new similarity measure to the fundamental problem of shot boundary detection. Having calculated the WMSE values for the whole sequence using Equation (7.1), a histogram over the WMSE values can be computed to detect shot changes on I-frames. A shot change on an I-frame causes a high WMSE value and if no shot change appears the WMSE value is small. Then, a threshold λ_{c} can be defined on the x-axis separating

the WMSE values in the high range associated to shot changes from the others without shot change (see Figure 7.7). In the ideal case, the threshold corresponds to the typical percentage of the real shot changes with respect to the pairs of I-frames to compare.

If shot changes are only detected on I-frames, the shot boundary location error can be up to the size of one GOP. In order to reduce it, we used a specific shot boundary detector for P-frames [38]. If a shot change appears between two I-frames, on the one hand a P-frame is generated with a strong motion compensation error and on the other hand a break in motion occurs. In this case the motion estimation fails and an inaccurate geometric transformation results. Although, the two I-frames, respectively showing a different scene content, are matched by the incorrect geometric transformation as illustrated in Figure 7.6. This causes a high WMSE value and the shot boundary will be detected on the second I-frame $DC(k)$. Thus, if the change was already detected on the P-frame, an overdetection occurs. Therefore, the pair of surrounding I-frames will not be considered for shot boundary detection and we remove the corresponding WMSE values from the histogram.

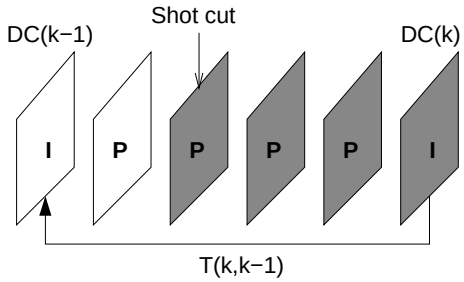


Figure 7.6: Matching of two I-frames in case if a cut appears on a P-frame between them.

Figure 7.7 shows a histogram for the MPEG-2 compressed documentary "Hiragasy" CERIMES-SFRS®. This film consists in 1381 pairs of I-frames with 19 shot changes on I-frames whereas 81 shot changes on P-frames has been removed from the histogram. The WMSE values corresponding to the ground truth, i.e. shot changes on I-frames, are marked on the top of the figure. The WMSE values of I-frames with shot change are in reality not absolutely separated from the others without shot change. We get a zone of mixture in the high range of the histogram values where WMSE values corresponding to both, pairs of I-frames corresponding to shot changes and without shot changes, can be found. In order to get a low overdetection rate, our aim is to determine the threshold λ_{A} (4.100) causing the smallest zone of mixture.

Our similarity measure will not be efficient in the case of slow progressive changes on shot boundaries, as the content of successive I-frames risks to be very similar. Thus, just on the left in the histogram a dissolve is situated. It causes a too small WMSE value and can not be detected with this method.

7.1.4 Results

To evaluate the results of shot change detection based on the proposed similarity measure, we use the usual recall and precision metrics with respect to a ground truth preliminary

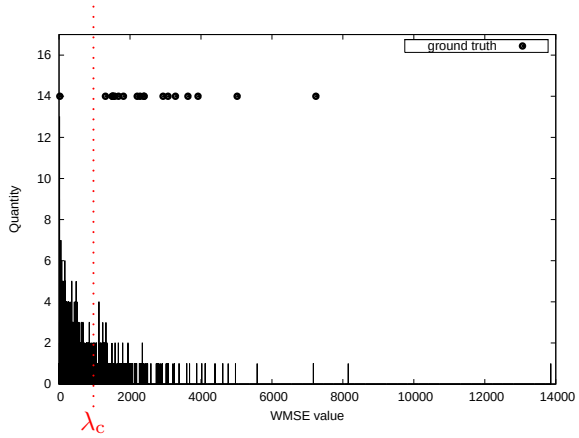


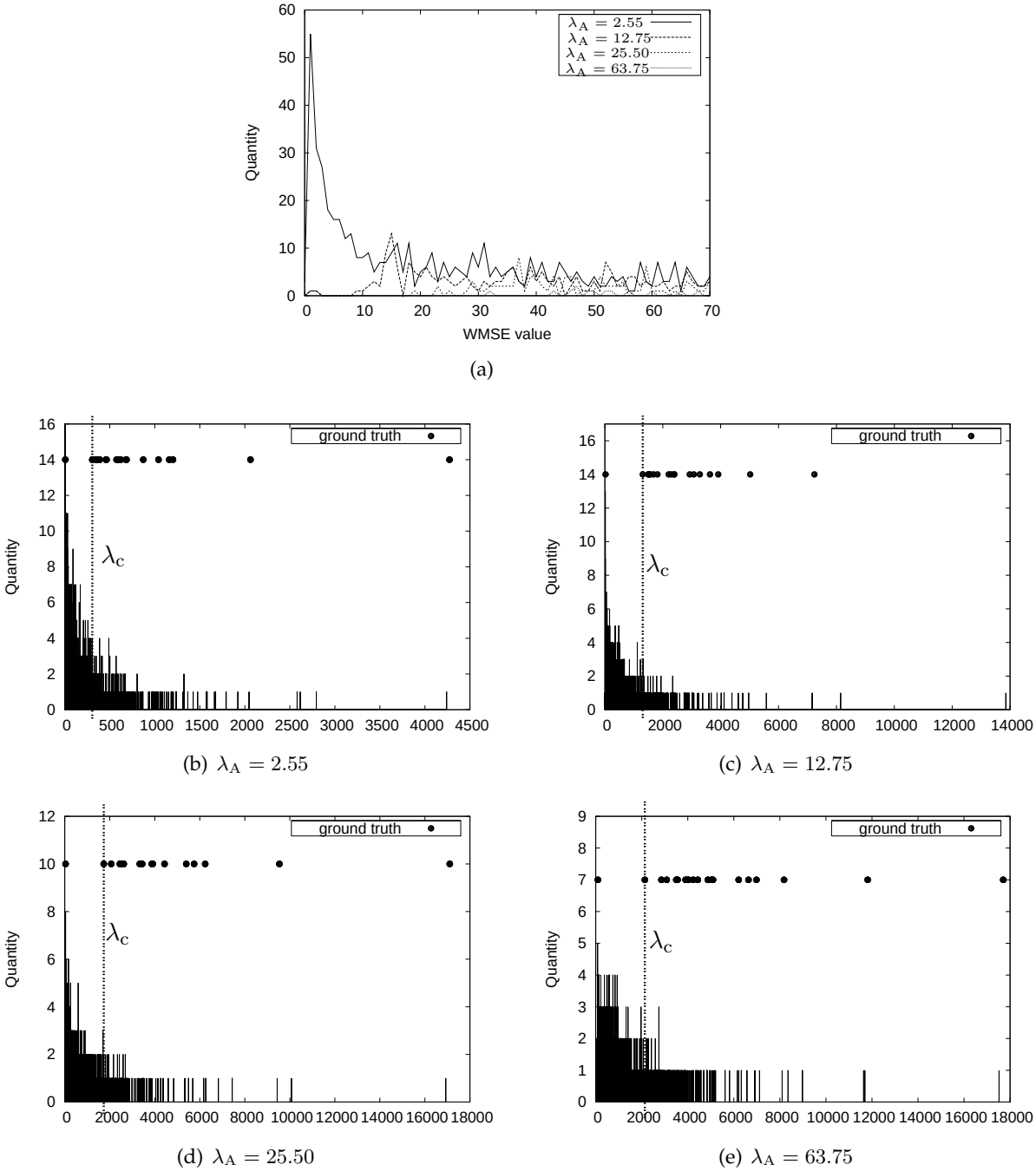
Figure 7.7: Histogram over the WMSE values ($\lambda_A = 12.75$) of the film "Hiragasy" CERIMES-SFRS®.

defined by a user. The ground truth consists of the frame numbers of all shot boundaries on I-frames. Here, $Recall = \frac{c}{gt}$ is the percentage of how many shot boundaries have been detected and $Precision = \frac{c}{d}$ indicates how many overdetections are done, where c is the number of the correct detected shot boundaries, gt the number of the ground truth and d the number of all (correct and false) detected shot boundaries. Recall and precision are both measured in the range of $[0, 1]$, whereas values near to 1 indicate a good result.

The evaluation of the results in the context of rough indexing paradigm is slightly different from classical benchmarking schemes such as used in TRECVID campaign for shot boundary detection e.g. see [2, 181]. In the case of an "exact" evaluation, not only the presence of a shot boundary is checked, but also the type of transition (cut or progressive) and the length of progressive transition. In the rough indexing paradigm, as we state above, the type of transition does not matter. The essential is the detection of a shot change whatever its nature is. In this case the ground truth of shot boundaries is constituted of all changes (cuts and progressive ones). As we separately developed a detector of changes on P-frames [38], here only shot boundaries on I-frames and neighbouring B-frames are included in the ground truth.

The method presented above has been tested with different thresholds λ_A for the weighting function (4.100). It is chosen corresponding to 1%, 5%, 10% and 25% of the maximum absolute error which is 255 in our case. In Figure 7.8 the histogram curves for the different values of the threshold λ_A and the corresponding choice of the classification thresholds λ_c (dotted line) are shown for the same film. Table 7.1 shows the corresponding results evaluated with the measures of recall and precision. The classification threshold λ_c is chosen empirically according to the smallest WMSE values observed on shot cuts in various films. If the WMSE value of an I-frame is higher than λ_c, a shot change is detected on this frame.

In the case of $\lambda_A = 2.55$, that is 1% of the maximum absolute error, the similarity measure (7.1) is strongly generalised by the local contrast in images. This leads to the homogenisation of change and not a change cases. For $\lambda_A = 12.75$ and $\lambda_A = 25.5$ which corresponds respectively to 5% and 10% of the maximum absolute error we obtain the best results. According



(a)

(b) $\lambda_A = 2.55$

(c) $\lambda_A = 12.75$

(d) $\lambda_A = 25.50$

(e) $\lambda_A = 63.75$

Figure 7.8: WMSE histograms with different weighting thresholds λ_A for the film "Hiragasy" CERIMES-SFRS® : (a) the behaviour of the WMSE as a function of λ_A, (b)-(e) the different choices of λ_c according to λ_A (see Table7.1).

λ_A	gt	λ_c	d	c	*Recall*	*Precision*
2.55	20	306.56	293	19	0.950	0.065
12.75	20	1307.57	149	19	0.950	0.128
25.50	20	1734.87	168	19	0.950	0.113
63.75	20	2131.13	221	19	0.950	0.086

Table 7.1: Results for the film "Hiragasy" CERIMES-SFRS®.

to the classical edge detection methods, the most of the edges are detected with a threshold on gradient norms corresponding to 10% of the dynamic range. Then, when the threshold is chosen smaller than 10% of the dynamic range, edges in images and textured areas are penalised and not flat areas. If it is chosen higher, the penalisation will concern only highly contrasted edges which can be very rare on these frames. In this case the behaviour of this similarity measure will converge to that one of the MSE (3.46).

Table 7.2 shows a summary of the recall and precision results we obtained on several documentaries (each of them being of 13 min duration) using the same classification thresholds λ_c than in Table 7.1, i.e. $\lambda_c = 1307.57$ for $\lambda_A = 12.75$ and $\lambda_c = 1734.87$ for $\lambda_A = 25.50$. The column gt/all indicates the ground truth i.e. transitions on I-frames (cuts and progressive ones) with respect to all transitions in the film. It comes out that the precision is quite low for some films. Furthermore, the general trend as a function of λ_c and λ_A is difficult to identify. In fact, it strongly depends on the penalty function Ψ and on the classification threshold as well.

A characterisation of these parameters with respect to the nature of the film may ameliorate the precision results. According to our observations, in highly textured videos even the weighted MSE is strong and thus the distinction between a shot change and a continuous content is difficult.

Another question arises on the precision of the location of a shot boundary in the rough indexing paradigm. Obviously, the high frame precision is interesting for a content with very short shots. In the case of an artistic content it happens in the production credits and cast part. This is usually not relevant for the analysis of the main part of a content item. If the method for shot boundary detection on I-frames we propose in this paper is combined with a specific detector for P-frames such as [38], then the absolute error in location of cuts in a video is limited by the number of B-frames between an I-frame and the closest P-frame. Otherwise, the precision of the location is limited by the length of the GOP. In the case of dissolves, the question is ill-posed for the rough indexing paradigm since we are not interested in the localisation of dissolve boundaries, but in a content change. This detection is difficult when successive I-frames are very similar (low dissolve).

To compute the similarity (7.1) between the frames $DC(k-1)$ and $DC(k)$, we use a bilinear interpolation of the frame $DC(k)$. This choice is typical for standard schemes of motion estimation with a subpixel accuracy. It could seem "too precise" in a rough indexing paradigm. Thus, we show some results of comparison with the roughest interpolation scheme "closest integer" in Table 7.3. The detection results are slightly better if the bilinear

Film	gt/all	λ_{A}	*Recall*	*Precision*
Hiragasy	19/81	12.75	0.950	0.128
		25.50	0.950	0.113
La Joueuse de Tympanon (I)	25/134	12.75	0.730	0.633
		25.50	0.730	0.650
La Joueuse de Tympanon (II)	62/77	12.75	0.571	0.782
		25.50	0.556	0.814
Aquaculture (I)	7/86	12.75	0.750	0.171
		25.50	0.750	0.214
Aquaculture (II)	11/82	12.75	0.500	0.452
		25.50	0.500	0.727
Nosy Hira	12/75	12.75	0.769	0.076
		25.50	0.769	0.072

Table 7.2: Summary of the recall and precision results for some documentaries CERIMES-SFRS®.

λ_{A}	gt	λ_{c}	d	c	*Recall*	*Precision*
2.55	20	306.56	287	18	0.900	0.062
12.75	20	1307.57	151	19	0.950	0.125
25.50	20	1734.87	175	19	0.950	0.109
63.75	20	2131.13	233	19	0.950	0.085

Table 7.3: Results for the documentary "Hiragasy" CERIMES-SFRS® using closest integer interpolation in the motion compensation.

interpolation is used.

It is difficult to compare the performance of our method with full shot boundary detection methods such as [2, 181] since they use frames for shot changes for all video frames. These methods are not relevant with our objective defined by the rough indexing paradigm. Tables 7.1 – 7.3 show in general a good recall of our method, while the precision remains low. Generally speaking, the shot detection on I-frames is not an objective per se, but an illustration how the partial and rough information (noisy motion fields, their corresponding motion compensation error DC images and DC frames polluted by high frequency noise) can be used for content segmentation when appropriate similarity measures are proposed. The same is true for the complexity of the method. If motion is estimated for the whole video, it can be used as a content descriptor and not only as a mean to compensate the I-frames. The intent of the method is based on the estimation of global camera motion. If global camera motion is considered, we can limit the computation of operational costs only by compensation and similarity measure computation which is of linear complexity $\Theta(N)$ with N as the number of pixels in the DC frames which is 64 times less than in the original images. The

method performs in average 3 times quicker than the video decoding process, that is 3 times quicker than real-time on the same PC.

7.1.5 Conclusion

In this section, we presented how the methods developed for the mosaic construction can be used for video indexing. To this end, we presented a new similarity measure which is based on the regularisation approach, and a related method for shot change detection in the context of rough indexing paradigm. The similarity measure takes into account the degradation of the data: the motion vector fields are noisy and incomplete due to inaccurate MPEG encoder motion estimations and the DC frames are corrupted by aliasing effects. Therefore, a robust global motion estimation method is used and a confidence measure based on the motion compensation error is incorporated in the extrapolation of the motion parameters. In order to deal with the aliasing effects of the DC image, we regularise the MSE weighting it by the local contrast, as we did in the mosaic restoration process, with the aim to favour the comparison on flat areas and to penalise edges and textured regions.

Our method performs fast since only rough data from the MPEG compressed stream is extracted. In the evaluation, we obtain mostly quite high values for recall, but the precision values remain low. The study of the regularisation function and the classification threshold as well can help to ameliorate these results. In addition, an integration of color can be incorporated to obtain a better precision.

We further developed this shot boundary detection for I-frames method and adopted it in a system combined with a detection on P-frames for the use in the TRECVID campaigns 2004, 2005 and 2006 [1], and in the ARGOS campaign 2006 [2]. Figure 7.9 shows our results of the TRECVID 2004 campaign compared to the other participants.

7.2 Camera Motion Characterisation

Digital videos are more and more available and pervasive due to recent progresses in storage, communication, and compression technologies. This consequently implies the increasing need for efficient indexing, browsing, search and retrieval of video archives. Requests for video material in archives often specify desired or required camera motion. Therefore, camera motion identification was a new task in TRECVID 2005. Given the feature test collection and the shot boundary reference, all shots needed to be identified in which a certain camera motion (pan, tilt or zoom) is present. Therefore, we propose classifying the motions in static camera, pan, tilt, zoom and other camera motions. The presented method is also capable to identify complexer camera motions which are here combined in the class of other camera motions. Nevertheless, working within the scope of the TRECVID 2005 camera motion task, we focus here on the identification of pure motions such as pan, tilt and zoom.

[1] `http://www-nlpir.nist.gov/projects/trecvid/`
[2] `http://www.irit.fr/recherches/SAMOVA/MEMBERS/JOLY/argos/`

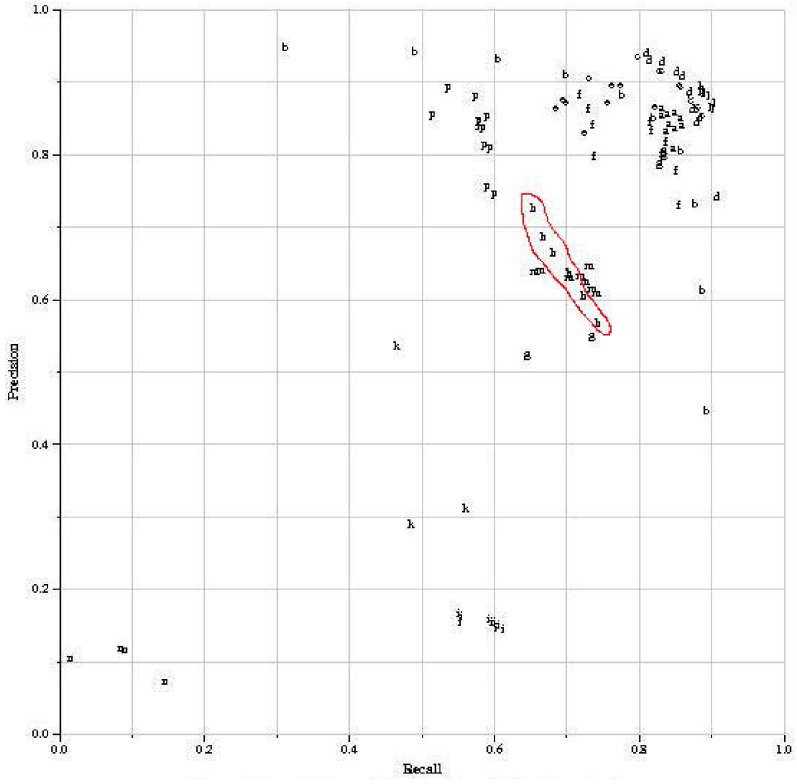


Figure 7.9: Recall and Precision for all transitions (cut and progressive) of the TRECVID 2004 campaign. Our results are circled in red.

The videos in the TRECVID 2005 test collection contained several scenes captured by a hand-carried camera. Jitter motions of the camera resulted. The main challenge in this task was then to overcome these jitter camera motions in order to avoid overdetections. Working on archived or broadcasted content, it is interesting to reuse motion low-level descriptors contained in compressed streams whatever is their quality. Hence, this was an ideal framework for the application of the methods we developed in this thesis.

The fundamental problem for camera motion characterisation consists in estimating the camera model. Current methods for camera motion characterisation in MPEG compressed video generally work on motion compensation vectors or DC images extracted from the compressed stream. The approach of Cao and Suganthan [20] is based on a neural-network scheme to characterise the camera motion in shots. They extract and reconstruct frame-by-frame motion vectors for all frames from the MPEG stream. Sàez et al. [153] estimate global motion parameters based on the Hough transform. Their method works on DC images extracted from the compressed stream. Ewerth et al. [45] compute a 3D camera motion model only processing motion vectors from P-frames. Due to the 3D motion model, the method allows to distinguish between translation along the x-axis (y-axis) i.e. track (boom), and rotation around the y-axis (x-axis) i.e. pan (tilt). Since in the TRECVID 2005 camera motion task any distinction is supposed between track and pan or boom and tilt, these types of motion belong to the same feature groups. Ngo et al. [123] characterise camera and object motions by analysing spatio-temporal image volumes. Then, motion is depicted as oriented patterns in spatio-temporal image slices coming from DC images. They propose a tensor histogram computation in order to represent these patterns. The approach of Doulaverakis et al. [36] proposes the computation of direction histograms of MPEG motion vectors. Depending on the distribution of the histogram and the number of intra-coded macroblocks, camera motion is characterised by applying threshold values on the normalised variance of the histogram. In [86], Kim et al. present a simple thresholding scheme for the motion parameters of the affine six-parameter model. The affine six-parameter model is computed from MPEG motion vectors for each frame, whereas motion vectors for I-frames are interpolated from P-frames.

Bouthemy et al. [16] compute the 2D affine global motion model in order to characterise camera motion. Since thresholding on motion parameters is difficult mainly if jitter motions are present in the scene, thresholding is performed on likelihoods of motion parameters. This method can not be applied to compressed video. In order to evaluate their method on compressed video, they decoded MPEG compressed frames. Due to its robustness, we refer to this method as a basis of our algorithm in the compressed domain.

Figure 7.10 shows the global scheme to characterise the dominant (camera motion) in a video segment. We use first the affine motion model estimated from the compressed stream by the method presented in Section 3.2.2 (see the first block of Figure 7.10). Despite the robust estimation scheme, the resulting motion model parameters are still noisy due to complex motions e.g. due to a hand-carried camera. In addition, they have different meanings so that simple thresholding in order to find the dominant motion is not possible. Therefore, we chose a significance test of the motion model parameters based on [16] which is

the second step in Figure 7.10. This test is formulated as a maximisation of likelihoods s_j associated to two statistical hypotheses. One stands for the significance of the motion parameters expressing the pure physical camera motion. The second stands for the absence of the corresponding motion. Thus, the problem is turned into a better controllable problem of thresholding likelihood-ratios. We suppose that a specific motion is present in the shot if it is present in all P-frames in a video segment of a sufficient duration. Thus, we segment a shot into video segments with homogeneous motion (step three in Figure 7.10). We define a segment of *homogeneous motion* as a series of consecutive frames where the same motion parameters are significant. Therefore, we consider the likelihood motion values as a normally distributed stochastic signal. Based on [16], we apply the Hinkley test on this signal allowing to detect changes on a temporal mean value of the significance values. Due to this segmentation, the duration of a detected motion can be determined. If the duration is too short, then it is considered as a jitter motion and is rejected. Then, we threshold the mean likelihood values $\bar{s}_m$ in a fourth step. Finally, a classification scheme (step five in Figure 7.10) is applied to the thresholded mean likelihood values $\bar{\zeta}_m$ of each segment in order to define the physical character of the motion i.e. "pure" or not. We consider only segments with pure motions (pan, tilt, zoom) as a detection result. The classification using mean values eliminates subliminal jitter motions and provides the dominant motion.

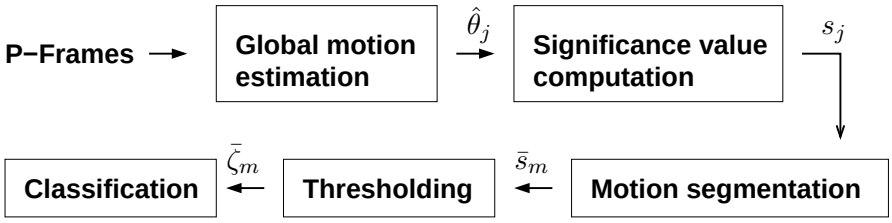


Figure 7.10: The steps of the camera motion characterisation a shot. (The index j is related to frames and m to homogeneous motion segments in a shot.)

In the following, we address each of these steps. We refer the reader to Section 3.2.2 for the estimation of global camera motion. The camera motion characterisation based on the estimated global camera motion is discussed below. Section 7.2.1 presents the significance computation of the motion parameters and the thresholding scheme, Section 7.2.2 the segmentation into segments of homogeneous motion, and Section 7.2.3 the classification scheme. Some results are analysed in Section 7.2.4. Finally, Section 7.2.5 concludes our work and outlines our points of interest for future research.

7.2.1 Significance Value Computation of the Motion Parameters

The estimated global motion model parameters (see Section 3.2.2) may still be noisy due to complex motions e.g. due to a hand-carried camera. This was often the case in the TRECVID 2005 video data. In addition, the parameters in the affine model have different meanings so that simple thresholding in order to find the dominant motion is difficult. Therefore, we present a significance test of the motion model parameters based on [16].

Since we are interested in dominant camera motion, we express the vector of motion parameters $\theta = (a_1, ..., a_6)^T$ in another basis of elementary motion-subfields as in [16]:

$$\begin{array}{l} \phi = (pan, tilt, zoom, rot, hyp1, hyp2) \quad \text{with} \\ \begin{array}{ll} pan = a_1 & tilt = a_4 \\ zoom = \frac{1}{2}(a_2 + a_6) & rot = \frac{1}{2}(a_5 - a_3) \\ hyp1 = \frac{1}{2}(a_2 - a_6) & hyp2 = \frac{1}{2}(a_3 + a_5) \end{array} \end{array} . \tag{7.2}$$

This basis is more convenient for the interpretation of the dominant motion in the scene since its is more related to the physical meaning. In [85], the affine model is transformed into a similar basis, but they consider only one hyperbolic term defined as a combination of $hyp1$ and $hyp2$. Thus, the model (3.23) becomes:

$$\mathbf{d}(\mathbf{p}, \phi) = \left\{ \begin{array}{l} d_x = pan + zoom \cdot (x - x_0) - rot \cdot (y - y_0) + hyp1 \cdot (x - x_0) + hyp2 \cdot (y - y_0) \\ d_y = tilt + zoom \cdot (y - y_0) + rot \cdot (x - x_0) - hyp1 \cdot (y - y_0) + hyp2 \cdot (x - x_0) \end{array} \right. \tag{7.3}$$

where $(x_0, y_0)^T$ denotes the image center.

If the dominant motion is for example a pure panning, the parameter pan is supposed to be the only non zero. This is the same for $tilt$, $zoom$ and rot. If the camera is static all parameters are supposed to be zero. In practice, this is never the case due to noise, estimation errors or moving objects. In addition, the physical meaning of the parameters is different as we stated in in Section 3.2.2. Thus, the typical range is not the same. Hence, it is difficult to propose an appropriate thresholding scheme to decide if a motion feature is present or not.

The significance test from [16] is a statistical approach based on a likelihood ratio test which turns the problem of direct thresholding into a better controllable problem of thresholding likelihood ratios.

Let us consider two competing hypotheses for each component of ϕ. The first hypothesis H_0 assumes that the considered component of ϕ is significant. The second one H_1 assumes that the component is not significant i.e. it equals zero, while the other five parameters are free. Let $\hat{\phi}_0$ and $\hat{\phi}_1$ be respectively the motion models corresponding to the hypotheses H_0 and H_1. The advantage of such a test is that it is independent from the values of the other parameters which remain free.

The likelihood function f for each hypothesis of the considered component is defined with respect to the residuals $\mathbf{r}_i = (r_{x,i}, r_{y,i})^T$ of Equation (3.24). They are supposed to be independent, and to follow a zero-mean Gaussian law. The covariance matrices $\Sigma_l, l = 0, 1$ corresponding to the two hypotheses are a-posteriori estimated as:

$$\Sigma_l = \begin{pmatrix} \sigma_{x,l}^2 & 0 \\ 0 & \sigma_{y,l}^2 \end{pmatrix} \tag{7.4}$$

with

$$\sigma_{m,l}^2 = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} r_{m,i}(\hat{\phi}_l)^2, \; m = x, y \tag{7.5}$$

where $\mathcal{D}$ is the set of motion inliers on which the residuals are computed and $|\mathcal{D}|$ denotes its cardinality.

The two likelihood functions f for the optimised values of the motion parameters $\hat{\phi}_l$ are given by:

$$\begin{aligned} f(\hat{\phi}_l) &= \prod_{i\in\mathcal{D}} \left(\frac{1}{2\pi\sqrt{\det(\Sigma_l)}} \exp\left(-\frac{1}{2}(\mathbf{r}_i^T \Sigma_l^{-1} \mathbf{r}_i) \right) \right) \\ &= \frac{1}{(2\pi\sigma_{x,l}\sigma_{y,l})^{|\mathcal{D}|}} \exp\left(-\frac{1}{2} \sum_{i\in\mathcal{D}} (\mathbf{r}_i^T \Sigma_l^{-1} \mathbf{r}_i) \right) \\ &= \frac{1}{(2\pi\sigma_{x,l}\sigma_{y,l})^{|\mathcal{D}|}} \exp\left(-|\mathcal{D}| \right) \end{aligned} \tag{7.6}$$

In order to estimate the five free parameters for hypothesis H_1, we can profit from the previous estimation of $\hat{\theta}$ i.e. $\hat{\phi}_0$ which already furnishes $\mathcal{D}$. Then, a least square estimation similarly to (3.31) can be used only on macroblock vectors from D. The observation matrix $\mathbf{H}_\phi$ corresponding to the parameter vector ϕ is:

$$\mathbf{H}_\phi = \begin{pmatrix} 1 & 0 & x_1 & -y_1 & x_1 & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & x_N & -y_N & x_N & y_N \\ 0 & 1 & y_1 & x_1 & -y_1 & x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & y_N & x_N & -y_N & x_N \end{pmatrix} . \tag{7.7}$$

for N observed motion vectors.

In order to set the ith component to zero in the parameter vector ϕ, we suppress in practice the ith column in the observation matrix $\mathbf{H}_\phi$. Thus, the least square equation is:

$$\tilde{\phi} = (\mathbf{H}_0{}^{\mathrm{T}} \mathbf{W} \mathbf{H}_0)^{-1} \mathbf{H}_0{}^{\mathrm{T}} \mathbf{W} \mathbf{Z} \tag{7.8}$$

where $\mathbf{H}_0$ is the reduced observation matrix and $\tilde{\phi}$ is the reduced parameter vector which contains the free five parameters. The matrix $\mathbf{W}$ and the vector $\mathbf{Z}$ remain the same as in (3.35) and (3.33).

The ratio s is called the significance value:

$$\begin{aligned} s &= \ln\left(\frac{f(\hat{\phi}_1)}{f(\hat{\phi}_0)} \right) \\ &= \ln\left(\frac{\frac{1}{(2\pi\sigma_{x,1}\sigma_{y,1})^{|\mathcal{D}|}} \exp\left(-|\mathcal{D}|\right)}{\frac{1}{(2\pi\sigma_{x,0}\sigma_{y,0})^{|\mathcal{D}|}} \exp\left(-|\mathcal{D}|\right)} \right) \\ &= |\mathcal{D}| \left(\ln(\sigma_{x,0}\sigma_{y,0}) - \ln(\sigma_{x,1}\sigma_{y,1}) \right) \end{aligned} \tag{7.9}$$

If we assume that $\sigma_x = \sigma_y$ (this assumption is rather strong, but still if the motion is homogeneous it is hold), then s becomes:

$$s = |\mathcal{D}| \left(\ln(\sigma_0^2) - \ln(\sigma_1^2) \right) \tag{7.10}$$

where σ_l^2 is computed on the amplitude of the residuals $\mathbf{r}_i$. This, simplification conducts also to a gain in computational time.

We now aim to use this value for testing the significance of the parameters. Therefore, our idea is if a motion feature is present in a shot, its corresponding motion parameter is significant during a sufficient number of frames. We can not directly use the significance values for the following reasons. First of all, they can be noisy due to jitter motions. Therefore, we try to smooth them along the time and take the decision on the temporal mean of the significance values. Based on this mean, we will segment a shot into subshots of homogeneous motion (presented below in Section 7.2.2). In order to get a temporal regularity, we interpolate the significance values for I-frames. Assuming linear and constant motion, for each I-frame the significance values of the preceding P-frame are repeated. This is a simplified approach and it will be a point of future research.

The second source of noise in the significance values are the failures of the MPEG encoder as it furnishes inaccurate motion vectors. These vectors are considered as outliers by the robust motion estimator (see Section 3.2.2) and camera motion will be estimated only on a small part of a frame. Thus, we decide to exclude the frames where the estimation support is too small. To do this we introduce the confidence measure $c_{\mathcal{D}}$:

$$c_D = \frac{|\mathcal{D}|}{\mathcal{D}_{max}} \tag{7.11}$$

where $\mathcal{D}_{\max}$ is the maximum number of motion inliers i.e. the number of macroblocks in a frame. If c_D is lower than a threshold $\lambda_{\mathcal{D}}$, then the motion estimated on a given P-frame will not be further considered.

It might also be that the global motion estimation algorithm fails. In order to control the accuracy of the motion model with respect to the MPEG motion vectors, the variance of the residuals (3.24) computed on $\mathcal{D}$ are used as a second confidence measure c_σ. If c_σ exceeds a predefined threshold λ_σ, the motion estimated on this frame is not further considered either.

Then, to decide which hypothesis has to be selected, the mean significance value $\bar{s}$ is computed on a homogeneous motion segment $\mathcal{M}$ (excluding the frames rejected by the confidence measures $c_{\mathcal{D}}$ and c_σ) and the following mean log-likelihood test is performed:

$$\bar{s} = \frac{1}{|\mathcal{M}|} \sum_{s \in \mathcal{M}} s \begin{array}{c} H_0 \\ < \\ > \\ H_1 \end{array} \lambda_s \tag{7.12}$$

where $|\mathcal{M}|$ is the number of frames in $\mathcal{M}$.

If the mean significance value $\bar{s}$, which is in general negative, is lower than a predefined threshold λ_s, then the component at hand is declared to be significant, otherwise it is considered to be null.

It is obvious that not only one component exceeds the threshold λ_s because motion in the scene is mostly a combination of basic motions. It is still possible that one dominant motion exists and though the omission of its parameter causes a much more higher increase of the error than in the case of the other remaining significant parameters. Therefore, we

retain only the components which exceed λ_s and $C_s \cdot \min\{\bar{s}_{pan}, \bar{s}_{tilt}, \bar{s}_{zoom}, \bar{s}_{rot}, \bar{s}_{hyp1}, \bar{s}_{hyp2}\}$, $C_s \in [0, 1]$.

7.2.2 Segmentation into Segments of Homogeneous Motion

Here, we describe the method for segmenting shots into sequences of homogeneous motion. We assume that the likelihood motion values s form a stochastic signal that is normally distributed. Based on [16], we apply the Hinkley test to the signal allowing to detect changes on a temporal mean value. These changes delimit the borders of homogeneous motion segments.

Two tests are performed in parallel to look for downwards or upwards jumps. They are respectively defined by:

$$U_k = \sum_{t=0}^{k} \left(s_t - \tilde{s} + \frac{\delta_{min}}{2} \right) \ (k \geq 0) \tag{7.13}$$

$$M_k = \max_{0 \leq i \leq k} U_i; \text{ detection if } M_k - U_k > \lambda_H \tag{7.14}$$

$$V_k = \sum_{t=0}^{k} \left(s_t - \tilde{s} - \frac{\delta_{min}}{2} \right) \ (k \geq 0) \tag{7.15}$$

$$N_k = \min_{0 \leq i \leq k} V_i; \text{ detection if } V_k - N_k > \lambda_H \tag{7.16}$$

where $\tilde{s}$ is the online mean significance value before the jump, δ_{min} is the minimal jump magnitude that we want to detect, and λ_H is a predefined threshold. We perform this test simultaneously on all mean significance values $(\bar{s}_{pan}, \bar{s}_{tilt}, \bar{s}_{zoom}, \bar{s}_{rot}, \bar{s}_{hyp1}, \bar{s}_{hyp2})$. If a jump has been detected on one of the signals, the means $\bar{s}$ are re-initialised for each signal.

This segmentation allows to know the duration of a certain camera motion. If the duration is too short, then the segment is considered to represent jitter motion and is rejected.

7.2.3 Camera Motion Classification

When the segments of homogeneous motion are known, finally a classification scheme can be applied to the thresholded mean significance values $\bar{\zeta}$ of each segment in order to define the physical character of the motion. We consider only segments with pure motions (pan, tilt, zoom) as a detection result. The classification using mean values eliminates subliminal jitter motions and provides the dominant motion.

Table 7.4 shows the classification scheme we used for TRECVID 2005 to determine the physical meaning of the set of the thresholded mean significance values $\bar{\zeta} = (\bar{\zeta}_{pan}, \bar{\zeta}_{tilt}, \bar{\zeta}_{zoom}, \bar{\zeta}_{rot}, \bar{\zeta}_{hyp1}, \bar{\zeta}_{hyp2})$. In this classification, we followed [16]. If a motion segment of a shot with a sufficient long duration is classified in one of the classes 2, 3 or 4, then the shot is identified to contain the corresponding motion. Since a zoom is often combined

	$\bar{\zeta}$	camera motion
1	$(0,0,0,0,0,0)$	static camera/ no significant motion
2	$(\bar{\zeta}_{pan},0,0,0,0,0)$	pan
3	$(0,\bar{\zeta}_{tilt},0,0,0,0)$	tilt
4	$(\bar{\zeta}_{pan},\bar{\zeta}_{tilt},\bar{\zeta}_{zoom},0,0,0)$	zoom
5	others	complex camera motion

Table 7.4: Classification scheme

with a small pan or tilt, it is possible that the pan or tilt parameters are significant as well. This is also due to inaccurate MPEG motion vectors.

Finally, if successive segments are labelled with the same motion, the segments are joined. Two segments labelled with the same motion are joined as well if they are separated by a rejected segment.

7.2.4 Results

For TRECVID 2005, we had several parameters to manipulate. Since no annotation was available for the development set, we annotated only a few videos for the parameter training. In addition, if manually annotating it is difficult to decide if the feature is clearly true or not and if the annotator of the ground truth of the test set will decide in the same manner.

The parameters we use for camera motion characterisation are:

- δ_{min} denotes the minimum jump magnitude that we want to detect in the Hinkley test.
- λ_H is the peak validation threshold for the Hinkley test.
- λ_s is the absolute threshold for the significance values.
- C_s is the constant for the relative thresholding of the significance values.
- t_{min} denotes the minimal motion duration i.e. the minimal number of frames in a valid homogeneous motion segment.
- $\lambda_{\mathcal{D}}$ is the absolute threshold for the confidence measure c_D i.e. the minimum size of the estimation support D.
- λ_σ is the absolute threshold for the confidence measure c_σ i.e. the accepted maximum variance of the residuals $\mathbf{r}_i(\hat{\phi}_0)$.

Then seven runs (RI-1,..., RI-7) with respectively different parameter settings were submitted and evaluated by the NIST. The most balanced result (RI-3: 0.912 mean precision and 0.737 mean recall) and the best result for recall as well was obtained for the following parameter settings: $\delta_{min} = 100$, $C_s = 0.3$, $\lambda_s = -30$, $\lambda_H = 0.1$, $t_{min} = 15$, $\lambda_\sigma = 1000000$, and $\lambda_{\mathcal{D}} = 0.01$. The threshold λ_σ is chosen quite high and $\lambda_{\mathcal{D}}$ quite low in order to not reject too much frames. The best result for precision (RI-2: 0.967 mean precision and 0.541 mean

recall) are obtained for $\delta_{min} = 100$, $C_s = 0.25$, $\lambda_s = -70$, $\lambda_H = 0.1$, $t_{min} = 25$, $\lambda_\sigma = 1000000$, and $\lambda_{\mathcal{D}} = 0.01$.

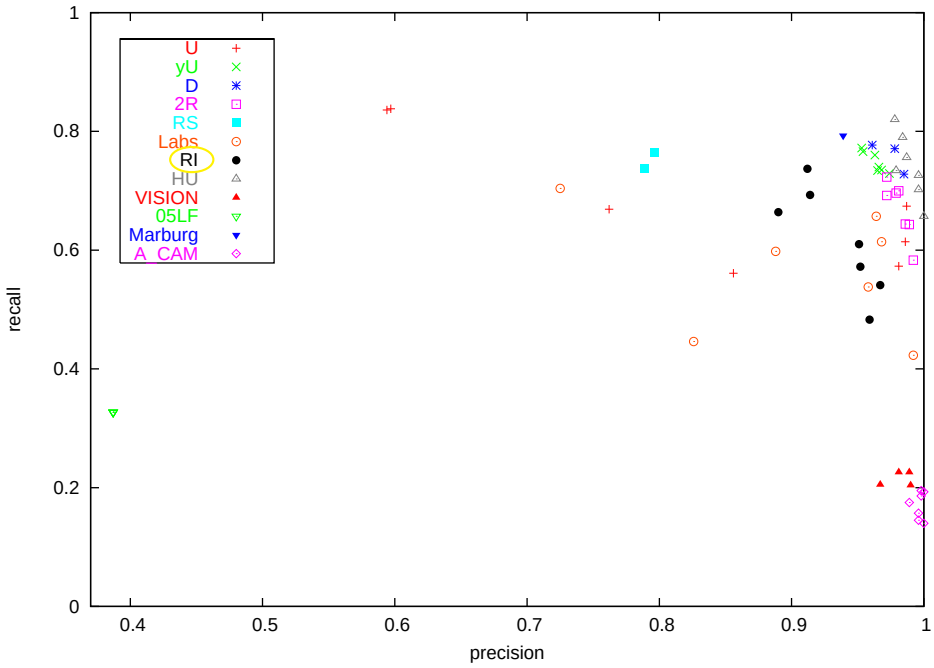
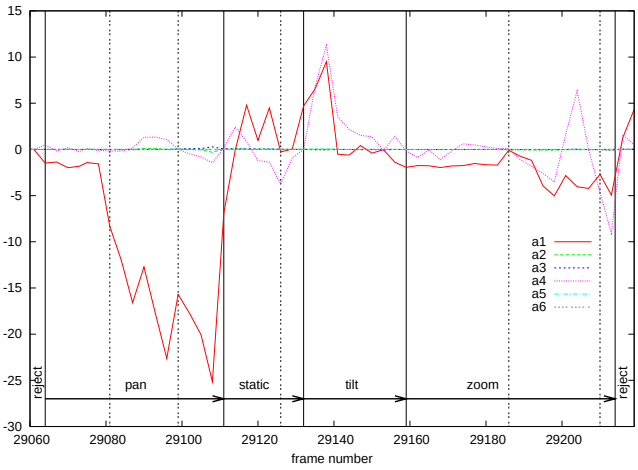


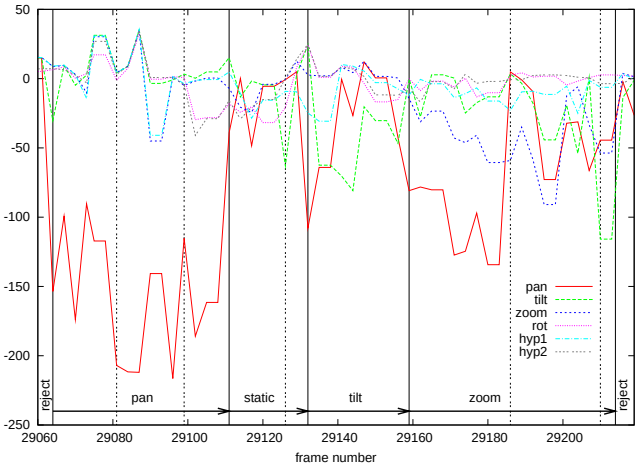
Figure 7.11: Precision and recall results for the submissions of all participants in the TRECVID 2005 camera motion detection task (RI corresponds to LaBRI).

Figure 7.11 shows the precision and recall results for the submissions of all participants in this TRECVID task. The submission results of the group LaBRI are the black points denoted as "RI" in the key.

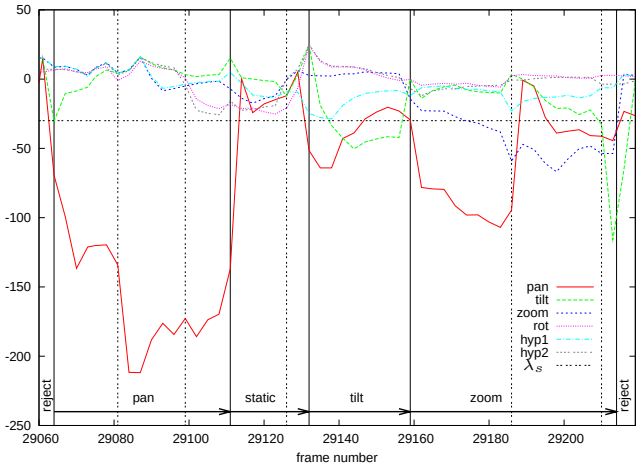
Figure 7.12 shows some results obtained in the run RI-3. It visualises the graphs of the motion model parameters $\hat{\theta}$, the corresponding significance values s and the online mean significance values $\bar{s}$ of the shot labelled as "shot106_136". This shot is captured by a hand-carried camera and so contains a lot of jitter motions. The black lines (dotted and solid) in the graphs indicate the borders of the homogeneous motion segments. The motion segments we obtain after the joining of neighboured similar motions and the rejection of too short motion segments are marked with solid lines. Note that in this shot only the motion segments at the beginning and the end of the shot have been rejected as too short i.e. jitter motion. The camera features we detect in this shot are pan, static camera/ no significant motion, tilt and zoom. The real camera motion is a pan left followed by a zoom in. Both are correctly detected and are visualised in Figure 7.13. A lot of jitter motion is present between these two camera motions. One part is correctly labelled as static camera or non significant motion. The other part is falsely detected as a tilt. The graphs in Figure 7.12(a) show the motion parameters which are very noisy. Here no zoom is visible, since the parameters indicating zoom (a_2 and a_6) have a different meaning than the parameters a_1 and a_4 respectively responsible for pan and tilt. If the significance values of the motion parameters are computed, the motions



(a)



(b)



(c)

Figure 7.12: 7.12(a), 7.12(b), and 7.12(c) show respectively the graphs of the estimated affine global motion parameters $\hat{\theta}$, the corresponding significance values s and the online mean significance values $\bar{s}$ for the shot labelled as "shot106_136".

Figure 7.13: Shot "shot106_136" of the TRECVID 2006 corpus: (a) and (b) show respectively the first, an intermediary and the last image for the pan and zoom correctly detected in Figure 7.12.

and mainly the zoom become more clear. However the graphs of the significances values in Figure 7.12(b) are still quite noisy. This improves after the mean value computation which is shown in Figure 7.12(c).

7.2.5 Conclusion

In this section we presented another application of our mosaicing method: the characterisation of dominant camera motion and its benchmarking in the TRECVID 2005 campaign. It is shown that the use of MPEG motion vectors was justified, despite their noisiness, in a stochastic estimation and decision framework. The main difficulty resulted from the lack of ground truth for the training of parameters of the proposed models. The next step would be high-level indexing tasks and the use of this motion characterisation tool in this context.

We further used this motion characterisation method with some modifications in the ARGOS campaign 2006 and within the BBC rushes exploitations tasks for TRECVID 2006.

7.3 Conclusion

In this application chapter we used several components of the proposed mosaicing method for related indexing tasks: the shot boundary detection on I-frames and the characterisation of dominant camera motion. In the first case, we used the robust motion estimation from the compressed stream and the extrapolation of motion for I-frames, in order to warp one I-frame to another and measure their similarity. Then, we defined a similarity measure based

on the regularisation operator of the restoration approach. The latter allowed the efficient comparison in case of textured frames. In the second case, we completed the robust motion estimation by a temporal filtering and a probabilistic decision-making framework. These applications show the interest of working on compressed data, when taking into account their "rough" nature.

Chapter 8

Error Concealment

In this chapter, we design a possible application of our restoration method for error concealment, when transmitting compressed video. We stress here that we only show the perspective of applying our mosaicing method for error concealment, we can not be exhaustive in the subject. We propose applying our mosaicing method at the decoder side and conceal a frame by replacing corrupted regions in the frame with the corresponding regions of the mosaic. The purpose here is not to present a very subtle method. Some powerful method for temporal error concealment have been already presented [29]. Our objective is to show that the presented mosaicing method can be applied within this context. Therefore, we compare our results with a spatial error concealment method. We choose the frequency selective method of Meisinger and Kaup [111, 83] as it is a reference method in this domain. This work was developed in the research project [102].

Whenever digital video data is transmitted over an error-prone network, parts of the data might be altered or lost during transmission due to channel noise, congestion or other network errors. As a result, the video images suffer from visual errors, whose characteristics depend on the video coding method employed. To deal with these errors, three different approaches can be taken.

The first is to avoid the underlying network errors by using network protocols that guarantee a correct transmission, like the transmission control protocol (TCP). However, their reliability is achieved at the expense of speed deficits. The second approach aims at making the video coding more resilient to errors. Different kinds of redundancies are added to the video stream in order to support the correction of the video data in case of transmission errors or to minimise their effect. A review of these methods can be found in [201], where they are referred to as *forward error concealment* techniques. The third approach is to employ image restoration algorithms in order to approximate the lost data based on the successfully received data. This is possible by either exploiting the temporal redundancy inherent in a video, or by estimating lost parts of an image based on their surroundings. These approaches

are referred to as *temporal* and *spatial error concealment*, respectively, and are the main interest in this work.

In order to study the application of our video restoration method in an error concealment framework we first briefly survey losses in MPEG-2 coded video. Then, we chose a model of transmission errors and simulate them. Here, without being specialists on networking we restrict ourselves to classical models. Having simulated the transmission errors, we can apply our method to recover (partially) such errors.

This chapter is organised as follows. First, the characteristic effects of lost data on MPEG-2 coded video are examined, and methods to simulate such errors are evaluated in order to provide a realistic basis for later experiments. Section 8.1 deals with this subject. Then, we present our error concealment method in Section 8.2. In this section, we also discuss spatial and temporal error concealment algorithms, placing emphasis on the frequency selective extrapolation method of Kaup and Meisinger [111, 83]. Furthermore, experiments examining the quality of our error concealment techniques are conducted and evaluated in Section 8.3. In addition, we compare our results with the frequency selective extrapolation method [111, 83]. Section 8.4 closes this chapter by summing up the achieved results and giving future perspectives.

8.1 Transmission Errors in MPEG-2 Video

The object in this section is the simulation of transmission error for MPEG-2 compressed video. Therefore, we first precise the type data if transmitting MPEG-2 compressed video. Then, we present some classical loss models for transmission errors and our choice of loss models. Based on this, we then describe how we simulate transmission errors for our experiments.

8.1.1 Multiplexing

The term multiplexing denotes techniques to merge several different streams, like an audio and a video stream, into one stream. In Part 1 of the MPEG-2 standard [116], two distinct multiplexing methods are defined. The first one produces so called *program streams*, intended to be used in error-free environments like storage. The second method produces *transport streams* which are evidently designed for applications that involve some kind of transmission. Since this work treats transmission errors, we assume that we are always dealing with transport streams, and program streams can be neglected.

A transport stream is composed of transport stream packets of fixed size. In addition to 4 header bytes, each of these packets contains 184 bytes of payload, that is used to transport fragments of so called *packetised elementary stream* (PES) packets [172]. These packets contain reasonable amounts of audio or video data produced by the appropriate coder, in case of video data normally an integral number of images. However, no general statement can be made about which spatial extent is represented by a transport stream packet.

8.1.2 Loss Models for Transmission Channels

The next question to discuss is how the occurrence of transmission errors can be modelled. This requires a definition of the term transmission error first. A *transmission error* is an event that results in the loss of a minimal, "atomic" amount of data. This could be the loss of a single bit or a whole packet (e.g. MPEG-2 transport stream or TCP packet), depending on the transmission channel and protocol employed. In the following, the terms error and loss will be used synonymously.

To characterise loss patterns for error-prone transmission channels, several models have been proposed. Here, we consider classical models such as the Bernoulli model, the 2-state Markov chain model (also known as the Gilbert model), the kth order Markov chain model and the extended Gilbert model. In case of $k = 0$ the kth order Markov chain model corresponds to the Bernoulli model and in case of $k = 1$ it corresponds to the Gilbert model.

Loss models describe the occurrence of errors as a discrete stochastic process $\{X_t\}_{t=1}^{\infty}$ where X_t is a random variable representing the state of the channel at time t (see [206]). Possible values are 0 or 1 representing an error-free and an erroneous state respectively. A finite set of concrete values of X_t for $t = 1, \ldots, n$ is known as a realisation of $\{X_t\}$ and denoted by $\{x_t\}_{t=1}^{n}$.

Practically, all loss models presented here specify error probabilities for X_t, and they differ in the number of previous states of the channel $(x_{t-1}, x_{t-2}, \ldots)$ taken into account for these probabilities. The order of the Markov process, k, is the number of previous values of the process on which the current model depends and is a measure of complexity of the model and has 2^k states. Hence, the Bernoulli model which is of order 0 has no states, and the Gilbert model which is of order 1 has 2 states.

Bernoulli Model

The simplest loss model is the Bernoulli model [206, 80]. It assumes that $\{X_t\}$ is a Bernoulli process, meaning that error events are independent of each other. Mathematically spoken, the probability of X_t to be 1 does not depend on any previous state and is always the same. Therefore, one single parameter r, equivalent to this probability, specifies the Bernoulli model [206]:

$$r = P(X_t = 1) = \frac{n_1}{n} \tag{8.1}$$

where n_1 is the number of times the value 1 occurs in the observed times series $\{x_t\}_{t=1}^{n}$, and n is the number of samples in the time series.

Gilbert Model

A more elaborate model is the so called Gilbert model [206, 80, 154], which takes into account the previous state of the channel. As the error probability at time t depends on whether there was an error at time $t-1$ or not, two parameters are needed for the Gilbert model. These two

parameters p and q specify the probability of a transition from an error-free to an erroneous state and vice versa. The likelihoods of p and q are given by [206]:

$$p = P(X_t = 1|X_{t-1} = 0) = \frac{n_{01}}{n_0} \tag{8.2}$$

$$q = P(X_t = 0|X_{t-1} = 1) = \frac{n_{10}}{n_1} \tag{8.3}$$

where n_{01} is the number of times in the observed time series that 1 follows 0, n_{10} is the number that 0 follows 1, n_0 is the number of 0s in the trace, and n_1 is the number of 1s in the trace. The Gilbert model is illustrated in Figure 8.1.

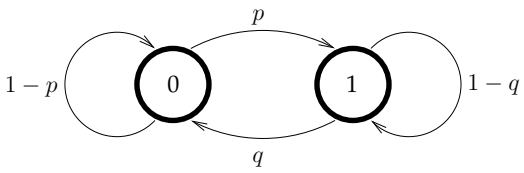


Figure 8.1: The Gilbert model [80].

Markov Chain Model

The Bernoulli and Gilbert model are special cases of a more general model, the Markov chain model. It is mentioned as an error model in [199, 206]. A Markov chain model of kth order specifies the error probability at time t depending on the values of the k preceding states $x_{t-1}, \ldots, x_{t-k}$.

Such a process is characterised by a $k \times 2$ conditional probability matrix P_k whose rows may be interpreted as probability mass function according to which, the next random variable X_t is generated when the process is in a state $x_{t-k} \ldots x_{t-1}$ [206]:

$$P(X_t = x_t|X_{t-1} = x_{t-1}, \ldots, X_{t-k} = x_{t-k}) = P_k(x_t|x_{t-1}, \ldots, x_{t-k}) \tag{8.4}$$

A process $\{X_t\}$ is a Markov chain of order k, if the conditional probability:

$$P(X_t = x_t|X_{t-i} = x_{t-i}) \tag{8.5}$$

is independent of x_{t-i} for all $i > k$.

Let $\{x_t\}_{t=1}^n$ be an observed sequence from a Markov source, and $\underline{b} = b_1 \ldots b_k$ be a given state of the chain, $n_{\underline{b}a}$ the number of times state $\underline{b}$ is followed by a value a in the sample sequence, and $n_{\underline{b}}$ the number of times state $\underline{b}$ is seen. Let $p_{\underline{b}a}$ be an estimate of the probability that $x_t = a$, given that $x_{t-k} \ldots x_{t-1} = \underline{b}$. Then $p_{\underline{b}a}$ estimates the state transition probability from state $\underline{b}$ to state $b_2 \ldots b_{k-1}a$. It is given by [206]:

$$p_{\underline{b}a} = \begin{cases} \frac{n_{\underline{b}a}}{n_{\underline{b}}} & \text{if } n_{\underline{b}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8.6}$$

The Markov model for $k = 3$ is illustrated in Figure 8.2.

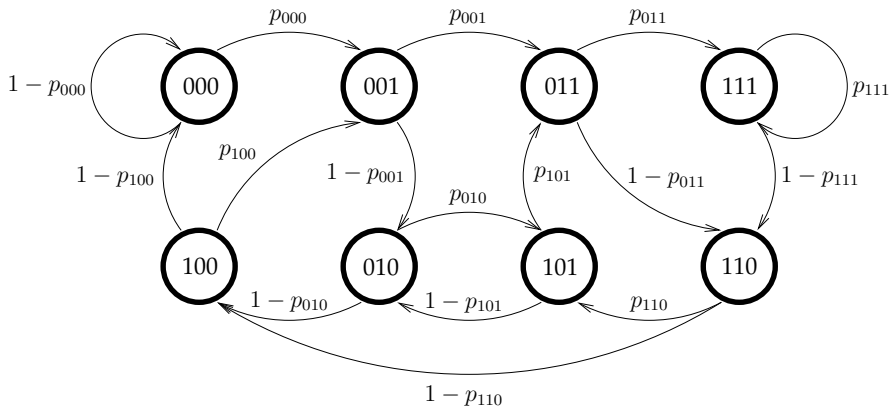


Figure 8.2: The 3rd order Markov chain model.

Extended Gilbert Model

The Markov chain model is able to model complex error patterns, but the amount of parameters necessary is a major drawback. A less complex model is desired for practical use such as the extended Gilbert model [80, 154]. It is a generalisation of the 2-state Gilbert model and is based on the assumption that the error probability for X_t only depends on preceding erroneous states, not on error-free ones.

As a result, only k parameters are necessary to model dependencies on up to $k-1$ preceding erroneous states [154, 80]:

$$p_{01} = P(X_t = 1 | X_{t-1} = 0) = \frac{\sum_{i=1}^{k-1} o_i}{n_0}$$
$$p_{(n-1)n} = P(X_t = 1 | X_{t-1} = 1, \ldots, X_{t-n} = 1) = \frac{\sum_{i=n}^{k-1} o_i}{\sum_{i=n-1}^{k-1} o_i} \tag{8.7}$$

where $1 < n < k$, n_0 is the number of times the value 0 occurs in the trace, and o_i is the occurrence of a loss of the length i (sequence of 1s). Thus, from (8.7) it is clear that the Gilbert model is a special case of the extended Gilbert model when $k = 2$. The extended Gilbert model is depicted in Figure 8.3.

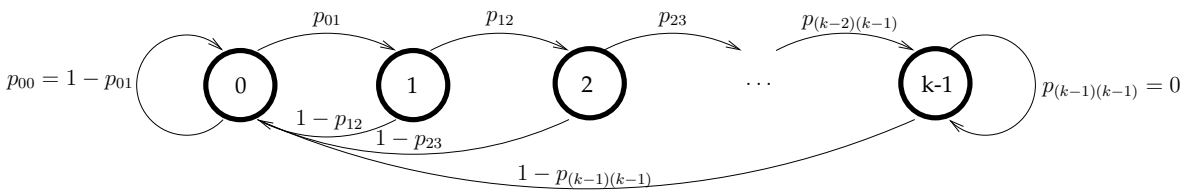


Figure 8.3: The extended Gilbert model [80].

8.1.3 Selection of a Loss Model

The loss models discussed above differ largely in complexity and performance. The Bernoulli model is known to not approximate well losses [206, 80]. Yajnik et al. [206] examined the accuracy of the Bernoulli model, the Gilbert model and the Markov chain model. They found that an Markov chain model of order 2 or greater is necessary to accurately model losses. Jiang and Schulzrinne [80] evaluate the Bernoulli model, the Gilbert model, and the extended Gilbert model. The Markov chain model is not included in the comparison, since the authors consider it as too complex due to the numerous parameters. The best results are obtained for the extended Gilbert model.

Thus, we have to decide which loss model we retain to simulate transmission errors. On one hand, it should realistically model the occurrence of errors, but on the other hand, it should not be too complex in terms of its parameters. Therefore, the inaccurate Bernoulli model and the complex Markov Chain model are out of the question.

Moreover, the Gilbert model is easier to configure than the extended Gilbert model, as it only needs two parameters. Thus, we retain the Gilbert model in order to simulate transmission errors. Since parameters p and q are dependent on the network characteristics, we chose them as $p = 0.001$ and $q = 0.1$. These values can be seen as a rough approximation of the estimated parameters in [206].

8.1.4 Simulation of Transmission Errors

In the context of actual work we consider the MPEG-2 architecture and the worst case of error at macroblock basis. We assume that all information for a macroblock has been lost: the motion vector and the DC coefficients as well. The MPEG-2 architecture allows the limiting of error propagation due to the partitioning of the macroblocks into slices. Then, if a lost occurs the following macroblocks up to the end of the slice are lost as well [195]. As the coding of each individual slice is independent from the others, it will not be propagated across the slice border. This will be taken into account in our error simulation.

In case of I-frames, Huffman-type coding using codes of variable length is applied to the quantised DCT coefficients. Hence a single bit error can cause the decoder to loose synchronisation, so that it cannot determine correctly where individual code words begin and end. That means, that after the occurrence of an error, even correctly received data cannot be decoded until synchronisation is reestablished. In order to allow a resynchronisation of the decoder, a certain synchronisation code is included in every slice header. As a consequence, an error in a video stream will result in a loss of the rest of the affected slice. P-frames and B-frames suffer twice from transmission errors: (i) errors that directly occur due loss and (ii) error propagation due to motion compensation. Figure 8.5 illustrates the error propagation from an erroneous I-frame to the following P-frame. A part of the macroblock pointed by the motion vector is erroneous. Thus, the compensated region in the P-frame is erroneous as well.

Nevertheless, in this work we consider only transmission errors in I-frames. Thus, error simulation of each frame is independent from the others. The extension to P and B-frames is

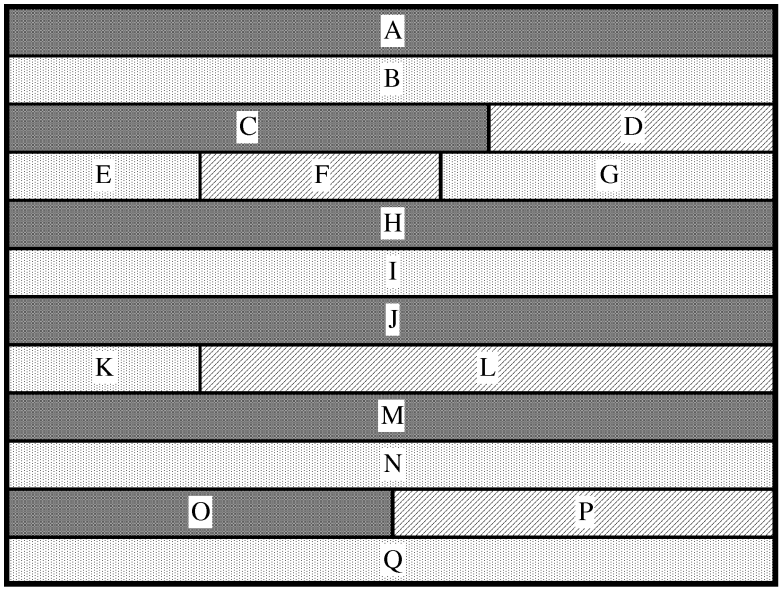


Figure 8.4: Slice structure of a frame [114].

in the perspective of future work.

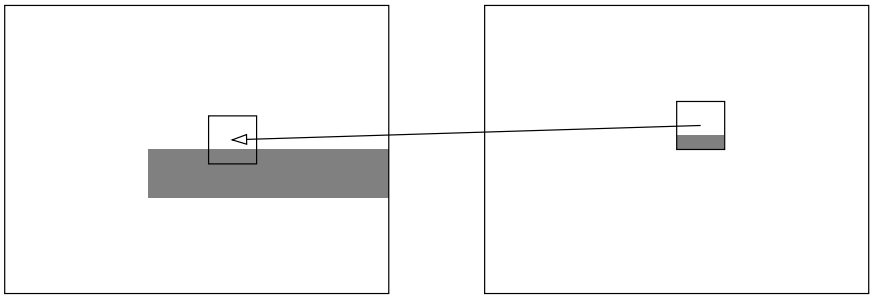


Figure 8.5: Error propagation from an erroneous I-frame to the following P-frame.

Figure 8.6 shows the I-frames of the sequence "Tympanon" corrupted by simulated loss using the Gilbert model. If a macroblock is lost, all the macroblocks until the end of the slice are declared to be lost too.

8.2 Error Concealment by Image Restoration

Having simulated transmission errors, we focus now on their concealment. Hence, we present in this section our method for error concealment in I-frames based on the super-resolution mosaicing technique presented in Chapter 4. To this end, we review image restoration methods for error concealment and describe then our approach to this problem.

Figure 8.6: The simulated loss by the Gilbert model for the sequence "Tympanon".

8.2.1 State of the Art

An common approach to error concealment is to employ image restoration algorithms in order to approximate the lost data based on the successfully received data. This is possible by either exploiting the temporal redundancy inherent in a video, or by estimating lost parts of an image based on their surroundings. These approaches are referred to as *temporal* and *spatial error concealment*, respectively. In the following, we give first a brief overview of spatial error concealment techniques, focusing on the method proposed by Meisinger and Kaup [111, 83] as we will compare our results with this method. Second, we outline temporal error concealment methods.

Spatial Error Concealment

Spatial error concealment methods aim at reconstructing lost areas of an image based on the available surrounding data. To obtain an estimation of the lost areas, different approaches have been taken, including various interpolation techniques, methods that analyse the image with respect to edges and try to reconstruct them, and solutions in the frequency domain. Since spatial error concealment does not rely on temporal redundancy, i. e. preceding or succeeding images from a sequence, it can be applied not only to video, but also to still images. Here, only a short overview of common spatial error concealment methods is given. A more extensive review can be found in [201].

Wang and Zhu [201] based their approach on the observation that images predominantly contain low frequencies and the consequent assumption that pixels in a local neighbourhood have similar values. They proposed two spatial smoothness constraints between the pixels of a lost block and a one-pixel border, which assure maximally smooth transitions across the block boundaries or the whole lost block. Obviously, this algorithm is suited for monotonous areas but falls short of reconstructing high frequencies like edges.

The technique of projection onto convex sets (POCS) has been applied to spatial error concealment by Sun and Kwok [183]. To overcome the problem of edge reconstruction, they use a Sobel filter to detect edges in the eight neighbouring blocks of a lost block. Then, a projection operator is chosen according to the results, while a second projection operator realises a range constraint. By alternately projecting an initial guess using these two projection operators, a reconstruction of a lost block can be obtained in five to ten iterations.

Unlike the previously mentioned methods, which recover a whole block at once, or *parallel*, a pixel-wise *sequential* recovery is proposed by Li and Orchard [100]. Once a pixel is reconstructed, it is used for further error concealment which is performed by an orientation adaptive interpolation. This technique allows the recovering of important image features in both high and low frequency areas. The results are convincing, but the computational complexity of this algorithm is high, since it has to be applied to every lost pixel separately.

Frequency selective extrapolation is a spatial error concealment technique that has been proposed by Meisinger and Kaup [111, 83]. It works on a rectangular image area $\mathcal{L}$ that is formed by the union of a set of known pixels $\mathcal{A}$ and a set of missing pixels $\mathcal{B}$, as shown in Figure 8.7. $\mathcal{A}$ is called *support area* while $\mathcal{B}$ is called *missing area*. Then, the basic idea is to approximate the support area $\mathcal{A}$ by a weighted linear combination of basis functions. Using basis functions with extrapolating properties that are defined over the entire area $\mathcal{L}$, the approximation of $\mathcal{A}$ provides an estimation of the missing area $\mathcal{B}$ at the same time. Periodic functions such as the discrete cosine transform (DCT) or discrete Fourier transform (DFT) basis functions are suited for this domain as they are able to extend a signal periodically.

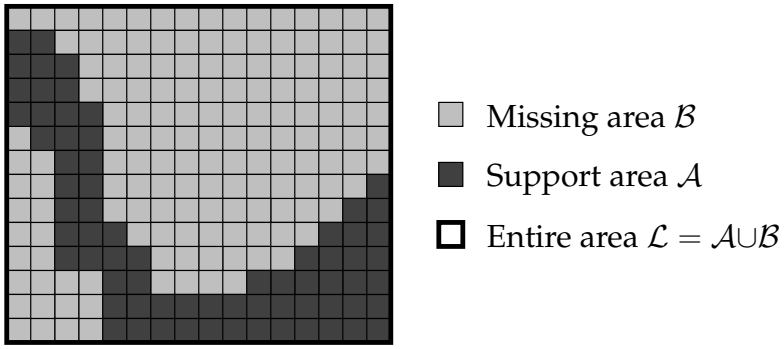


Figure 8.7: An illustration of the areas used for frequency selective extrapolation [83].

The approximation of $\mathcal{A}$ is realised by a parametric model g that is defined by the basis functions $\varphi_{k,l}$ and their expansion coefficients $c_{k,l}$:

$$g(m,n) = \sum_{(k,l)\in\mathcal{K}} c_{k,l}\varphi_{k,l}(m,n), \tag{8.8}$$

where $\mathcal{K}$ is the set of all basis functions.

The coefficients $c_{k,l}$ shall be chosen in a way that they minimise an error criterion, namely the squared difference between the parametric model and the actual image data denoted by f in a support area. Using a weighting function:

$$\mathrm{w}(m,n) = \begin{cases} \rho(m,n) & (m,n) \in \mathcal{A} \\ 0 & (m,n) \in \mathcal{B} \end{cases} \tag{8.9}$$

it is then expressed by:

$$\mathrm{E}_{\mathcal{A}} = \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\big(f(m,n) - g(m,n)\big)^2 \tag{8.10}$$

ρ can be used to emphasise certain parts of the support area $\mathcal{A}$. If no further weighting is desired, then it is set to $\rho(m,n) = 1$.

The common method to minimise the error criterion (8.10) would be setting its derivative with respect to the coefficients $c_{k,l}$ to zero. However, this method cannot be applied here, as the number of all basis functions $|\mathcal{K}|$, which is equivalent to the number of coefficients $c_{k,l}$, equals the number of pixels in the entire area, $|\mathcal{L}|$, while the number of known pixels $|\mathcal{A}|$ is smaller than $|\mathcal{L}|$. That means we have to deal with an underdetermined problem.

Instead, Meisinger and Kaup proposed to use an iterative technique of successive approximations. With this technique, the coefficients $c_{k,l}$ can be computed iteratively using two steps per iteration. First, a basis function is selected, then its corresponding coefficient is updated, fulfilling the condition that a maximal reduction of the error criterion $\mathrm{E}_{\mathcal{A}}$ is obtained with these values.

Let $g^{\nu}(m,n)$ be the parametric model and $c_{k,l}^{\nu}$ the coefficient at the νth iteration, then Equation (8.8) becomes:

$$g^{\nu}(m,n) = \sum_{(k,l)\in\mathcal{K}} c_{k,l}^{\nu}\varphi_{k,l}(m,n) \ . \tag{8.11}$$

Initially, all coefficients $c_{k,l}^{0}$ are equal to zero and consequently the parametric model $g^{0}(m,n)$ equals zero as well. Assuming a certain basis function $\varphi_{u,v}$ has already been chosen in an iteration ν, we can express the resulting parametric model for iteration $\nu+1$ based on the previous one and the weighted new basis function:

$$g^{\nu+1}(m,n) = g^{\nu}(m,n) + \Delta c\varphi_{u,v}(m,n), \tag{8.12}$$

where Δc is the desired coefficient update. It can be obtained by minimizing the new error criterion:

$$\mathrm{E}_{\mathcal{A}}{}^{\nu+1} = \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\left(f(m,n) - g^{\nu}(m,n) - \Delta c\varphi_{u,v}(m,n)\right)^2 \tag{8.13}$$

with respect to Δc.

For the sake of simplicity, the term $\mathrm{w}(m,n)(f(m,n) - g^{\nu}(m,n))$ is substituted by a new variable $\mathrm{r}_{\mathrm{w}}^{\nu}(m,n)$ called the weighted residual error. This yields the following equation for Δc:

$$\Delta c = \frac{\sum_{(m,n)\in\mathcal{L}} \mathrm{r}_{\mathrm{w}}^{\nu}(m,n)\varphi_{u,v}(m,n)}{\sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\varphi_{u,v}(m,n)^2} \tag{8.14}$$

At the end of this step, the coefficient update Δc calculated according to this equation is applied to the coefficient corresponding to the chosen basis function:

$$c_{u,v}^{\nu+1} = c_{u,v}^{\nu} + \Delta c \tag{8.15}$$

The question of which basis function $\varphi_{u,v}$ is chosen in an iteration ν also depends on the error criterion. In order to minimise it, we calculate the difference $\Delta\mathrm{E}_{\mathcal{A}}{}^{\nu} = \mathrm{E}_{\mathcal{A}}{}^{\nu} - \mathrm{E}_{\mathcal{A}}{}^{\nu+1}$ and maximise it. Taking into account that the residual error $\mathrm{r}_{\mathrm{w}}^{\nu}$ is orthogonal to the basis function $\varphi_{u,v}$ to be selected, we obtain:

$$\Delta\mathrm{E}_{\mathcal{A}}{}^{\nu} = \Delta c^2 \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\big(\varphi_{u,v}(m,n)\big)^2 \ . \tag{8.16}$$

Using (8.14), this can be extended to:

$$\Delta \mathrm{E}_{\mathcal{A}}{}^{\nu} = \frac{\Big(\sum_{(m,n)\in\mathcal{L}} \mathrm{r}_{\mathrm{w}}^{\nu}(m,n)\varphi_{u,v}(m,n)\Big)^2}{\sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\big(\varphi_{u,v}(m,n)\big)^2} \quad . \tag{8.17}$$

Finally, the index of the basis function is selected by:

$$(u,v) = \mathrm{argmax}_{(u,v)} \Delta \mathrm{E}_{\mathcal{A}}{}^{\nu} \quad . \tag{8.18}$$

The algorithm terminates when the maximal reduction of the error criterion $\Delta \mathrm{E}_{\mathcal{A}}{}^{\nu}$ drops below a certain threshold $\Delta \mathrm{E}_{\min}$, or after a fixed number of iterations $\nu_{\max}$.

So far, the algorithm has been described generally for any class of basis functions with extrapolating properties. According to Meisinger and Kaup, 2D-DFT basis functions are a good choice for signal extrapolation in images, as they not only contain horizontal and vertical structures like the 2D-DCT basis functions, but also diagonal ones. Therefore, 2D-DFT basis functions are chosen for the actual application of the frequency selective extrapolation principle to the domain of error concealment. The special properties of the 2D-DFT basis function allow for and require some specialisation of the algorithm which are discussed in this section.

Using the 2D-DFT basis functions $\varphi_{k,l}(m,n) = e^{i2\pi/Mmk} e^{i2\pi/Nnl}$ with M being the height and N the width of the area $\mathcal{L}$, the parametric model is rewritten as:

$$g^{\nu}(m,n) = \frac{1}{2MN} \sum_{(k,l)\in\mathcal{K}} \big(c_{k,l}^{\nu}\varphi_{k,l}(m,n) + c_{M-k,N-l}^{\nu}\varphi_{M-k,N-l}(m,n)\big) \tag{8.19}$$

and the coefficients $c_{k,l}$ become DFT coefficients.

This modification assures that the parametric model provides only real values by utilising a conjugate complex symmetry that the coefficients fulfil:

$$c_{k,l} = \bar{c}_{M-k,N-l} \tag{8.20}$$

$$\varphi_{k,l}(m,n) = \bar{\varphi}_{M-k,N-l}(m,n) \tag{8.21}$$

Figure 8.8 illustrates the conjugate complex symmetry as well as a reduced search area for the selection of the basis functions. It is sufficient to choose basis functions from the tagged area only, because all other basis functions have symmetric counterparts in this area and will be updated together with them.

Consequently, the error criterion from iteration ν to iteration $\nu+1$ changes to:

$$\begin{aligned} \mathrm{E}_{\mathcal{A}}{}^{\nu+1} = \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\Big(f(m,n) - g^{\nu}(m,n) \\ - \frac{1}{2MN}\big(\Delta c\varphi_{u,v}(m,n) + \Delta\bar{c}\varphi_{M-u,N-v}(m,n)\big)\Big)^2 \quad . \end{aligned} \tag{8.22}$$

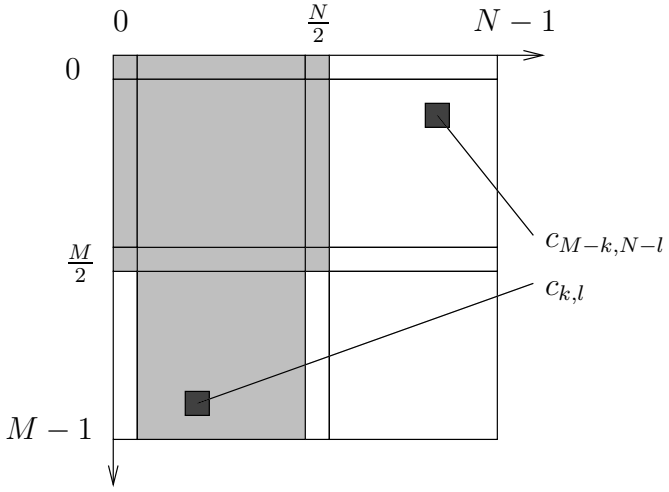


Figure 8.8: Conjugate complex symmetry and reduced search area (marked in gray) [83].

Taking its derivative with respect to Δc and setting it to zero yields the following:

$$\begin{aligned} &\Delta c \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\varphi_{u,v}(m,n)\varphi_{M-u,N-v}(m,n) \\ &+\Delta\bar{c} \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\varphi_{M-u,N-v}(m,n)\varphi_{M-u,N-v}(m,n) \\ &= 2MN \sum_{(m,n)\in\mathcal{L}} \mathrm{r}_{\mathrm{w}}^{\nu}(m,n)\varphi_{M-u,N-v}(m,n) \ . \end{aligned} \tag{8.23}$$

Δc is then obtained by solving the above, and the coefficient update is adapted to the real-valued parametric model as well:

$$c_{u,v}^{\nu+1} = c_{u,v}^{\nu} + \Delta c \tag{8.24}$$

$$c_{M-u,N-v}^{\nu+1} = c_{M-u,N-v}^{\nu} + \Delta\bar{c} \tag{8.25}$$

The reduction of the error criterion can be calculated for the special case of 2D-DFT basis functions by:

$$\begin{aligned} \Delta\mathrm{E}_{\mathcal{A}}{}^{\nu} = \frac{1}{(2MN)^2}\Big(&|\Delta c|^2 \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\varphi_{u,v}(m,n)\bar{\varphi}_{u,v}(m,n) \\ &+\Re\{\Delta c^2 \sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\big(\varphi_{u,v}(m,n)\big)^2\}\Big), \end{aligned} \tag{8.26}$$

where the operator $\Re$ takes the real part of the complex term in braces.

The computation of (8.23) and (8.26) would involve several Fourier transforms and hence be computationally expansive. Therefore, Meisinger and Kaup also propose a frequency domain implementation based on the fact that a multiplication of the weighting function

with the complex exponential $\varphi_{u,v}$ is equivalent to a shift of its DFT by u and v:

$$\sum_{(m,n)\in\mathcal{L}} \mathrm{w}(m,n)\varphi_{u,v}(m,n)\bar{\varphi}_{k,l}(m,n) = \mathrm{W}(k-u,l-v) \ . \tag{8.27}$$

Using this equation, (8.23) can be expressed in the frequency domain as:

$$\Delta c\mathrm{W}(0,0) + \Delta\bar{c}\mathrm{W}(2u,2v) = 2MN\mathrm{R}_{\mathrm{w}}^{(\nu)}(u,v), \tag{8.28}$$

yielding a frequency domain formula for the coefficient update when solved for Δc:

$$\Delta c = \begin{cases} MN\frac{\mathrm{R}_{\mathrm{w}}^{(\nu)}(u,v)}{\mathrm{W}(0,0)}, & (u,v)\in M \\ 2MN\frac{\mathrm{R}_{\mathrm{w}}^{(\nu)}(u,v)\mathrm{W}(0,0)-\bar{\mathrm{R}}_{\mathrm{w}}^{(\nu)}(u,v)\mathrm{W}(2u,2v)}{\mathrm{W}(0,0)^2-|\mathrm{W}(2u,2v)|^2}, & \text{otherwise.} \end{cases} \tag{8.29}$$

$\Delta\mathrm{E}_{\mathcal{A}}{}^{\nu}$ can also be expressed in the frequency domain:

$$\Delta\mathrm{E}_{\mathcal{A}}{}^{\nu} = \begin{cases} 2\frac{\mathrm{R}_{\mathrm{w}}^{(\nu)}(k,l)^2}{\mathrm{W}(0,0)}, & (k,l)\in M \\ 2\frac{\mathrm{R}_{\mathrm{w}}^{(\nu)}(k,l)^2\mathrm{W}(0,0)-\Re\{\mathrm{R}_{\mathrm{w}}^{(\nu)}(k,l)^2\bar{\mathrm{W}}(2k,2l)\}}{\mathrm{W}(0,0)^2-|\mathrm{W}(2k,2l)|^2}, & \text{otherwise.} \end{cases} \tag{8.30}$$

The case differentiation is necessary due to the conjugate complex symmetry. The set M contains the four coefficients that are symmetric to themselves:

$$M = ((0,0),(0,\frac{N}{2}),(\frac{M}{2},0),(\frac{M}{2},\frac{N}{2})) \tag{8.31}$$

To complete the frequency domain implementation, the change of the residual error r_{w} from iteration ν to iteration $\nu+1$ is computed in the frequency domain by:

$$\mathrm{R}_{\mathrm{w}}^{(\nu+1)}(k,l) = \mathrm{R}_{\mathrm{w}}^{(\nu)}(k,l) - \frac{1}{2MN}\big(\Delta c\mathrm{W}(k-u,l-v) + \Delta\bar{c}\mathrm{W}(k+u,l+v)\big) \tag{8.32}$$

We use this frequency selective extrapolation method in order to evaluate our results. The algorithm is applied individually to every lost macroblock of an image, line by line from the top left to the bottom right. At a time, a lost macroblock is considered to be the missing area $\mathcal{B}$, while a 13 pixel wide border in all directions is used as support area $\mathcal{A}$. The resulting entire area $\mathcal{L}$ is a square window of 42×42 pixels centered around the lost macroblock.

The parameters were chosen according to the fixed parameter set proposed by Meisinger and Kaup in [111]. The weighting function (8.9) is realised by an isotropic model:

$$\rho(m,n) = \hat{\rho}^{\sqrt{(m-\frac{M}{2})^2+(n-\frac{N}{2})^2}} \tag{8.33}$$

with $\hat{\rho}=0.74$. If the support area contains unknown regions of the image, which happens in case of consecutive loss, these regions are excluded from the computation by setting the corresponding regions of the weighting function to zero. Already concealed macroblocks are included in the concealment of following macroblocks, but further weighted by 0.1. The algorithm stops after $\nu_{\max}=11$ iterations or when the error decrease drops below $\Delta\mathrm{E}_{\min}=15$.

For colour images, Meisinger and Kaup recommend working in the YC_bC_r colour space, applying the algorithm to the luminance and the two chrominance channels individually. Since we used RGB images extracted from the test sequences for our experiments, they had to be converted back into the YC_bC_r colour space before the error concealment algorithm could be applied to every channel.

Figure 8.9 shows an example of this method. Figure 8.9(b) shows the frame with the simulated loss and Figure 8.9(c) shows the frame concealed by the frequency selective extrapolation of Meisinger and Kaup.

(a) Original

(b) Corrupted

(c) Concealed

Figure 8.9: Example of the frequency selective restoration [111, 83] for a frame of the sequence "Hiragasy".

Temporal Error Concealment

Temporal error concealment exploits the temporal redundancies within an image sequence. More precisely, it tries to find information missing image data in the preceding or succeeding

frames. So basically, temporal error concealment of a block in an image means to reconstruct a motion vector that points to the desired area in a reference image.

The simplest solution is to assume a motion vector of 0, i. e. replacing lost image data by the values found at the same place in the previous image. While this approach works out for sequences with very little motion, it fails in all other cases.

When working on MPEG video images, P- and B-frames already contain motion vectors. For a macroblock whose motion vector is lost, it can be estimated by taking the average or the median of the available surrounding macroblocks. This method usually provides acceptable estimations, however, it can only be applied when motion vectors are available for an image. Thus, it cannot be directly applied to I-frames as they do not contain motion vectors.

A more elaborate method is the decoder motion-vector estimation introduced by Zhang et al. in [211]. Since it is intended to be used for MPEG-2 coded video, it is developed to conceal 16×16 pixel macroblock losses, but the concept could also be applied to errors of different characteristics. The main idea of the algorithm is to take a 2 to 8 pixel border around the lost macroblock and perform a search in the previous image for the best match to this border, similar to the motion compensation technique used in MPEG-2 coding. The area surrounded by this best match in the previous image is then taken as an estimation of the lost macroblock. Decoder motion-vector estimation outperforms the previous methods and can be computed efficiently when narrowing down the search area.

Another interesting temporal error concealment technique has been proposed by Suh and Ho [182]. They suggest using optical flow fields in order to estimate motion vectors. In order to recover the motion vector of the lost block, optical flow fields are computed for correctly decoded neighbouring macroblocks denoted as OFR in Figure 8.10. Then, taking the average of the optical flow vectors within the macroblock that is in touch with the lost macroblock (denoted as MVEB in Figure 8.10) yields the motion vector of the lost block.

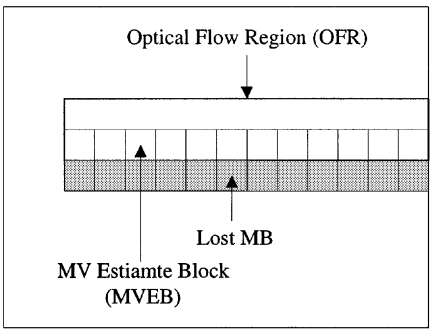


Figure 8.10: Regions of optical flow estimation for error concealment [182].

Zhao [213] already considered super-resolution for error concealment. His approach consists in computing flow fields from neighbouring frames to the damaged frame and performing image interpolation to repair the damaged frame. Afterwards, a super-resolution algorithm is applied to the repaired frame in order to improve its resolution. Nevertheless, considering super-resolution for error concealment as well, our approach differs from that of Zhao. We present it below.

8.2.2 Error Concealment using Super-Resolution Mosaicing

We see the interest of super-resolution mosaicing for temporal error concealment as it allows the recovery of meaningful information even if multiple errors occur in successive frames. In addition, it seems more interesting than spatial error concealment as real information can be recovered from original frames (in absence of occlusions and a scene change).

Here, we only consider the error concealment of I-frames. However, this work can be extended to P and B-frames taking into account the error propagation due to motion compensation. Our approach consists in progressively constructing a mosaic during the decoding of a shot. Then, when a transmission error occurs in the current I-frame, the missing information can be concealed thanks to the constructed mosaic. Lost areas can be replaced with the corresponding regions in the mosaic.

We consider two different approaches for the mosaic construction. The first approach is to construct the mosaic from decoded full-resolution I-frames, whereas the second consists in using only partially decoded I-frames, i.e. DC images, and to increase the resolution by a factor $\varsigma = 8$. The resulting mosaic will have the same resolution than the video frames. To examine the influence of super-resolution, mosaics were created both with 4 iterations of the super-resolution algorithm of Chapter 4 and without applying any super-resolution at all.

We assume that transmission errors have already been detected. A method for this can be found in [91]. Thus, we mask out the errors, so that they do not appear in the mosaic. As long as the loss rate is not too high and the errors are not distributed unfavourable, we still obtain an almost complete mosaic, as most of the errors in one image are overlapped by intact parts of the other images.

Hence, to exclude errors from the mosaicing process, we introduce a mask $L(k)$ for each low-resolution image $G(k)$:

$$L(x,y) = \begin{cases} 0 & \text{if } (x,y) \text{ is lost in } G(k) \\ 1 & \text{if } (x,y) \text{ is available} \end{cases} \tag{8.34}$$

Then, the initial guess of the super-resolution mosaic F^0 (4.58) can be computed as:

$$F^0 = \mu(K) \sum_{k=1}^{K} \mathrm{T}(k).\mathrm{S}.L(k).G(k) \tag{8.35}$$

where $G(k)$ are the observed low-resolution images, S the upsampling operator by a factor ς, $\mathrm{T}(k)$ the geometric transformation from kth low-resolution image to the super-resolution mosaic, and $\mu(\mathbf{p}, K) = \frac{1}{|\mathbf{p}|}$ with $|\mathbf{p}|$ as the number of available pixels at position $\mathbf{p}$.

The simulation process (4.56) of the low-resolution images remains unchanged as it only depends on the current guess of the super-resolution mosaic F^i:

$$G^i(k) = \mathrm{S}^{-1}.\mathrm{B}(k) * \left[\mathrm{T}^{-1}(k).F^i(k)\right] \tag{8.36}$$

where $G^i(k)$ are the simulated low-resolution images, $\mathrm{T}^{-1}(k)$ the geometric transformation from the super-resolution mosaic to the kth low-resolution image, $\mathrm{B}(k)$ is the PSF of the kth

low-resolution image, $*$ is the convolution operator and S^{-1} the downsampling operator inverse to S.

However, the masks $L(k)$ have to be introduced in the backprojection process (4.57):

$$F^{i+1} = F^i + \mu(K) \sum_{k=1}^{K} \mathrm{T}(k).\mathrm{R}(k) * \left[\mathrm{S}.L(k) \left[G(k) - G^i(k)\right]\right] \tag{8.37}$$

where $\mathrm{R}(k)$ is the restoration filter defined by $\mathrm{B}(k)$.

Note, that contrary to (4.57) we do not use the regularisation operator A (see Section 4.2.5) as it was developed for highly undersampled image such as DC images, but we apply here the super-resolution method to full-resolution and low-resolution frames. Furthermore, in case of decoded full-resolution I-frames $\varsigma = 1$ and in case of DC images of I-frames $\varsigma = 8$. The computation of the geometric transformations $\mathrm{T}(k)$ was presented in Section 3.2.2. In our experiments, we used the anisotropic Gaussian blur model for the PSF $\mathrm{B}(k)$ and the pseudo-inverse restoration filter for $\mathrm{R}(k)$ (see Section 4.2.3).

Contrary to the mosaicing scheme of Chapter 4, we do not perform illumination correction of the input sequence. Since the mosaic is here constructed progressively this would mean that illumination correction can only be applied recursively with respect to the previously decoded I-frame. We showed in Section 3.2.4 that this accumulates estimation errors so that the result is not satisfying. Moreover, it is preferable in this application to correct illumination with respect to the frame to be concealed. Thus, a possibility is to adjust the illumination of the mosaic with respect to that frame. However, a more efficient and less costly approach is to correct illumination of the patches to be inserted in the frame with respect to the neighbourhood. This is in the perspective of future work.

Finally, we obtain a concealed image $\bar{G}(k)$ by replacing lost regions of $G(k)$ by the corresponding areas in the mosaic:

$$\bar{G}(k) = \begin{cases} \mathrm{T}^{-1}(k).F & \text{where } G(k) \text{ was erroneous} \\ G(k) & \text{otherwise} \end{cases} \tag{8.38}$$

Using the above modifications of the super-resolution mosaicing method of Chapter 4, we can produce a mosaic from partially erroneous low-resolution images as illustrated in Figure 8.11. When the camera is moving fast and a loss appears in the new regions entering the frame, they cannot be concealed as they are only present in the current frame. Therefore, we introduce a delay of one I-frame assuming that the next I-frame contains the desired information. For the sake of computational cost, we did not use the whole shot in our experiments for the mosaic construction. (We showed in Section 4.3.2 that mosaic construction using full-resolution frames is very costly.) In fact, we used two I-frames in the past and one I-frame in the future in order to conceal the current I-frame. Hence, we chose the frame to be concealed as the reference frame for mosaic construction. In this case the geometric transformation in (8.38) reduces to an identity transform. Another advantage of this choice of reference frame is that the mosaic is at the same resolution as the frame to be concealed. This approach is similar to the dynamic mosaicing approach we presented in Section 3.1.

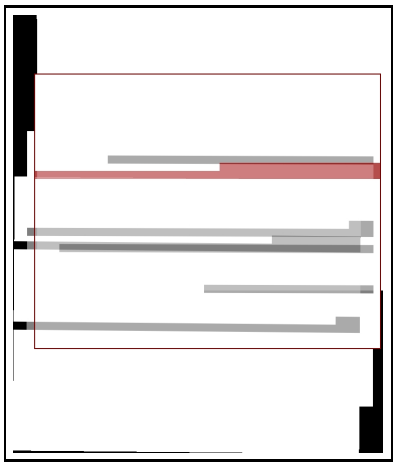


Figure 8.11: A mosaic constructed of error masks. The dark red frame marks the reference image.

8.3 Results

Our first experiment consists in using an artificial pattern of isolated block loss as shown in Figure 8.12. This pattern is useful to study the performance of the error concealment techniques under "optimal", isolated loss conditions. Thus, we applied the isolated loss pattern only to one frame of the sequence "Chancre1" (see Figure 3.21(b)) which has to be concealed (the reference frame) where as the other frames were intact.

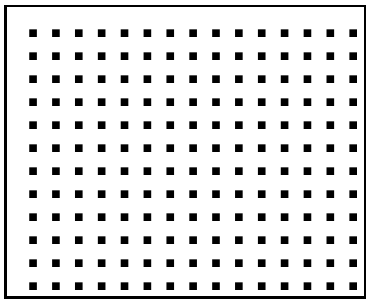


Figure 8.12: The isolated loss pattern.

For the second experiment, we simulated error patterns for the sequences "Chancre1" and "Tympanon" (Figure 3.27(a)). The corrupted sequences are shown in Figures 8.13 and 8.6, respectively. These error patterns are randomly generated using the error simulation algorithm presented in Section 8.1 using the Gilbert model with $p = 0.001$ and $q = 0.1$.

As an objective measure of concealment quality, we computed the traditional peak signal to noise ration (PSNR). It is given by [100]:

$$\text{PSNR} = 10 \log_{10} \frac{m^2}{\text{MSE}} \tag{8.39}$$

Figure 8.13: The simulated loss by the Gilbert model for the sequence “Chancre1”.

where m is the maximal pixel value (usually 255). The MSE (3.46) is computed only on concealed pixels between the original image $I(k)$ and the concealed image $\bar{I}(k)$.

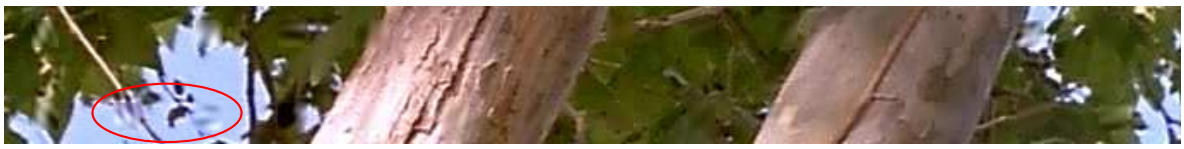
In the following, we evaluate our results in terms of PSNR and visual quality and compare them with the frequency selective extrapolation method of Meisinger and Kaup [111, 83] (see Section 8.2.1).

Figure 8.14 shows a detail of frame 6782 of the sequence “Chancre1”. The isolated loss pattern (Figure 8.12) was applied to this frame, then the errors were concealed using the different techniques discussed above.

The concealed blocks are heavily blurred for the super-resolution mosaicing using DC images (Figure 8.14(f)) due to the high zoom factor. The four iterations of super-resolution do not improve the image quality significantly (Figure 8.14(e)). That is because super-resolution as used in this work becomes an ill-posed problem for high zoom factors. Too many pixels in the super-resolution image have to be reconstructed using too little information available in the low-resolution sequence. Thus, high frequencies can not be reconstructed. This result is not satisfying and confirms the assumption that current super-resolution algorithms only perform well for relatively small zoom factors. For example, Lin and Shum [101] consider $\varsigma = 1.6$ as the fundamental limit for practical situations, or $\varsigma = 2.5$ as an optimistic attempt under beneficial noise conditions. Therefore, the initial DC resolution is inappropriate for this task.

On the other hand, super-resolution mosaicing at full resolution produces good results. Generally, the structure of the content of the lost blocks is reconstructed correctly, already without any iteration of super-resolution (Figure 8.14(d)). These reconstructions are slightly blurred, though, because several images are averaged in the mosaicing process. Applying some iterations of super-resolution solves the problem (Figure 8.14(c)).

(a) Erroneous



(b) Frequency Selective Extrapolation, PSNR=19.8

(c) Super-Resolution Mosaicing (full resolution, 4 iterations), PSNR=17.5

(d) Super-Resolution Mosaicing (full resolution, 0 iterations), PSNR=17.6

(e) Super-Resolution Mosaicing (DC resolution, 4 iterations), PSNR=16.9

(f) Super-Resolution Mosaicing (DC resolution, 0 iterations), PSNR=16.9

(g) Original

Figure 8.14: A detail of frame 6782 of the Chancre sequence.

However, there is another obvious problem, concerning the lost blocks in the area of the right trunk for example (red ellipses in Figures 8.14(c) and 8.14(d)). Due to differences in luminance between the images used to construct the mosaic, some of the concealed blocks are brighter than their surroundings. A possible solution for this issue consists in applying an illumination correction e.g. that one we proposed in Section 3.2.4.

The detail that is concealed using frequency selective extrapolation appears more homogeneous to the viewer, as all concealed blocks are of appropriate brightness. Therefore, the PSNR value is better than for super-resolution mosaicing, but taking a closer look we notice some blocks containing artefacts. Especially blocks in monotonous regions adjacent to an object suffer from these artefacts, as they appear when higher frequencies caused by edges are wrongly extrapolated into monotonous regions (red ellipse in Figure 8.14(b)) .

Having examined optimal, isolated loss conditions, we consider now "real" losses as we simulated for the sequences "Chancre1" (Figure 8.13) and "Tympanon" (Figure 8.6). The PSNR values for the concealed I-frames of both sequences are shown in Figures 8.15 and 8.16. As expected the values for error concealment using super-resolution mosaicing with DC images are poor as high frequencies are not reconstructed. They are below 10 dB in most cases.

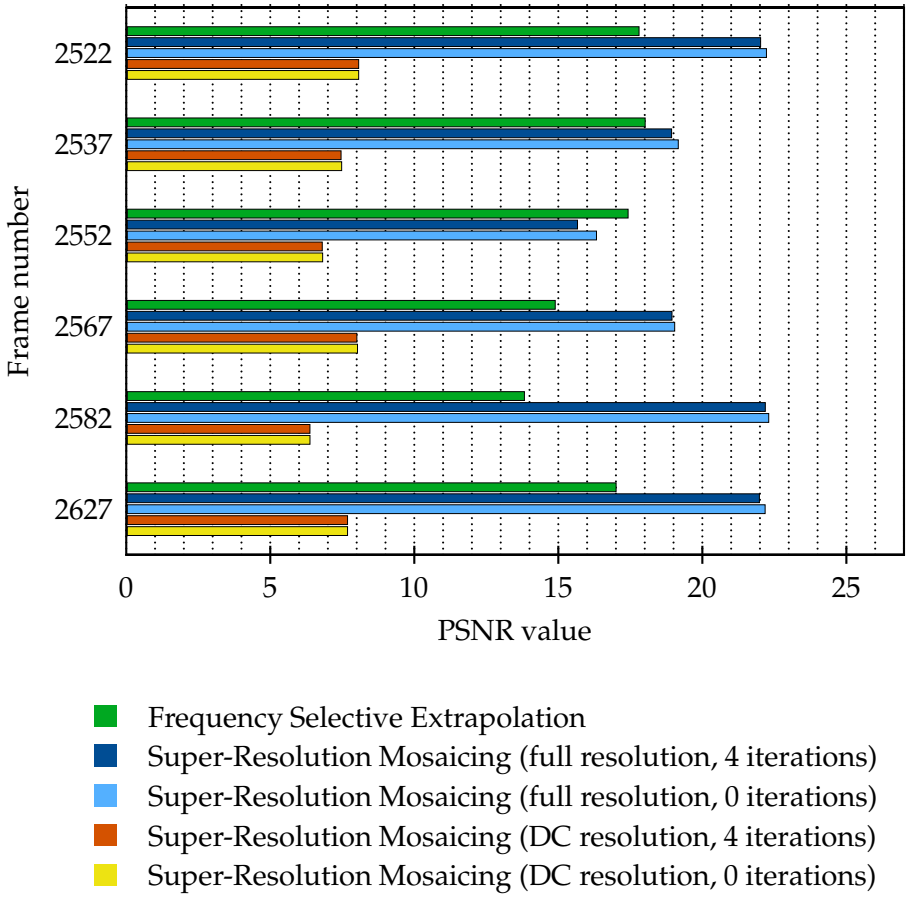


Figure 8.15: PSNR values for the sequence "Tympanon".

Super-resolution mosaicing at full resolution however gives good results in terms of PSNR. For both sequences, "Tympanon" and "Chancre1", the average PSNR value is higher than 20 dB which is better than the frequency selective extrapolation (16.5 dB and 19 dB respectively).

Surprisingly, the PSNR values for mosaicing with 4 iterations of super-resolution are slightly lower than for mosaicing without any iteration of super-resolution. However, PSNR values do not always correspond to the quality as perceived by a human viewer [150]. Thus, we will give actual examples below and analyse them visually.

In contrast to the optimal conditions from the previous example, the situation changes when it comes to consecutive loss as seen in the detail from the Tympanon sequence shown in Figure 8.17. Less data is available for the spatial error concealment, as the predecessor and the successor of a macroblock have also been lost. As the error concealment is applied macroblock by macroblock anyways, the individual macroblocks are clearly visible within the two lost rows.

Using super-resolution mosaicing, the reconstructions of the rows are of one piece, thus no explicit blocking artefacts are visible. The result is better, as well in the high frequency area around the eyes and in the area of the hand.

When two or more consecutive rows of macroblocks are lost, the results of frequency selective extrapolation are even worse, as seen in Figure 8.18. Evidently, the reason for this is that for the concealment of a lost block only three intact and one concealed macroblock are available. Better results might be achieved by applying the algorithm on the two lost rows together.

The version using super-resolution mosaicing shows a more convincing result, but reveals another problem of the algorithm: The concealed region contains a good reconstruction, but is slightly displaced. Most likely, this is caused by an inaccurate estimation of the motion between the images used for mosaicing. In this case the method for motion correction or motion reestimation we presented in Chapter 3 can be applied.

8.4 Conclusion

We presented in this chapter the application of our super-resolution mosaicing method to error concealment in I-frames. Therefore, we surveyed classical loss models for transmission errors. We consider the Gilbert model as sufficient to model transmissions error of MPEG-2 compressed video and simulate them at macroblock basis.

Based on the super-resolution method of Chapter 4, we construct a mosaic from partially erroneous I-frames. Thanks to temporal redundancy of several frames, the mosaic is likely to contain the lost information of the current frame. Thus, lost areas can be replaced with the corresponding regions in the mosaic.

First results are promising. For both test sequences using the super-resolution mosaicing approach we obtain an average PSNR higher than 20 dB for the concealed regions. These values are higher than the average PSNR values we obtain for the same sequences using frequency selective extrapolation. Also, in our visual comparison in case of a consecutive loss of

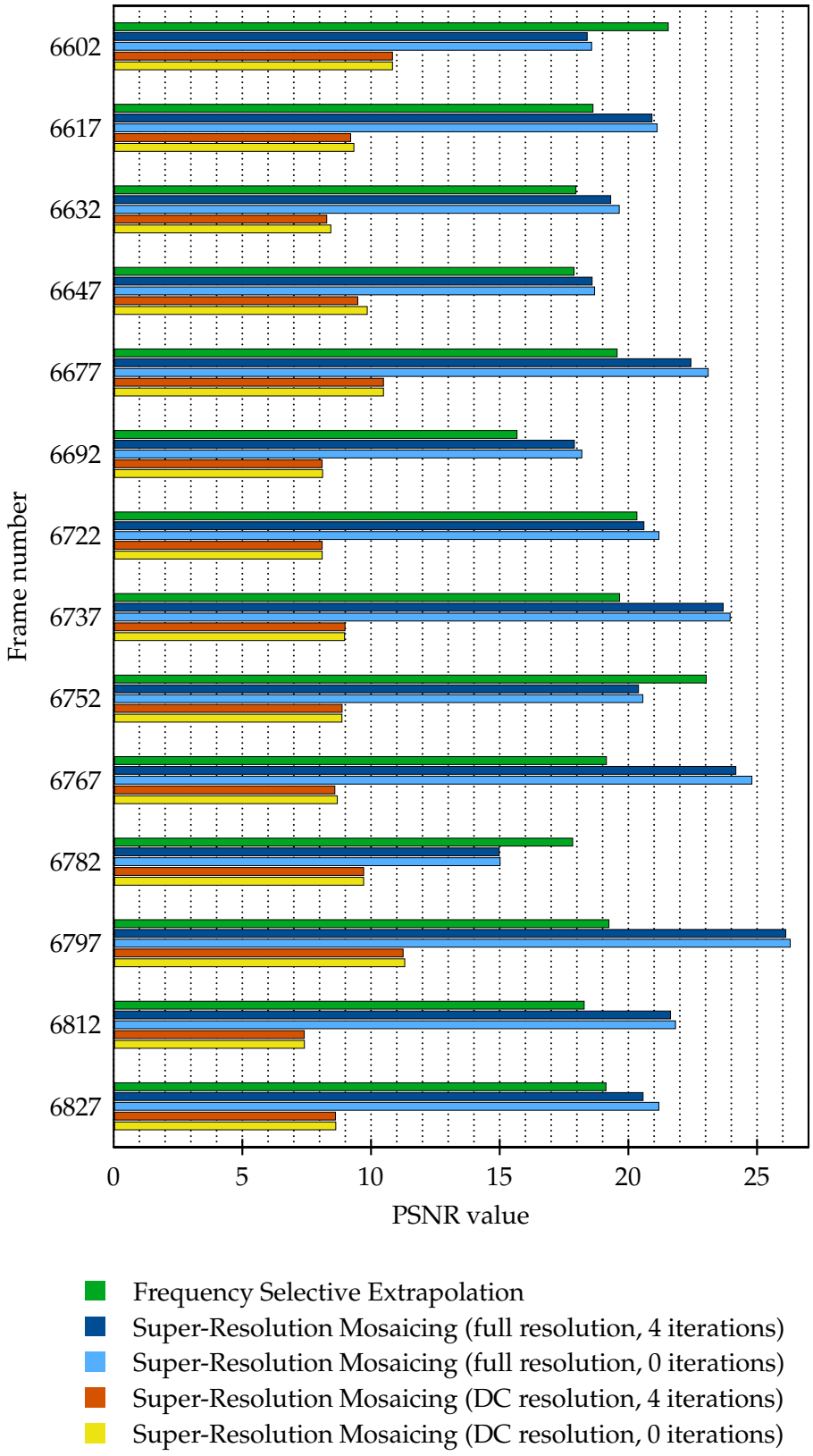
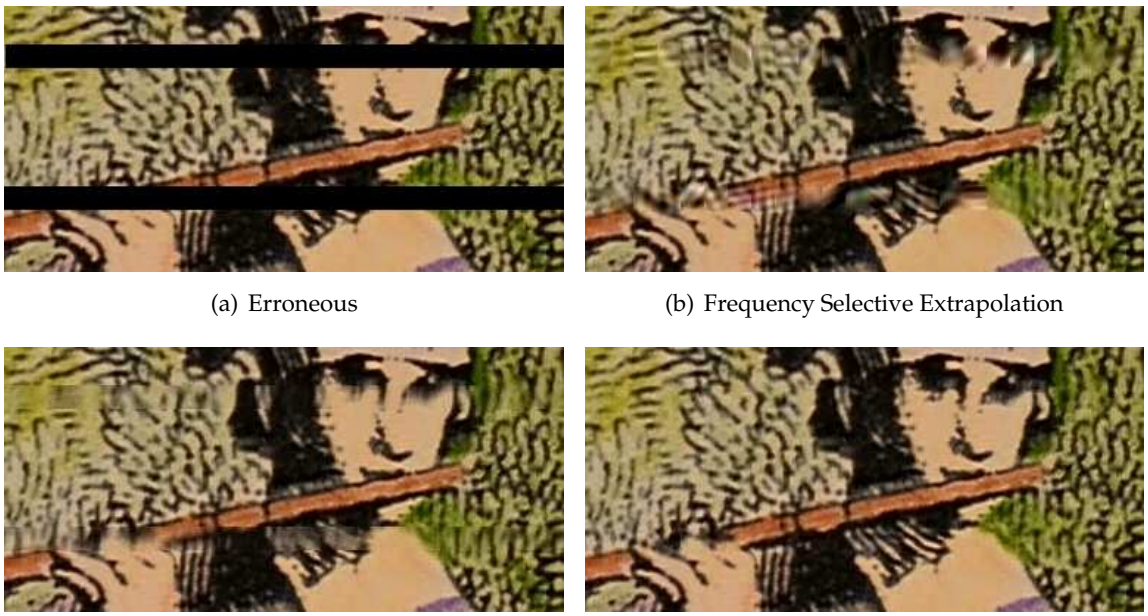


Figure 8.16: PSNR values for the sequence “Chancre1”.



(a) Erroneous

(b) Frequency Selective Extrapolation

(c) Super-Resolution Mosaicing (full resolution, 4 iterations)

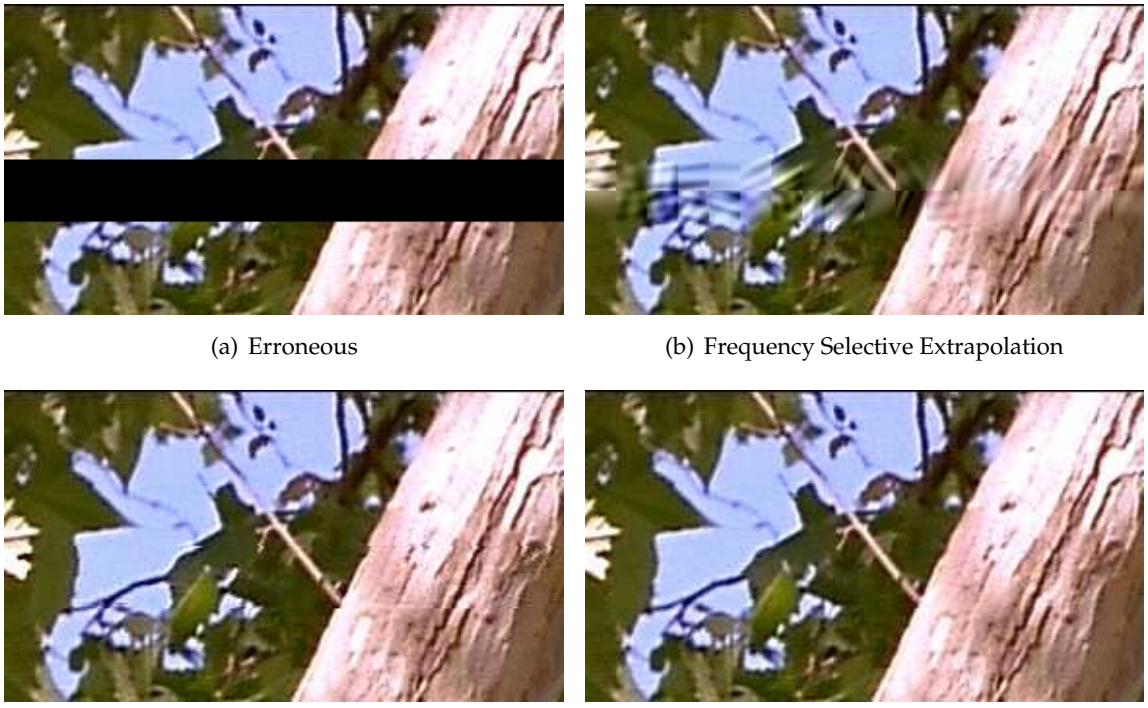
(d) Original

Figure 8.17: A detail of frame 2567 of the sequence "Tympanon".

macroblock the super-resolution mosaicing approach gives better results than frequency selective error concealment. The results are less blocky, and adapt better to the high-frequency pattern of the image.

Drawbacks of our method are that when motion is not accurately estimated the inserted patch may be blurred and slightly displaced. A solution is to correct motion by the method presented in Chapter 3.

Perspectives are numerous. First of all, this error concealment approach for I-frames can be extended to P and B-frames. Another perspective is the illumination correction of inserted patches with respect to the neighbourhood in the concealed frame, so that seams disappear. This can be achieved by the illumination correction method we presented in Chapter 3. Finally, the extension of this method to moving objects can be considered. To this end, the moving objects have to be detected e.g. using the method we presented in Chapter 3 and super-resolution method of Chapter 5 allowing the processing of objects can be adopted.



(a) Erroneous

(b) Frequency Selective Extrapolation

(c) Super-Resolution Mosaicing (full resolution, 4 iterations)

(d) Original

Figure 8.18: A detail of frame 6692 of the sequence “Chancre1”.

Chapter 9

Conclusion and Perspectives

In this PhD thesis, we proposed a complete framework for the construction of mosaics from MPEG-1/2 compressed video. With regard to the visualisation of video, we proposed summarising each shot by a background mosaic where the most representative objects of the shot are inserted.

In the last decade significant research work has been done in video analysis for mosaic construction. This can be explained by the fact that they allow for a quick understanding of a video scene content via interpreting a static image which gives an overview of the content of a video segment. Mosaic representations of video segments shot by the same camera are a part of the MPEG-4 video coding standard as they allow an efficient encoding of the scene by decomposing it into foreground objects and the background mosaic. They are also defined as a descriptor of a shot in the MPEG-7 standard which can be used for video retrieval.

Mosaicing methods proposed in literature typically perform on raw video, but today an enormous quantity of video content is already available in compressed form. Up to now the MPEG-2 standard is the most frequent encoding for video available in archives or coming from broadcast channels. Applying raw video mosaicing methods to compressed video requires its full decoding. Furthermore, aligning and warping frames in the mosaicing process requires the estimation of motion which is a costly operation. Thus, this is performed after decoding, while motion information was already available in the compressed stream. Thus, the first motivation of our work was to reuse motion information from the compressed stream such as P-frames motion vectors in order to reduce the computational work load. Our second motivation was that mosaics created from raw or decoded video are too large to display on the screen without scrolling or downsampling. Thus, they are not appropriate for video summarisation. To this end, we proposed creating mosaics of a lower resolution using DC images of I-frames.

For the mosaic construction several steps can be considered. We proposed in this thesis

a *complete framework for mosaic construction from the compressed stream*. We gave solutions to various problems which can be encountered in real video: the inaccuracy of motion vectors in the compressed stream, illumination changes, the presence of moving objects and difficulties in their segmentation. In addition, computational times are interesting, the method performs in about 3 times decoding time.

Taking into account the noisiness, sparseness, and inaccuracy of P-frame motion vectors, we proposed an appropriate scheme to align the DC images of I-frames. Global camera motion is estimated for P-frames using an robust estimator. However, this is not sufficient to obtain a complete motion trajectory in the sequence of DC images. Therefore, in case of I-frames we proposed extrapolating motion parameters from previous P-frames. Moreover, it happens that encoded motion vectors are inaccurate. We detected the concerned P-frames based on the DC image of the encoded motion estimation error and proposed a method, respectively, to correct the erroneous motion model or to reestimate it. To our knowledge such a *thorough study of motion from compressed streams* with its refinement has not been proposed in literature.

Another aspect of our mosaicing method is the removal of moving objects. We presented a method to detect moving objects in the sequence of DC images of I-frames. This allowed not only the removal of objects to create a background mosaic without artefacts, but also the posterior insertion of representative objects into the mosaic in order to achieved a complete description of the sequence. Additionally, we took into account the possibility that moving objects are not accurately segmented and proposed a solution.

Furthermore, we considered the harmonisation of illumination conditions of the input sequence in order to avoid seems in the mosaic. Therefore, we proposed a robust method to correct global illumination changes in DC images sequence. We obtain robustness by excluding moving objects, edges and textures from the computation, and rejecting outliers. We showed that the estimation errors can occur yet as the illumination model may be violated by noise. In order to limit error propagation, we proposed correcting illumination in the sequence hierarchically.

Nevertheless, the achieved mosaics are of very low resolution and are degraded by blur and aliasing artefacts due to the nature of DC images. To visualise the scene content the user would prefer a higher resolution and a better visual quality. Therefore, we presented *two super-resolution methods* in this thesis in order to enhance the resolution and to improve the visual quality of the mosaics.

The first super-resolution method is based on iterative backprojections performing restoration in frequency domain. We presented several blur models in order to approximate the unknown blur in DC images: isotropic Gaussian blur, anisotropic Gaussian blur and linear motion blur. Furthermore, we developed an efficient *method to estimate the blur parameters in motion direction* and an appropriate restoration scheme. We tested the blur models in combination with the pseudo-inverse filter and the Wiener filter. We obtained the best results for the linear motion blur in combination with the pseudo-inverse filter.

Due to the strong aliasing of DC images, it may be not possible to exactly superimpose DC images on textures and edges even if motion is estimated accurately. This causes arte-

facts in the super-resolution image which amplify along the iteration of the super-resolution algorithm. For this reason, we incorporated a *regularisation operator* in super-resolution algorithm which penalises edges and textures in the backprojection process.

Nevertheless, this method has some drawbacks. Due to restoration in frequency domain only global blur can be restored, but in real video more complex situations are encountered where local blurs appear e.g. due to moving objects. Thus, this method does not allow for the processing moving objects. Additionally, we obtain an important increase in computational time due to padding and successive Fourier transforms.

The objective of the second super-resolution method was to address the shortcomings of the first super-resolution method. Thus, restoration was realised in spatial domain allowing the *restoration of local blurs*. The motion of objects can be very complex and in case of DC images the objects are typically represented by only few pixels. Hence, its it very difficult to estimate the motion of objects and to superimpose them accurately enough for super-resolution. For this reason, we proposed super-resolving the background and restoring moving objects by single image restoration. We tested the different blur model in our restoration scheme. We obtained the best results for the anisotropic Gaussian blur whereas the results for the isotropic Gaussian blur were quite close. Here, the linear motion blur PSF does not seem an appropriate blur model due to the discretisation of the convolution kernel.

We compared both super-resolution methods using video sequences without moving objects. The super-resolution method based in frequency domain restoration performs slightly better in terms of error measures than the super-resolution method based on spatial domain restoration. Nevertheless, much less artefacts appear in the mosaics achieved by the super-resolution method based on spatial domain restoration. Additionally, we obtain an important gain in computational time as convolution is performed at low resolution and successive forward and inverse Fourier transforms are omitted.

In this thesis, we were interested in additional applications of our mosaicing method namely shot boundary detection, camera motion characterisation, and error concealment.

We used several components of the proposed mosaicing method for related indexing tasks: the shot boundary detection on I-frames and the characterisation of dominant camera motion. In the first case, we used the robust motion estimation from the compressed stream and the extrapolation of motion for I-frames, in order to warp one I-frame to another along the time and measure their similarity. Then, we defined a similarity measure based on the regularisation operator of our restoration approach. The latter allowed the efficient comparison in case of textured frames. In the second case, we completed the robust motion estimation by a temporal filtering and a probabilistic decision-making framework.

We presented the application of our super-resolution mosaicing method to error concealment in I-frames. Therefore, we surveyed classical loss models for transmission errors. We considered the Gilbert model as sufficient to model transmissions error of MPEG-2 compressed video and simulated them at macroblock basis. Based on the super-resolution mosaicing method, mosaics were constructed of partially erroneous I-frames. Thanks to temporal redundancy of several frames, the mosaic is likely to contain the lost information of the current frame. Thus, lost areas were replaced by the corresponding regions in the mosaic.

We compared our results with the frequency selective extrapolation method. In our experiments, we obtained an average PSNR higher than 20 dB for the concealed regions which is higher than the average value obtained by the frequency selective extrapolation method [111, 83]. Moreover, in case of a consecutive loss of macroblocks the super-resolution mosaicing approach gives visually better results than the frequency selective extrapolation. The results is less blocky, and adapts better to the high-frequency pattern of the image.

We recently started further experiences on biomedical images. We applied our super-resolution methods to image sequences acquired by Magnetic Resonance Imaging (MRI) in order to improve the in-plane resolution and restore blur due to organ motions, breathing and blood supply. First results are promising and show perspectives for further research.

We assumed for the mosaicing method that the choice of the representative objects, i.e. the best segmented objects, were already selected. Thus, a study of automatic selection of best segmented objects will be the next step in order to complete the mosaicing method.

High definition (HD) video is the new generation broadcasting system and is getting more and more popular. In our experiments, we observed that when encoding an HD video sequence in MPEG-2, motion information can not be recovered as the motion estimation of MPEG-2 encoder fails. Hence, new compression standards such as H.264/MPEG-4 AVC have been developed with an improved and more reliable motion estimation. We briefly outlined in Appendix B later MPEG standards and showed that if appropriate partially decoding is realised the presented methods can extended to them. Additionally, in HD video still the problem of motion blur appears due to nonzero aperture time. Thus, a further perspective of this work is the extension of the presented super-resolution mosaicing method to later MPEG standards such as H.264/MPEG-4 AVC and H.264/MPEG-4 SVC in order to restore motion blurs.

We presented error concealment as an additional application of our super-resolution mosaicing method without being exhaustive in the subject. Thus, a deeper research in this area and an extension of the method could be in the perspective of future work. An obvious extension would be taking into account P and B-frames or moving objects.

Anyhow, further application areas of the presented method can be considered such as the transmission of video via low bit-rate networks. Only low-resolution frames may be transmitted which are super-resolved at the receiver side in order to recover the full resolution of the video. Another application may be the restoration/resolution enhancement of regions of interest and their tracking in video surveillance.

Appendix A

Spatial Domain Restoration

Here, we present the mathematical derivation of the spatial domain restoration methods presented in Chapter 5. Both methods are derived from the deconvolution method of Van Cittert [31] and Jansson [78].

They model the blurred image as:

$$\tilde{F}(\gamma) = \int_{-\infty}^{+\infty} F(x)\mathrm{B}(\gamma - x)\,\mathrm{d}x \tag{A.1}$$

where $\tilde{F}$ is the blurred image, F the unknown optimal image and B is the PSF.

Then by approximating successively the desired optimal image as:

$$F(\gamma) = \tilde{F}(\gamma) + \Delta(\gamma) \tag{A.2}$$

the following iterative scheme results:

$$F^k(\gamma) = F^{k-1}(\gamma) + \left[F(y) - \int_{-\infty}^{+\infty} F^{k-1}(x)\mathrm{B}(\gamma - x)\,\mathrm{d}x\right] \tag{A.3}$$

We showed the derivation of this iterative scheme in Chapter 5.

In the following, we first derive the single image restoration method (5.31) from the deconvolution method of Van Cittert and Jansson. Then, we derive the super-resolution method (5.32) from our single image restoration method. Finally, we derive a relationship between the convolution of a low-resolution image with a low-resolution PSF and the down-sampled blurred super-resolution image.

A.1 Derivation of the Single Image Restoration Algorithm with Increase of Resolution

In this section, we demonstrate the derivation of the spatial domain restoration method (5.31) allowing to increase resolution from the successive approximations (A.2). Therefore, we consider an extended image formation model relating a super-resolution image with a low-resolution image by incorporating motion and downsampling:
Super-resolution image $\rightarrow$ motion $\rightarrow$ blur $\rightarrow$ downsampling $\rightarrow$ low-resolution image

Hence, we can rewrite Equations (A.1) and (A.2) as:

$$G(y) = \mathrm{S}^{-1} \int_{-\infty}^{+\infty} (\mathrm{T}.F(x))\, \mathrm{B}(\gamma - x)\, \mathrm{d}x \tag{A.4}$$

$$F(\gamma) = \mathrm{T}^{-1} \left(\mathrm{S}.G(y) + \Delta(\gamma)\right) \tag{A.5}$$

where T is the geometrical transformation from the desired super-resolution image F to the observed low-resolution image G, T^{-1} is the inverse geometric transformation, S is the upsampling operator, S^{-1} is the downsampling operator, and B the PSF.

Inserting (A.5) in (A.4):

$$\begin{aligned} G(y) &= \mathrm{S}^{-1} \int_{-\infty}^{+\infty} \left(\mathrm{T}\left(\mathrm{T}^{-1}\left(\mathrm{S}.G(y) + \Delta(\gamma)\right)\right)\right) \mathrm{B}(\gamma - x)\, \mathrm{d}x \\ &= \mathrm{S}^{-1} \int_{-\infty}^{+\infty} \left(\mathrm{S}.G(y) + \Delta(x)\right) \mathrm{B}(\gamma - x)\, \mathrm{d}x \\ &= \mathrm{S}^{-1} \int_{-\infty}^{+\infty} (\mathrm{S}.G(y))\, \mathrm{B}(\gamma - x)\, \mathrm{d}x + \mathrm{S}^{-1} \int_{-\infty}^{+\infty} \Delta(x) \mathrm{B}(\gamma - x)\, \mathrm{d}x \end{aligned} \tag{A.6}$$

Denoting:

$$G^1(y) = \mathrm{S}^{-1} \int_{-\infty}^{+\infty} (\mathrm{S}.G(x))\, \mathrm{B}(\gamma - x)\, \mathrm{d}x \tag{A.7}$$

and:

$$\Delta(\gamma) = \int_{-\infty}^{+\infty} \Delta(x) \mathrm{B}(\gamma - x)\, \mathrm{d}x \tag{A.8}$$

Then, (A.6) becomes:

$$\begin{aligned} G(y) &= G^1(y) + \mathrm{S}^{-1}.\Delta(\gamma) \\ \Leftrightarrow \Delta(\gamma) &= \mathrm{S}\left(G(y) - G^1(y)\right) \end{aligned} \tag{A.9}$$

Inserting (A.9) in (A.5):

$$\begin{aligned} F(\gamma) &= \mathrm{T}^{-1}\left(\mathrm{S}.G(y) + \mathrm{S}\left(G(y) - G^1(y)\right)\right) \\ &= \mathrm{T}^{-1}.\mathrm{S}.G(y) + \mathrm{T}^{-1}.\mathrm{S}\left(G(y) - G^1(y)\right) \end{aligned} \tag{A.10}$$

Assuming that:

$$F^0(\gamma) = \mathrm{T}^{-1}.\mathrm{S}.G(y) \tag{A.11}$$

and:

$$\begin{aligned} F^0(\gamma) &= \mathrm{T}^{-1}.\mathrm{S}.G(y) \mid \mathrm{T} \\ \Leftrightarrow \mathrm{T}.F^0(\gamma) &= \mathrm{S}.G(y) \end{aligned} \tag{A.12}$$

Inserting in (A.7):

$$G^1(y) = \mathrm{S}^{-1} \int_{-\infty}^{+\infty} \left(\mathrm{T}.F^0(x)\right) \mathrm{B}(\gamma - x)\,\mathrm{d}x \tag{A.13}$$

Finally, we derive from (A.10):

$$F^k(\gamma) = F^{k-1}(\gamma) + \mathrm{T}^{-1}.\mathrm{S}\left(G(y) - \mathrm{S}^{-1} \int_{-\infty}^{+\infty} \left(\mathrm{T}.F^{k-1}(x)\right) \mathrm{B}(\gamma - x)\,\mathrm{d}x\right) \tag{A.14}$$

A.2 Derivation of the Super-Resolution Algorithm with Spatial Domain Restoration

In this section, we demonstrate the derivation of the super-resolution method (5.32) from the restoration method (A.14). Thus, we consider now a sequence of low-resolution images $G_i, 1 \leq i \leq K$ and we rewite (A.14) as:

$$F_i^k(\gamma) = F_i^{k-1}(\gamma) + \mathrm{T}_i^{-1}.\mathrm{S}\left(G_i(y) - \mathrm{S}^{-1} \int_{-\infty}^{+\infty} \left(\mathrm{T}_i.F_i^{k-1}(x)\right) \mathrm{B}_i(\gamma - x)\,\mathrm{d}x\right) \tag{A.15}$$

where F_i is the ith super-resolution image, G_i is the ith low-resolution image, T_i is the geometrical transformation from the F_i to G_i, T_i^{-1} is the inverse geometric transformation, S is the upsampling operator, S^{-1} is the downsampling operator, and B_i the PSF of ith low-resolution image.

For the construction of the mosaic M, we suppose:

$$M = \frac{1}{K} \sum_{j=1}^{K} F_j \tag{A.16}$$

Thus (A.15) becomes:

$$\begin{aligned}\frac{1}{K}\sum_{j=1}^{K}F_j^k(\gamma) &= \frac{1}{K}\sum_{j=1}^{K}\left(F_j^{k-1}(\gamma)+\mathrm{T}_j^{-1}.\mathrm{S}\left(G_j(y)-\mathrm{S}^{-1}\int_{-\infty}^{+\infty}\left(\mathrm{T}_j.F_j^{k-1}(x)\right)\mathrm{B}_j(\gamma-x)\,\mathrm{d}x\right)\right)\\ M^k(\gamma) &= M^{k-1}(\gamma)+\frac{1}{K}\sum_{j=1}^{K}\mathrm{T}_j^{-1}.\mathrm{S}\left(G_j(y)-\mathrm{S}^{-1}\int_{-\infty}^{+\infty}\left(\mathrm{T}_j.F_j^{k-1}(x)\right)\mathrm{B}_j(\gamma-x)\,\mathrm{d}x\right)\end{aligned} \tag{A.17}$$

Assuming that F_i is a cut-out of M, then $\mathrm{T}_iF_i=\mathrm{T}_iM$:

$$M^k(\gamma)=M^{k-1}(\gamma)+\frac{1}{K}\sum_{j=1}^{K}\mathrm{T}_j^{-1}.\mathrm{S}\left(G_j(y)-\mathrm{S}^{-1}\int_{-\infty}^{+\infty}\left(\mathrm{T}_j.M^{k-1}(x)\right)\mathrm{B}_j(\gamma-x)\,\mathrm{d}x\right) \tag{A.18}$$

A.3 Relationship between Convolution at Low and High Resolution

Our objective in this section is to establish the relationship between $\mathrm{S}^{-1}(\mathrm{B}*F)$ and $(\mathrm{S}^{-1}.\mathrm{B})*(\mathrm{S}^{-1}.F)$ where $*$ is the convolution operator and S^{-1} is the downsampling operator by the factor ς.

Considering the PSF B and the super-resolution image F, then the blurred super-resolution image $\tilde{F}$ is:

$$\tilde{F}(\gamma)=\int_{-\infty}^{+\infty}F(x)\mathrm{B}(\gamma-x)\,\mathrm{d}x \tag{A.19}$$

Denoting B_ς and F_ς as the subsamples of $\mathrm{B}(\gamma)$ and $F(\gamma)$ by the factor ς:

$$\mathrm{B}_\varsigma(\gamma) = \mathrm{B}(\varsigma\gamma) \tag{A.20}$$

$$F_\varsigma(\gamma) = F(\varsigma\gamma) \tag{A.21}$$

Denoting $\tilde{F}_\varsigma(\gamma)$ as the result of the convolution $(\mathrm{S}^{-1}\mathrm{B})*(\mathrm{S}^{-1}F)$:

$$\begin{aligned}\tilde{F}_\varsigma(\gamma) &= \int_{-\infty}^{+\infty}F_\varsigma(x)\mathrm{B}_\varsigma(\gamma-x)\,\mathrm{d}x\\ &= \int_{-\infty}^{+\infty}F(\varsigma x)\mathrm{B}(\varsigma(\gamma-x)\,\mathrm{d}x\\ &= \int_{-\infty}^{+\infty}F(\varsigma x)\mathrm{B}(\varsigma\gamma-\varsigma x))\,\mathrm{d}x\\ &= \frac{1}{\varsigma}\int_{-\infty}^{+\infty}F(\varsigma x)\mathrm{B}(\varsigma\gamma-\varsigma x)\,\mathrm{d}\varsigma x\end{aligned} \tag{A.22}$$

Replacing $y = \varsigma x$:

$$\begin{aligned} \tilde{F_\varsigma}(\gamma) &= \frac{1}{\varsigma} \int_{-\infty}^{+\infty} F(y) \mathrm{B}(\varsigma\gamma - y)) \, \mathrm{d}y \\ &= \frac{1}{\varsigma} \tilde{F}(\varsigma\gamma) \end{aligned} \tag{A.23}$$

Thus:

$$(\mathrm{S}^{-1}.\mathrm{B}) * (\mathrm{S}^{-1}.F) = \frac{1}{\varsigma} \mathrm{S}^{-1}(\mathrm{B} * F) \tag{A.24}$$

The constant $1/\varsigma$ corresponds to a normalisation factor and in case of a normalised convolution mask it can be neglected.

Appendix B

Other MPEG Video Compression Standards

We presented in Chapter 2 the popular MPEG-1 and MPEG-2 video compression standards and explained in this thesis how low-resolution data extracted from these streams can be used for mosaic construction. However, the MPEG-1/2 standards are progressively replaced by new compression standards such as H.264/MPEG-4 AVC which is specifically promising for high definition (HD) video broadcast. To this end, we outline in this chapter the succeeding MPEG standards, namely MPEG-4, H.264/MPEG-4 AVC and H.264/MPEG-4 SVC. As as these standards are built on the structure of MPEG-1/2 coding, they provide the video data in low-resolution according to their specific architecture and coding algorithms. Our objective is to show that this work can deal in certain circumstances with these video compression standards or can be extended to them.

The MPEG-4 standard [118], established in 2001, addressed a new generation of multimedia applications and services, e.g. interactive TV, internet video, and games, where access to coded audio and video objects might be needed. The MPEG-4 video coding standard, referred to as MPEG-4 part 2, provides a variety of tools enabling the efficient coding of video objects and their transmission on error prone networks. Further, MPEG-4 did clearly improve the coding efficiency over earlier standards, as it builds on the coding structure of MPEG-2, but adding enhanced and new tools within the same coding structure.

In the meantime, while highly interactive multimedia applications have appeared, there seemed to be an inexhaustive demand for higher compression with a better video quality. Hence, the H.264/MPEG-4 AVC standard [120] was established in 2003 by the *Joint Video Team* (JVT), a joint team of experts from the MPEG and the ITU-T *Video Coding Export Group* (VCEG). This standard is referred to as MPEG-4 part 10 by the MPEG, and H.264 by VCEG. Sometimes is also called *Advanced Video Coding* (AVC). H.264/MPEG-4 AVC is a new state-

of-the art video coding standard addressing applications such as internet multimedia, wireless video, video-on-demand, and videoconferencing, including HD video. It provides significantly higher compression than earlier standards.

Currently, *Scalable Video Coding* (SVC) is a very active research area aiming at the establishment of the scalable extension of H.264/MPEG-4 AVC. It is called H.264/MPEG-4 SVC and is a current standardisation project of the JVT. It is mainly designed for networking applications.

In the following, we present the developments in video compression and outline the main differences with respect to the MPEG-2 standard. Thus, Section B.1 presents the MPEG-4 standard, Section B.2 addresses the H.264/MPEG-4 AVC standard and Section B.3 its scalable extension.

B.1 MPEG-4

The MPEG-4 standard includes the coding of natural video and synthetic video which is referred to as MPEG-4 Visual. Neither MPEG-1 nor MPEG-2 considers synthetic video or computer graphics for coding. The MPEG-4 standard improves the MPEG-2 standard both in terms of compression efficiency and flexibility. This is achieved in two ways, by using a more advanced compression algorithm and providing a number of additional coding tools such as shape coding, texture coding, and sprite coding. The main features of MPEG-4 are the coding of objects and their scalability. Like MPEG-2 it defines profiles and levels. Figure B.1 illustrates the structure of the MPEG-4 encoder.

In the following, we outline the principal coding tools and point out some differences with respect to MPEG-1/2. More information about MPEG-4 video coding can be found in [152, 39].

B.1.1 Coding Structure

The coding structure of MPEG-4 is like in MPEG-1/2 a hierarchical structure, but it is modified in order to support object-based coding.

The highest syntactic structure is the *video object sequence* (VS). It contains 2D or 3D natural or synthetic objects and their enhancement layers. A *video object* (VO) corresponds to a particular 2D object in the scene. It is defined as an arbitrarily shaped region of the video scene existing during an arbitrary time interval. Each VO can be encoded in scalable or non-scalable form, represented by the *video object layer* (VOL). A VO can be encoded using spatial or temporal scalability, going from coarse to fine resolution. A VO is sampled in time and the VO at a specific time is called a *video object plane* (VOP). A VOP is coded by shape coding and texture coding, which is specified at macroblock and block layer. A collection of VOPs is called a *group of video object planes* (GOV). GOVs provide points in the bitstream where VOPs are encoded independently of each other, and can thus provide random access to the sequence.

In the special case of a single rectangular VO, all of the MPEG-4 layers can be related

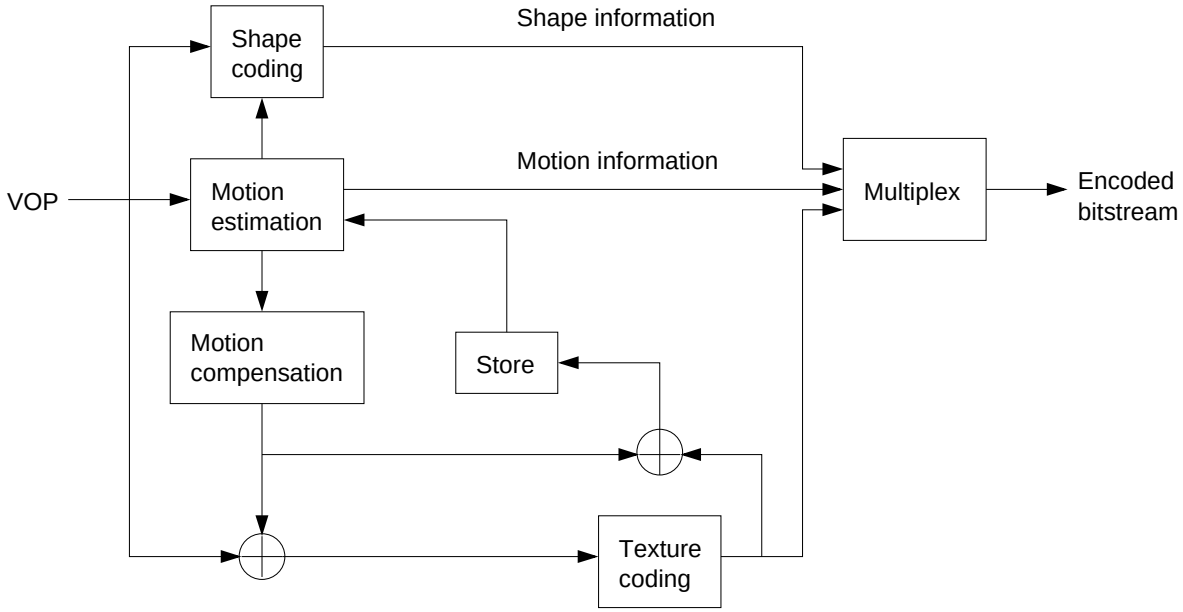


Figure B.1: The MPEG-4 encoder structure [172].

to MPEG-2 layers. The VS is the same as the VO since in this case a single VO is a video sequence, the VOL corresponds to the scalable extension, the GOV corresponds to the GOP, and the VOP corresponds to the frame. The definitions of I-VOP, P-VOP, and B-VOP correspond to the I-frame, P-frame and B-frame. The macroblock and the block are similarly to MPEG-1/2. In MPEG-4 the macroblock contains shape information, motion information, and texture information.

Nevertheless, object-based encoding has not really become reality and so the Simple and Advanced Simple Profile are very popular for rectangular frame coding. We showed above that if only one rectangular VO is encoded, the MPEG-4 layers can be related to MPEG-2 layer. Thus, in this PhD we could have worked on these profiles with a single rectangular VO. For more information on MPEG-4 profiles and levels, we refer the reader to [90, 152].

B.1.2 Shape Coding

Shape coding can be in binary mode, where the shape of each object is described by a binary mask, or in grey scale mode, where the shape is described by an alpha channel allowing transparency. The binary matrix representing the shape of a VOP is referred to as the binary mask. Its size is commonly the size of the bounding box of the VOP. In this mask every pixel belonging to the VOP is set to 255, the others are set to 0. The grey scale shape mask has a structure similar to that of the binary shape, except that each pixel can take a value in the range of 0 to 255 indicating the transparency of the pixel. In the scope of this PhD we are not supposed to work on shape coding, but on rectangular VOPs for the reason of compliance with MPEG-2. However, our mosaicing method segments frames into the background and

moving objects. Each of them is characterised by a binary mask. Thus, if the shapes are accurately encoded (the background and the moving objects), the segmentation process in the mosaicing method can be avoided and the encoded binary masks used instead after appropriate partial decoding.

B.1.3 Motion Estimation and Compensation

The motion estimation and compensation in MPEG-4 is block-based as in MPEG-1/2, but provides appropriate modifications for object boundaries. Motion estimation is only performed for macroblocks in the bounding box of the VOP in question. If a macroblock lies entirely inside of the VOP, motion estimation is performed in the usual way. The block size in the block matching can be 16×16 or 8×8. This results in one motion vector for the entire macroblock, and one for each of its blocks. For macroblocks that partially belong to the VOP, motion vectors are estimated using a modified block matching technique. The matching criterion is only computed on the pixels that belong to the VOP. In case the reference block lies on the VOP boundary, a repetitive padding technique assigns values to pixels outside of the VOP. The matching criterion is then computed using these padded pixels as well.

If *short header* mode (a tool to provide compatibility between MPEG-4 Visual and the H.263 video coding standard) is enabled, block matching is only performed at macroblock basis. Thus, MPEG-4 motion vectors can also be exploited by our method.

B.1.4 Texture Coding

In case of an intra macroblock, the texture information consists in the luminance and chrominance components. In case of a motion-compensated macroblock the texture information consists in the residual error remaining after motion compensation. The method for texture coding is similar to MPEG-2 based on a 8×8 block DCT. Macroblocks that are completely inside the VOP are encoded without modifications. For macroblocks on the boundary of the VOP, a specific padding process is used to extend the shape into a rectangular macroblock. Once the block has been padded is it coded similar to an internal block by a 8×8 block DCT. Hence, MPEG-4 follows in case of rectangular VOPs the same principle of block-based DCT and quantisation as in MPEG-1/2, but MPEG-4 allows for non-linear quantisation of DC values.

The entropy value of the quantised coefficients in intra blocks can further be reduced by a prediction from neighbouring blocks. In MPEG-1/2 only the DC coefficient of the current block is spatially predicted from the DC coefficient of the prior block. In MPEG-4 DC and AC coefficients are predicted either from the block to the left or the block above as illustrated in Figure B.2. The direction of the prediction is selected based on the comparison of the horizontal and vertical DC gradients of the surrounding blocks A,B, and C. For DC prediction (see Figure B.2(a)) the DC value of the current block (X) is predicted either from the DC coefficient of the block to the left (A) or the block above (C). For AC prediction (see Figure B.2(b)), either the coefficients from the first row, or the coefficients from the first column are predicted from the co-sited coefficients in the selected candidate block.

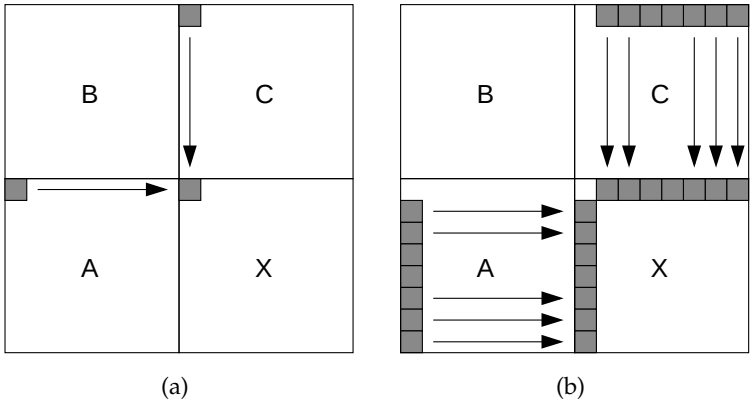


Figure B.2: Coefficient prediction in intra blocks [152]: (a) Prediction of DC coefficients, (b) prediction of AC coefficients.

This is followed by scanning. Three different scan methods exist, a zigzag scan and an alternate horizontal scan as in MPEG-2, and additionally an alternate vertical scan which is similar to the the horizontal scan, but applied in vertical direction. The type of DC prediction determines the type of scan that is performed. If there is no DC prediction, zigzag scan is applied, if there is DC prediction in horizontal direction, alternate vertical scan is applied, otherwise alternate horizontal is applied. Finally, run length coding and variable length coding is performed on the reordered transform coefficients.

As outlined above, MPEG-4 texture coding follows a coding scheme similar to that one of MPEG-1/2. Applying adequate partial decoding, DC coefficients can be extracted at 8×8 block basis for the use in our work.

B.1.5 Sprite Coding

An obvious example of sprite coding is a background sprite, also referred to as the background mosaic. A static sprite is generated before encoding using original VOPs. This sprite is then used to predict each of the source frames, and the residuals are spatially coded. The decoder receives the static sprite before the rest of the video segment. Then, the reconstructed VOPs can be generated easily by warping the sprite with appropriate parameters and adding the residual error. Sprite-based coding provides a high coding efficiency. It is very well suited for synthetic objects, but can also be used for objects in natural scenes that undergo rigid motion.

We can see that MPEG-4 foresees a mosaic-based coding, but these mosaics are constructed before encoding the frames. In our work, on the contrary, the mosaic is constructed at the decoder side.

B.1.6 Scalability

The scalability framework of MPEG-4 includes spatial, temporal and SNR scalability similar to MPEG-2. The major difference is that MPEG-4 extends the concepts of scalability to be

content based. Thus, scalability is provided on object basis with the restriction that the object shape has to be rectangular. By using the multiple VOP structure, different resolution enhancements can be applied to different parts of the video scene. Therefore, the enhancement layer may only be applied to a particular object or region of the base layer instead of the entire base layer. In addition, MPEG-4 supports fine grain scalability enabling the quality of the sequence to be increased in small steps.

B.2 H.264/MPEG-4 AVC

Unlike MPEG-2 and MPEG-4, the H.264/MPEG-4 AVC standard does not support layered scalable coding, and further unlike MPEG-4 it does not support object-based coding. The focus of the standard is on achieving a higher coding efficiency for progressive and interlaced video. Similar to MPEG-2 or MPEG-4 it includes the concepts of profiles and levels. This standard follows a similar coding structure as earlier standards but with many important enhancements including multiframe prediction and variable block size. Figure B.3 illustrates the structure of the H.264/MPEG-4 AVC encoder. Here, we only outline the advancements of the H.264/MPEG-4 AVC standard, additional information about this standard can be found in [147, 152].

B.2.1 Coding Structure

The basic coding structure of H.264/MPEG-4 AVC is similar to that of earlier standards. Coding is performed picture by picture. Each picture to be coded is partitioned into slices. In this standard, slices are individual coding units, while pictures can be considered as access units. There are three basic slice types: *I, P, and B-slices*. This is basically an extension of the I, P and B-frames concept of MPEG-1/2. A special type of frame containing only I-slices is called *instantaneous decoder refresh* (IDR) picture. It is defined such that any picture following an IDR picture does not use pictures prior to the IDR picture as references for motion-compensated prediction. P-slices consist of macroblocks that can be compressed by using motion-compensated prediction, but they can also contain intra macroblocks. Macroblocks of P-slices use only one prediction, but unlike previous standards, the pixels used as reference for motion compensation can either be in the past or in the future. Macroblocks of a B-slice when using motion-compensated prediction can use two predictions. Unlike earlier standards it is possible to have both predictions in the past or future. In addition, unlike earlier standards B-slices can also be used as reference for motion prediction by other slices.

B.2.2 Motion-Compensated Prediction

Motion-compensated prediction uses variable block sizes such as 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4. This is illustrated in Figure B.4. The top row illustrates the partitions of a 16×16 macroblock while the bottom row illustrates the partitions of an 8×8 block. In the previous standards, for prediction only the immediately previous/succeeding I or P-frame is used as a reference. H.264/MPEG-4 AVC extends the reference frame selection

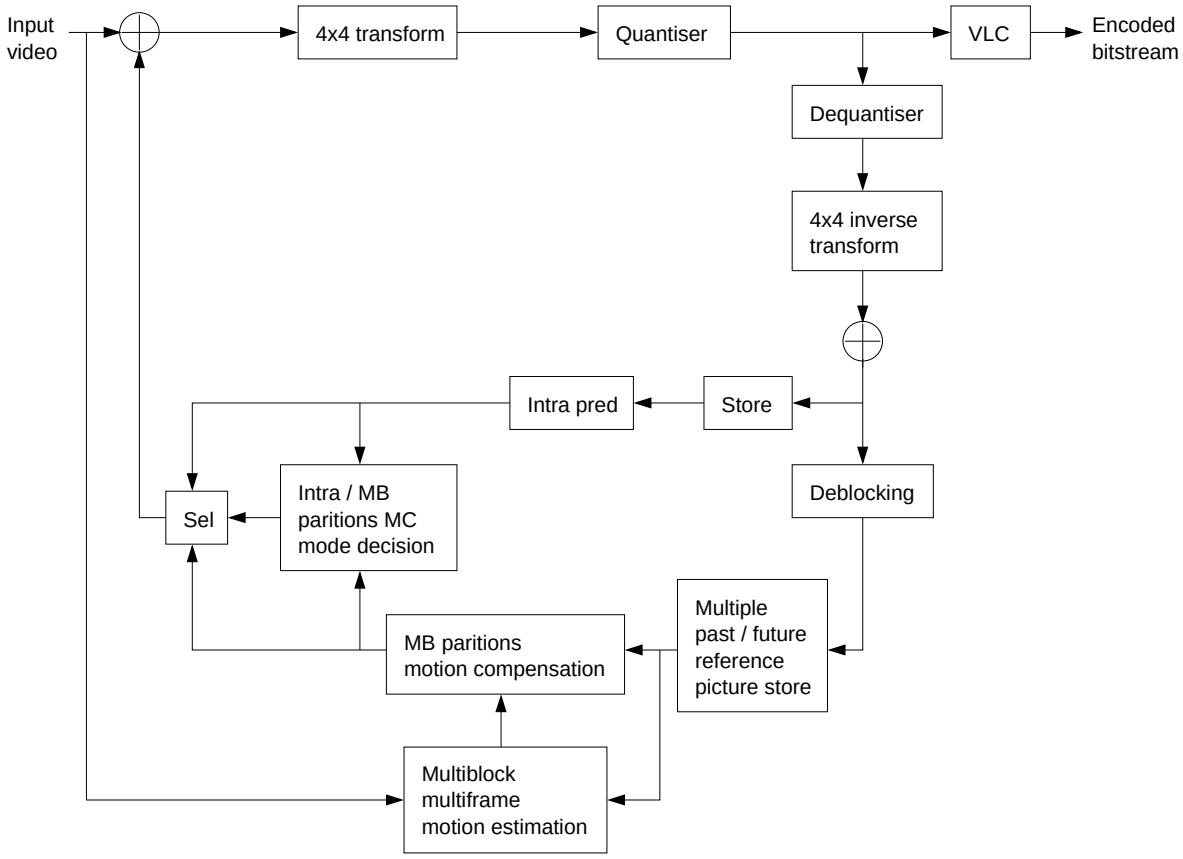


Figure B.3: The H.264/MPEG-4 AVC encoder structure [147].

by allowing the encoder to select among a larger number of frames that have been decoded and stored. The number of reference frames depends on the level of a profile.

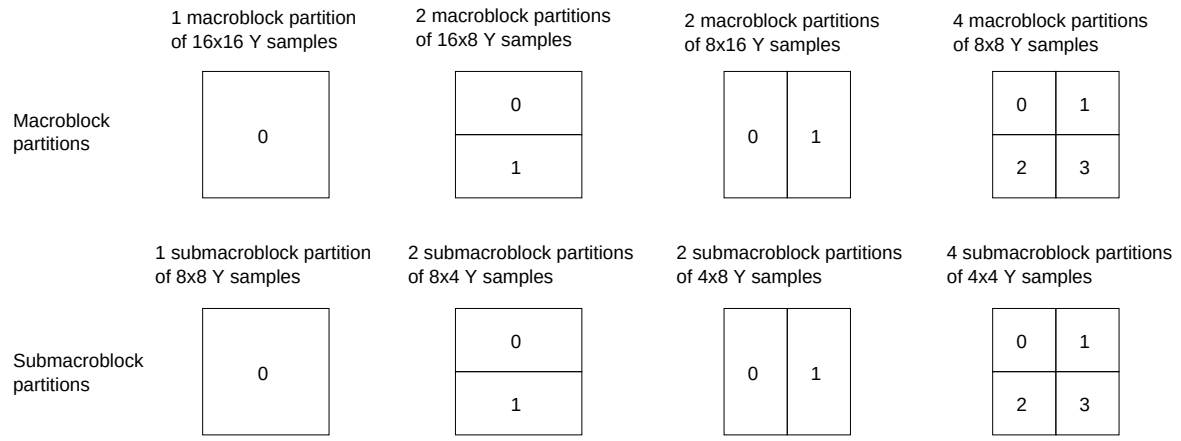


Figure B.4: Macroblock partitioning for motion compensation [147].

H.264/MPEG-4 AVC motion vectors can be used by our method in the very limited case, if motion prediction is restricted only to one reference picture at macroblock basis, if there is at least one picture containing only P-slices (P-frames) between two IDR pictures, and if P-frames are predicted with respect to the directly preceding P-frame or IDR picture. The latter is a necessary condition to obtain a continuous motion trajectory in the sequence (see Chapter 3).

B.2.3 Intra Prediction

Contrary to earlier standards where intra prediction is performed after transform coding, in H.264/MPEG-4 AVC it is performed in the pixel domain before transform coding. Figure B.5(a) shows a 4×4 block of luminance pixels $a, b, \ldots p$ to be predicted. The pixels $A, B, \ldots M$ have previously been encoded and reconstructed and are available to form a prediction reference. Each of the 16 pixels can be predicted using either DC mode or in one of the eight directions illustrated in Figure B.5(b). In case of DC prediction an average of 8 pixels, A, B, C, D, I, J, K, L, is used as prediction of each of the 16 pixels. As an alternative to the 4×4 luminance prediction, the entire 16×16 luminance component of a macroblock may be predicted in one operation. Three prediction directions are possible as illustrated in Figure B.6. The pixels used for prediction belong to the left hand or/and above decoded macroblock. The DC prediction uses an average of the neighbouring left pixel column (V) and the above pixel row (H). Each 8×8 chrominance component of an intra-coded macroblock is predicted from previously decoded chrominance samples above and/or to the left similar to the 16×16 luminance prediction modes described above.

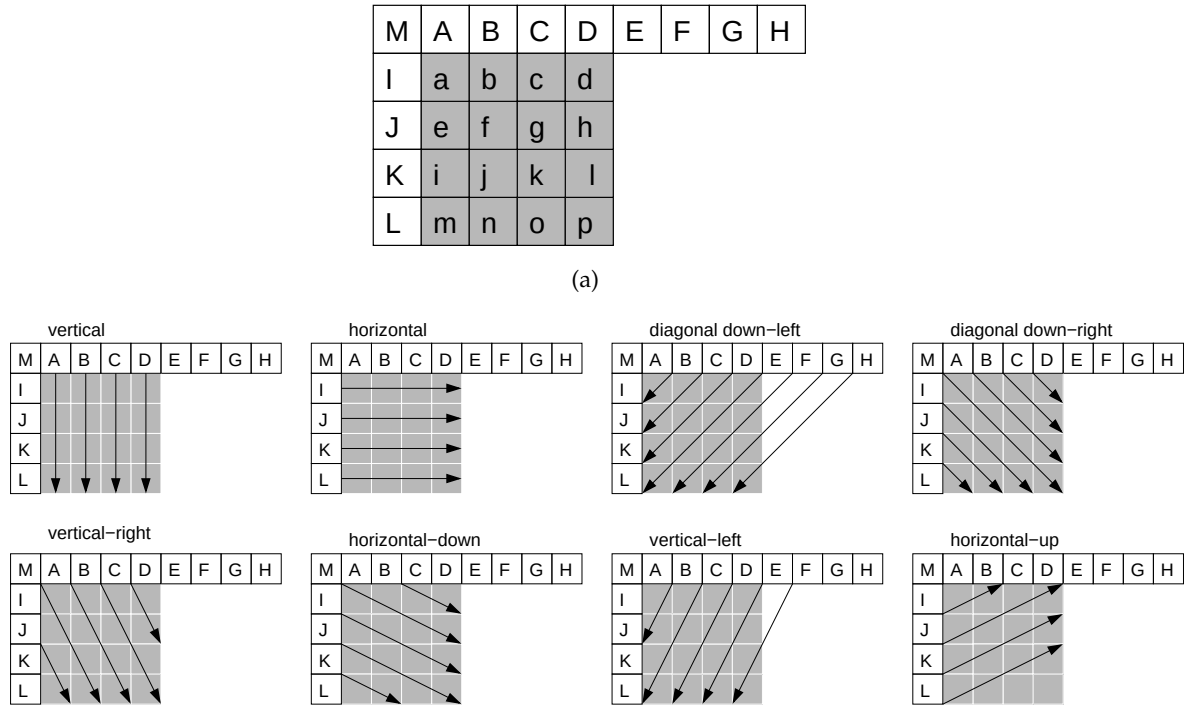


Figure B.5: 4×4 luminance prediction [152]: (a) Labelling of prediction samples, (b) prediction modes.

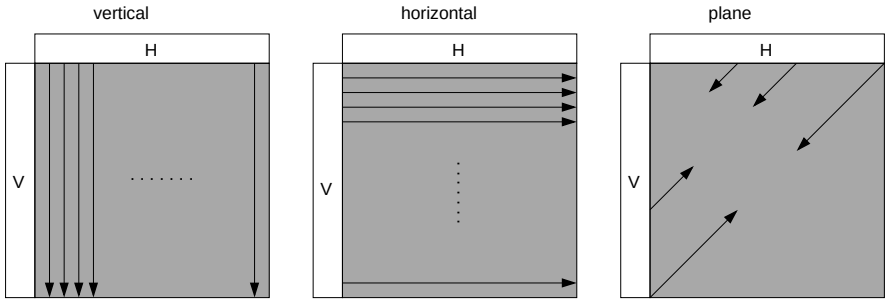


Figure B.6: Intra 16×16 prediction modes [152].

B.2.4 Transform Coding

Unlike earlier standards H.264/MPEG-4 AVC uses 4×4 transform block size instead of 8×8 block size. Further, it does not use the DCT transform but uses a simplified integer approximation: the high correlation transform (HCT). The 4×4 HCT whose transform coefficients are explicitly specified is perfectly invertible after quantisation contrary to the 8×8 floating point DCT transform of the previous standards. The core part of the transform can be implemented using only additions and shifts. A scaling multiplication (part of the transform) is integrated into the quantiser, reducing the total number of multiplications. In H.264/MPEG-4 AVC, the transform coding always uses pixel predictions i.e. the pixel values in a macroblock are always predicted, either from neighbouring pixels in the same frame (intra macroblocks) or from pixels in one or two previously decoded reference pictures (inter macroblocks), and then transform coded.

The extraction of DC coefficients in H.264/MPEG-4 AVC is more complex than for MPEG-1/2 as the DC value of a block is not directly available due to intra prediction before transform coding. Given that the HCT is a linear transform, the DC coefficient of the current block can be obtained by adding the value of its prediction.

B.2.5 Quantisation

Unlike MPEG-2 the current version of H.264/MPEG-4 AVC does not support quantisation matrices, it assumes a scalar quantiser. The transform coefficient $\mathrm{HCT}(u, v)$ is scaled and quantised in a single operation as:

$$\mathrm{HCT}_{\mathrm{Q}}(u, v) = \mathrm{round}\left(\mathrm{HCT}(u, v) \cdot \frac{PF}{S}\right) \tag{B.1}$$

where $\mathrm{HCT}_{\mathrm{Q}}(u, v)$ is the quantised transform coefficient, S the quantiser scale and PF a factor depending on the position (u, v). The latter describes the scaling operation of the HCT which is integrated into the quantiser.

B.2.6 Reordering

The zigzag scan of MPEG-2 works only well in the average and can be seen as a combination of three types of scans, a horizontal scan, a vertical scan an a diagonal scan. Often in natural images, on a block basis, a predominant preferred direction for scan exists depending on the orientations of significant coefficients. MPEG-4 uses adaptive scanning to exploit this property. However in H.264/MPEG-4 AVC fixed scanning similar to MPEG-2 is used: a 4×4 zigzag scan for frame macroblocks and 4×4 alternate scan for field macroblocks.

B.2.7 Entropy Coding

The H.264/MPEG-4 AVC standard includes two different entropy coding methods. Context adaptive binary arithmetic coding (CABAC) is a technique to losslessly compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context,

but is it not supported in all profiles. Context adaptive variable word length coding (CAVLC) is a lower-complexity alternative to CABAC for the coding of quantised transform coefficient values. Despite its lower complexity than CABAC, CAVLC is more elaborate and more efficient than the methods used to code coefficients in earlier standards.

B.2.8 Loop Filter

The loop filter, also called deblocking filter, operates on a macroblock after motion compensation and residual coding in case of inter coding, or after intra prediction and residual coding in case of intra coding. It operates on the edges of both macroblock and 4×4 sub-blocks. The loop filter operation is adaptive in response with several factors such as the quantisation parameter of the current and neighbouring macroblocks, the magnitude of the motion vector, and the macroblock coding type, as well as the values of the pixels to be filtered in both the current and neighboring macroblocks.

B.3 H.264/MPEG-4 SVC

In H.264/MPEG-4 SVC, most of the H.264/MPEG-4 AVC components such as motion-compensated and intra prediction, transform and entropy coding, and deblocking are used. The base layer of an H.264/MPEG-4 SVC bitstream is generally coded in compliance with H.264/MPEG-4 AVC. New tools are added for supporting spatial and SNR scalability. Thereby, the basic concepts of motion-compensated prediction and intra prediction are used as in H.264/MPEG-4 AVC. The redundancy between different layers is exploited by additional inter-layer prediction concepts. The H.264/MPEG-4 SVC encoder structure is illustrated in Figure B.7.

An important feature of H.264/MPEG-4 SVC is that scalability is provided at bitstream level. A bitstream with a reduced spatial and/or temporal resolution can be simply obtained by discarding network packets, the so-called NAL units, from the global H.264/MPEG-4 SVC bitstream that are not required for decoding the target resolution.

In the following, we outline the concepts for temporal, spatial and SNR scalability. For more information on the current H.264/MPEG-4 SVC development we refer the reader to [161, 162].

B.3.1 Temporal Scalability

Contrary to earlier standards, the coding and display order of frames is completely decoupled in H.264/MPEG-4 SVC. Any frame can be marked as reference frame and used for motion compensated prediction of following frames independent of the corresponding slice coding types. This allows the coding of frame sequences with arbitrary temporal dependencies. Thus, temporal scalable bitstreams can be generated by using hierarchical prediction structures as illustrated in Figure B.8 without any changes to H.264/MPEG-4 AVC. So-called key frames are coded in regular intervals by using only previous key frames as references. The frames between two key frames are then hierarchically predicted. The sequence of key

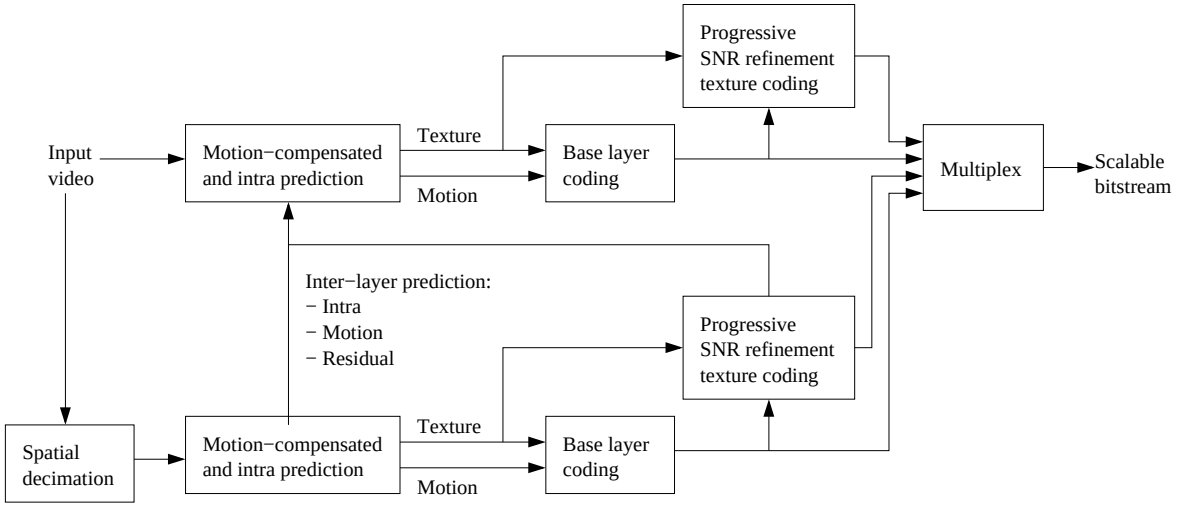


Figure B.7: The H.264/MPEG-4 SVC encoder structure with two spatial layers [162].

frame represents the coarsest supported temporal resolution, which can be refined by adding frames of following temporal prediction levels.

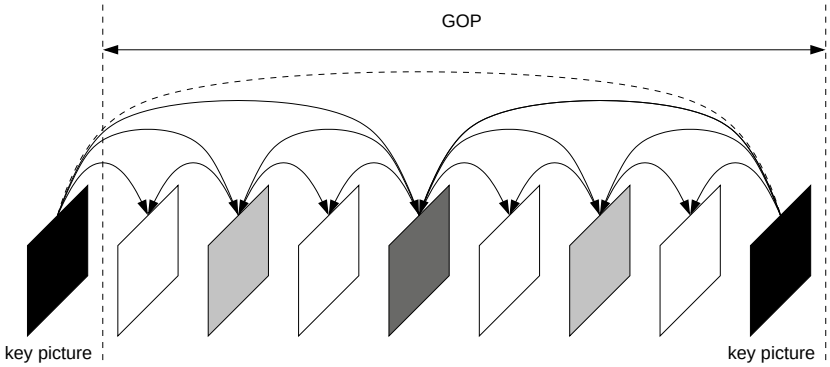


Figure B.8: Example of hierarchical prediction structure in H.264/MPEG-4 SVC [162].

In addition to enabling temporal scalability, the hierarchical prediction structures provide an improved coding efficiency compared to classical IBBP coding on the cost of an increased encoding-decoding delay.

B.3.2 Spatial Scalability

The frames of different spatial layers are independently coded with layer specific motion parameters for spatial scalability. In order to improve the coding efficiency of the enhancement layers, additional inter-layer predictions have been introduced: inter-layer motion prediction, inter-layer residual prediction, and inter-layer intra prediction.

The principle of inter-layer motion prediction is to use upsampled base layer motion information for coding the spatial enhancement layer. The reference frame indices are in-

herited from the lower layer, and the associated motion vectors are scaled and possibly refined. When inter-layer residual prediction is performed, the base layer residual macroblock is upsampled. Then, the enhancement layer residual is subtracted from the latter so that the difference is encoded. For inter-layer intra prediction, the reconstructed and upsampled macroblock of the lower layer is used as a prediction for the current macroblock, so that the difference is encoded.

B.3.3 SNR Scalability

For SNR scalability, coarse-grain scalability and fine-grain scalability are distinguished. Coarse-grain scalability is achieved using the concepts for spatial scalability. The only difference is that for CGS the upsampling operations of the inter-layer prediction mechanisms are omitted. Fine-grain scalability consists in truncating progressive refinement NAL units at any arbitrary byte-aligned point, so that the quality of the SNR base layer can be improved. Therefore, the coding order of transform coefficient levels is modified. Instead of scanning the transform coefficients macroblock by macroblock, the transform coefficient blocks are scanned in several paths.

Hence, in H.264/MPEG-4 SVC we found the same situation in the scalable bitstream as MPEG-2. The base layer motion vectors and DC coefficients can serve as data for our work, allowing for a fast video analysis and visualisation of content without complete decoding of the bitstream.

Bibliography

[1] The MathWorks – MATLAB and Simulink for technical computing. http://www.mathworks.com/.

[2] B. Adams, A. Amir, C. Dorai, S. Ghosal, G. Iyengar, A. Jaimes, C. Lang, C.-Y. Lin, A. Natsev, M. Naphade, C. Neti, H. H. Permuter H. J. Nock, R. Singh, J. R. Smith, S. Srinivasan, B. L. Tseng, T. V. Ashwin, and D. Zhang. IBM research TREC-2002 video retrieval system. In *TREC Video Retrieval Evaluation Online Proceedings, TRECVID'02,* http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html, 2002.

[3] K. Aizawa, T. Komatsu, and T. Saito. Acquisition of very high resolution images using stereo cameras. In *Proceedings of SPIE, Visual Communications and Image Processing*, volume 1605, pages 318–328, 1991.

[4] Y. Altunbasak and A. J. Patti. A fast method of reconstructing high-resolution panoramic stills from MPEG-compressed video. In *Proceedings of IEEE Second Workshop on Multimedia Signal Processing*, pages 99–104, 1998.

[5] Y. Altunbasak, A. J. Patti, and R. M. Mersereau. Super-resolution still and video reconstruction from MPEG coded video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4):217–227, 2002.

[6] E. Ardizzone, M. La Cascia, A. Avanzato, and A. Bruna. Video indexing using MPEG motion compensation vectors. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems, ICMCS'99*, volume 2, pages 725–729, 1999.

[7] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.

[8] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.

[9] M. Barlaud and C. Labit. *Compression et codage des images et des vidéos*. Hermès Sciecne, Lavoisier, 2002.

[10] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *Proceedings of European Conference on Computer Vision, ECCV'96*, volume 2, pages 573–582, 1996.

[11] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[12] J. Bescós. Real-time shot change detection over online MPEG-2 video. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):475–484, 2004.

[13] S. Borman and R. L. Stevenson. Spatial resolution enhancement of low-resolution image sequences – a comprehensive review with directions for future research. Technical report, University of Notre Dame, 1998.

[14] S. Borman and R. L. Stevenson. Super-resolution from image sequences - a review. In *Proceedings of the 1998 Midwest Symposium on Systems and Circuits, MWSCAS'98*, pages 374–378, 1998.

[15] S. Borman and R. L. Stevenson. Image sequence processing. In *Encyclopedia of Optical Engineering*, pages 840–879. Marcel Dekker, 2003.

[16] P. Bouthemy, M. Gelgon, and F. Ganansia. A unified approach to shot change detection and camera motion characterization. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1030–1044, October 1999.

[17] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[18] J. Ćalić and E. Izquierdo. Towards real-time shot detection in the MPEG-compressed domain. In *Proceedings of Workshop on Image Analysis for Multimedia Interactive Services WIAMIS'2001*, 2001.

[19] D. Calle. *Agrandissement d'images par synthèse de similarités et par induction sur un ensemble*. PhD thesis, Joseph Fourier University, France, 1999.

[20] X. Cao and P. N. Suganthan. Video shot motion characterization based on hierarchical overlapped growing neural gas networks. *Multimedia Systems*, 9(4):378–385, 2003.

[21] D. Capel. *Image Mosaicing and Super-resolution*. Distinguished Dissertations. Springer, 2003.

[22] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'98*, pages 885–891, 1998.

[23] D. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. In *Proceedings of International Conference on Pattern Recognition, ICPR'00*, volume 1, pages 600–605, 2000.

[24] D. Capel and A. Zisserman. Computer vision applied to super-resolution. *IEEE Signal Processing Magazine*, 20(3):75–86, 2003.

[25] A. Cavallaro and T. Ebrahimi. Classification of change detection algorithms for object-based applications. In *Proceedings of International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'03*, pages 129–136, 2003.

[26] S.-F. Chang and D. G. Messerschmitt. A new approach to decoding and compositing motion-compensated DCT based images. In *Proceedings of IEEE International Conference of Acoustics, Speech, Signal Processing, ICASSP'93*, volume 5, pages 421–424, 1993.

[27] J. Chanussot, M. Paindavoine, and P. Lambert. Real time vector median like filter fpga design and application to color image filtering. In *Proceedings of International Conference on Image Processing, ICIP'99*, volume 2, pages 414–418, 1999.

[28] D. Chen and R. R. Schultz. Extraction of high-resolution video stills from MPEG image sequences. In *Proceedings of IEEE International Conference on Image Processing, ICIP'98*, volume 2, pages 465–469, 1998.

[29] L. Y. Chen, S. C. Chan, and H. Y. Shum. A joint motion-image inpainting method for error concealment in video coding. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 2241–2244, 2006.

[30] M. Chiang and T. Boult. Local blur estimation and super-resolution. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 821–826, 1997.

[31] P. H. Van Cittert. Zum Einfluß der Spaltbreite auf die Intensitätsverteilung in Spektrallinien. II. *Zeitschrift für Physik*, 69:298–308, 1931.

[32] M. Coimbra and M. Davies. Approximating optical flow within the MPEG-2 compressed domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):103–107, 2005.

[33] M. Coimbra, M. Davies, and S. Velastin. Pedestrian detection using MPEG-2 motion vectors. In *Proceedings of International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'03*, pages 164–169, 2003.

[34] S. Daia, M. Yang, Y. Wu, and A. K. Katsaggelos. Tracking motion-blurred targets in video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 2389–2392, 2006.

[35] F. Dekeyser, P. Pérez, and P. Bouthemy. Restoration of noisy, blurred, undersampled image sequences using a parametric motion model. In *Proceedings of International Symposium on Image and Video Communications, ISIVC'00*, 2000.

[36] C. Doulaverakis, V. Vagionitis, M. Zervakis, and E. Petrakis. Adaptive methods for motion characterization and segmentation of MPEG compressed frame sequences. In *International Conference on Image Analysis and Recognition, ICIAR'04*, volume 1, pages 310–317, 2004.

[37] W. Dupuy, J. Benois-Pineau, and D. Barba. 1-D mosaics as a tool for structuring and navigation in digital video content. In *Proceedings of International Workshop on Visual Content Processing and Representation, VLBV'03*, pages 93–100, 2003.

[38] M. Durik and J. Benois-Pineau. Robust motion characterisation for video indexing based on MPEG2 optical flow. In *Proceedings of International Workshop on Content-Based Multimedia Indexing, CBMI'01*, pages 57–64, 2001.

[39] T. Ebrahimi and C. Horne. MPEG-4 natural video coding – an overview. *Signal Processing: Image Communication*, 15(4):365–385, 2000.

[40] M. Elad and Feuer A. Superresolution restoration of an image sequence: Adaptive filtering approach. *IEEE Transactions on Image Processing*, 8(3):387–395, 1999.

[41] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Transactions on Image Processing*, 6(12):1646–1658, 1997.

[42] M. Elad and A. Feuer. Super-resolution reconstruction of image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):817–834, 1999.

[43] P. E. Eren, M. I. Sezan, and A. M. Tekalp. Robust, object-based high resolution image reconstruction from low-resolution video. *IEEE Transactions on Image Processing*, 6(10):1446–1451, 1997.

[44] R. Ewerth and B. Freisleben. Improving cut detection in MPEG video by GOP-oriented frame difference normalization. In *Proceedings of International Conference on Pattern Recognition, ICPR'04*, volume 2, pages 807–810, 2004.

[45] R. Ewerth, M. Schwalb, P. Tessmann, and B. Freisleben. Estimation of arbitrary camera motion in MPEG videos. In *IEEE International Conference on Pattern Recognition, ICPR'04*, volume 1, pages 512–515, 2004.

[46] W. E. Farag and H. Abdel-Wahab. A new paradigm for detecting scene changes on MPEG compressed videos. In *Proceedings of International Symposium on Signal Processing and Information Technology*, pages 153–158, 2001.

[47] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in superresolution. *International Journal of Imaging Systems and Technology, Special Issue on High Resolution Image Reconstruction*, 14(2):47–57, 2004.

[48] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame superresolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, 2004.

[49] S. Fassino and A. Montanvert. Agrandissement des séquences vidéo et extension à un facteur d'agrandissement quelconque. In *Proceedings of Congrès francophone des jeunes chercheurs en vision par ordinateur, Orasis'03*, 2003. `http://orasis2003.loria.fr/actes/actes.php`.

[50] C. S. Fuh, S. W. Cho, and K. Essig. Hierarchical color image region segmentation for content-based image retrieval system. *IEEE Transactions on Image Processing*, 9(1):156–162, 2000.

[51] D. Gottlieb and C.-W. Shu. On the gibbs phenomenon and its resolution. *SIAM Review*, 39(4), 1997.

[52] M. Gràcia Pla. Correction and qualification of global motion in the rough indexing paradigm. Master's thesis, University of Bordeaux I, France and Technical University of Catalonia, Spain, 2006.

[53] H. Greenspan, G. Oz, N. Kiryati, and S. Peled. MRI inter-slice reconstruction using super resolution. *Magnetic Resonance Imaging*, 20:437–446, 2002.

[54] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Bayesian resolution-enhancement framework for transform-coded video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'01*, volume 2, pages 41–44, 2001.

[55] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. A multiframe blocking-artifact reduction for transform-coded video. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'01*, volume 3, pages 1789–1792, 2001.

[56] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Multiframe blocking-artifact reduction for transform-coded video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4):276–282, 2002.

[57] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Multiframe resolution-enhancement methods for compressed video. *IEEE Signal Processing Letters*, 9(6):170–174, 2002.

[58] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Super-resolution reconstruction of compressed video using transform-domain statistics. *IEEE Transactions on Image Processing*, 13(1):33–43, 2004.

[59] O. Hadar, I. Dror, and N. S. Kopeika. Image resolution limits resulting from mechanical vibrations, part iv: Real-time numerical calculation of optical transfer, functions and experimental verification. *Optical Engineering*, 33(2):566–578, 1994.

[60] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, 1986.

[61] Y. Haoran, D. Rajan, and L. Lang Tien. A unified approach to detection of shot boundaries and subshots in compressed video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'2003*, page 2003.

[62] S. Har-Noy and T. Nguyen. A deconvolution method for LCD motion blur reduction. In *Proceedings of IEE International Conference on Image Processing, ICIP'06*, pages 629–632, 2006.

[63] R. C. Hardie, K. J. Barnard, and E. E. Armstrong. Joint map registration and high resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6(12):1621–1633, 1997.

[64] R. C. Hardie, K. J. Barnard, J. G. Bognar, E. E. Armstrong, and E. A. Watson. High resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. *Optical Engineering*, 37(1):247–260, 1998.

[65] R. I. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997.

[66] D. Hasler, L. Sbaiz, S. Süsstrunk, and M. Vetterli. Outlier modeling in image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):301–315, 2003.

[67] D. Hasler and S. Süsstrunk. Mapping colour in image stitching applications. *Journal of Visual Communication and Image Representation*, 15(12):65–90, 2004.

[68] H. He and L. P. Kondi. A regularization framework for joint blur estimation and super-resolution of video sequences. In *Proceedings of IEEE International Conference on Image Processing, ICIP'05*, volume III, pages 329–332, 2005.

[69] Y. He, K.-H. Yap, L. Chen, and L.-P. Chau. Blind super-resolution image reconstruction using a maximum a posteriori estimation. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 1729–1732, 2006.

[70] M.-C. Hong, M. G. Kang, and A. K. Katsaggelos. An iterative weighted regularized algorithm for improving the resolution of video sequences. In *Proceedings of IEEE International Conference on Image Processing*, volume II, pages 474–477, 1997.

[71] S. Hsu and P. Anandan. Hierarchical representations for mosaic based video compression. In *Proceedings of Picture Coding Symposium*, pages 395–400, 1996.

[72] M. Irani and P. Anandan. Video indexing based on mosaic representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 86(5):905–921, 1998.

[73] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Mosaic representations of video sequences and their applications. *Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application*, 8(4), 1996.

[74] M. Irani and S. Peleg. Improving resolution by image registration. *Graphical Models and Image Processing*, 53:231–239, 1991.

[75] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993.

[76] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. 12(1):5–16, 1994.

[77] P. Jaillon and A. Montanvert. Image mosaicing applied to three-dimensional surfaces. In *IEEE International Conference on Pattern Recognition, ICRP'94*, pages 253–257, 1994.

[78] P. A. Jansson. Method for determining the response function of a high-resolution intrared spectrometer. *Journal of the Optical Society of America*, 60(2):184–191, 1970.

[79] P. A. Jansson. *Deconvolution of Images and Spectra*. Academic Press, 1997.

[80] W. Jiang and H. Schulzrinne. QoS measurement of internet real-time multimedia services. Technical Report CUCS-015-99, Department of Computer Science, Columbia University, 12 1999.

[81] A. K. Katsaggelos. Iterative image restoration algorithms. *Optical Engineering, special issue on Visual Communications and Image Processing*, 28(7):735–748, 1989.

[82] A. K. Katsaggelos and K. T. Lay. Maximum likelihood blur identification and image restoration using the em algorithm. *IEEE Transactions on Signal Processing*, 39:729–733, 1991.

[83] A. Kaup, K. Meisinger, and T. Aach. Frequency selective signal extrapolation with applications to error concealment in image communication. *International Journal of Electronics and Communication (AEÜ)*, 59:147–156, 6 2005.

[84] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using sub-pixel displacements. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–746, 1988.

[85] D.-W. Kim and K.-S. Hong. Enhanced mosaic blending using intrinsic camera parameters from a rotating and zooming camera. In *Proceedings of IEEE International Conference on Image Processing, ICIP'04*, volume 5, pages 3303–3306, 2004.

[86] J.-G. Kim, H. S. Chang, J. Kim, and H.-M. Kim. Threshold-based camera motion characterization of MPEG video. *ETRI Journal*, 26(3):269–272, 2004.

[87] S. P. Kim, N. K. Bose, and H. M. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics Speech and Signal Processing*, 38(6):1013–1027, 1990.

[88] S. P. Kim and W. Y. Su. Recursive high-resolution reconstruction of blurred multiframe images. *IEEE Transactions on Image Processing*, 2(4):534–539, 1993.

[89] V. Kobla, D. Doermann, K.-I. Lin, and C. Faloutsos. Compressed-domain video indexing techniques using DCT and motion vector information in MPEG video. In *Proceedings of SPIE, Storage and Retrieval for Image and Video Databases V*, volume 3022, pages 200–211, 1997.

[90] R. Koenen. Profiles and levels in MPEG-4: Approach and overview. *Signal Processing: Image Communication*, 15(4):463–478, 2000.

[91] A. C. Kokaram and S. J. Godsill. Joint detection, interpolation, motion and parameter estimation for image sequences with missing data. In *Proceedings of International Conference on Image Analysis and Processing, ICIAP '97*, volume 2, pages 719–726, 1997.

[92] T. Komatsu and T. Saito. Super-resolution sharpening-demosaicking method for removing image blurs caused by an optical low-pass filter. In *IEEE International Conference on Image Processing (ICIP)*, volume I, pages 845–848, 2005.

[93] N. S. Kopeika. *A System Engineering Approach to Imaging*. Prentice-Hall of India, 2003.

[94] P. Kornprobst, R. Peeters, T. Vieville, G. Malandain, S. Mierisova, S. Sunaert, O. Faugeras, and P. Van Hecke. Superresolution in MRI and its influence in statistical analysis. Technical report, INRIA – Sophia Antipolis, 2002.

[95] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *Proceedings of IEEE Workshop on Representations of Visual Scenes*, pages 10–17, 1995.

[96] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, 1996.

[97] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE Signal Processing Magazine*, 13(6):61–63, 1996.

[98] C. W. Lee and S. D. Kim. Multi-resolution mosaic construction using resolution map. In *Proceedings of International Workshop on Visual Content Processing and Representation, VLBV'03*, pages 180–187, 2003.

[99] S. Lefèvre, J. Holler, and N. Vincent. Real time temporal segmentation of compressed and uncompressed dynamic colour image sequences. In *Proceedings of International Workshop on Real Time Image Sequence Analysis*, pages 56–62, 2000.

[100] X. Li and M. T. Orchard. Novel sequential error-concealment techniques using orientation adaptive interpolation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(10):857–864, 10 2002.

[101] Z. Lin and H.-Y. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 26(1):83–97, 2004.

[102] G. Lorenz. Study and implementation of image restoration algorithms for error concealment. Research project, University of Bordeaux I, France and University of Koblenz, Germany, 2006.

[103] A. Lorette, H. Shekarforoush, and J. Zerubia. Super-resolution with adaptive regularization. In *Proceedings of International Conference on Image Processing, ICIP'97*, volume 1, pages 169–172, 1997.

[104] F. Manerba. *Efficient Identification in Image Sequences for Content Indexing*. PhD thesis, University of Bordeaux I, France and University of Brescia, Italie, 2005.

[105] F. Manerba, J. Benois-Pineau, and R. Leonardi. Extraction of foreground objects from a MPEG2 video stream in rough indexing framework. In *Proceedings of SPIE, Storage and Retrieval Methods and Applications for Multimedia*, volume 5307, pages 50–60, 2004.

[106] S. Mann and R. W. Picard. Virtual bellows: Constructing high quality stills from video. In *Proceedings of IEEE International Conference on Image Processing, ICIP'94*, volume 1, pages 363–367, 1994.

[107] B. Martins and S. Forchhammer. A unified approach to restoration, deinterlacing and resolution enhancement in decoding MPEG-2 video. *IEEE Transactions on Circuits Systems and Video Technology*, 12(9):803–811, 2002.

[108] J. Mateos, A. K. Katsaggelos, and R. Molina. Resolution enhancement of compressed low resolution video. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'00*, volume 4, pages 1919–1922, 2000.

[109] J. Mateos, A. K. Katsaggelos, and R. Molina. Simultaneous motion estimation and resolution enhancement of compressed low resolution video. In *Proceedings of IEEE Internation Conference on Image Processing, ICIP'00*, volume 2, pages 653–656, 2000.

[110] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143, 2001.

[111] K. Meisinger and A. Kaup. Minimizing a weighted error criterion for spatial error concealment of missing image data. In *Proceedings of IEEE International Conference on Image Processing, ICIP'04*, pages 813–816, 10 2004.

[112] R. Molina, A. K. Katsaggelos, and J. Mateos. Bayesian and regularization methods for hyperparameter estimation in image restoration. *IEEE Transactions on Image Processing*, 8(2):231–246, 1999.

[113] Motion2D. A software to estimate 2D parametric motion models. `http://www.irisa.fr/vista/Motion2D/`.

[114] ISO/IEC MPEG. Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s – part 2: Video, ISO/IEC 11172-2. International Standard, 1993.

[115] ISO/IEC MPEG. MPEG-2 test model 5. Document ISO/IEC JTC1/SC29 WG11/93-400, 1993.

[116] ISO/IEC MPEG. Information technology – generic coding of moving pictures and associated audio information: Systems, ISO/IEC 13818-1. International Standard, 1995.

[117] ISO/IEC MPEG. Information technology – generic coding of moving pictures and associated audio information: Video, ISO/IEC 13818-2. International Standard, 1995.

[118] ISO/IEC MPEG. Information technology – coding of audio-visual objects – part 2: Visual, ISO/IEC 14496-2. International Standard, 2001.

[119] ISO/IEC MPEG. Information technology – multimedia content description interface – part 5: Multimedia description schemes, ISO/IEC 15938-5. International Standard, 2003.

[120] ISO/IEC MPEG and ITU-T VCEG. Information technology – coding of audio-visual objects – part 10: Advanced video coding, ISO/IEC 14496-10. International Standard, 2003.

[121] MPEG Software Simulation Group (MSSG). `http://www.mpeg.org/MPEG/MSSG/`.

[122] J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing*, 19(4):1063–1082, 1998.

[123] C. W. Ngo, T. C. Pong, H. J. Zhang, and R. T. Chin. Motion characterization by temporal slices analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR'00*, volume 2, pages 768–773, 2000.

[124] N. Nguyen and P. Milanfar. A wavelet-based interpolation-restoration method for superresolution. *Circuits, Systems, and Signal Processing*, 19(4):321–338, 2000.

[125] N. Nguyen, P. Milanfar, and G. H. Golub. A computationally efficient image superresolution algorithm. *IEEE Transactions on Image Processing*, 10(4):573–583, 2001.

[126] N. Nguyen, P. Milanfar, and G. H. Golub. Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement. *IEEE Transactions on Image Processing*, 10(9):1299–1308, 2001.

[127] H. Nicolas. *Hiérarchie des modèles de mouvement et méthodes d'estimation associées. Application au codage des séquences d'images.* PhD thesis, University of Rennes I, France, 1992.

[128] H. Nicolas. *Contributions à la création et à la manipulation des objets vidéo*. IRISA + IFSIC, 2001.

[129] H. Nicolas. New methods for dynamic mosaicing. *IEEE Transactions on Image Processing*, 10(8):1239–1251, 2001.

[130] J. M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, 1995.

[131] University of Otago. Computer Vision Homepage. `http://www.cs.otago.ac.nz/research/vision/index.html`.

[132] S. C. Park, M. G. Kang, C. A. Segall, and A. K. Katsaggelos. Spatially adaptive high-resolution image reconstruction of dct-based compressed images. *IEEE Transactions on Image Processing*, 13(4):573–585, 2004.

[133] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003.

[134] A. J. Patti and Y. Altunbasak. Super-resolution image estimation for transform coded video with application to MPEG. In *Proceedings of IEEE International Conference on Image Processing, ICIP'99*, volume III, pages 179–183, 1999.

[135] A. J. Patti and Y. Altunbasak. Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants. *IEEE Transactions on Image Processing*, 10(1):179–186, 2001.

[136] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Robust methods for high quality stills from interlaced video in the presence of dominant motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):328–342, 1997.

[137] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Transactions on Image Processing*, 6(8):1064–1078, 1997.

[138] A. J. Patti, A. M. Tekalp, and M. I. Sezan. A new motion compensated reduced order model kalman filter for space-varying restoration of progressive and interlaced video. *IEEE Transactions on Image Processing*, 7(4):543–554, 1998.

[139] S. Peleg and J. Herman. Panoramic mosaicing with videobrush. In *Proceedings of DARPA Image Understanding Workshop*, pages 261–264, 1997.

[140] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR '97*, pages 338–343, 1997.

[141] S. Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Patter Recognition Letters*, 5(3):223–226, 1987.

[142] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1144–1154, 2000.

[143] M. Pilu. Using raw MPEG motion vectors to determine global camera motion. In *Proceedings of SPIE, Visual Communications and Image Processing*, volume 3309, pages 448–459, 1998.

[144] J.-M. Pinel. *Etude des conditions d'éclairement dans une sequence d'images et application à la composition et au codage de scenes video*. PhD thesis, University of Rennes I, France, 2002.

[145] W. K. Pratt. *Digital Image Processing*. Wiley, 1978.

[146] M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *European Conference on Computer Vision, ECCV'94*, volume 2, pages 295–304, 1994.

[147] A. Puri, X. Chen, and A. Luthra. Video coding using the H.264/MPEG-4 AVC compression standard. *Signal Processing: Image Communication*, 19(9):793–849, 2004.

[148] A. Rav-Acha and S. Peleg. Restoration of multiple images with motion blur in different directions. In *Proceedings of IEEE International Workshop Applications of Computer Vision, WACV'00*, pages 22–28, 2000.

[149] A. Rav-Acha and S. Peleg. Lucas-kanade without iterative warping. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 1097–1100, 2006.

[150] A. R. Reibman and T. Schaper. Subjective performance of super-resolution enhancement. In *Proceedings of International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM'06*, 2006.

[151] S. Rhee and M. G. Kang. Discrete cosine transform based regularized high-resolution image reconstruction algorithm. *Optical Engineering*, 38(8):1348–1356, 1999.

[152] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression*. Wiley, 2003.

[153] E. Saez, J. M. Palomares, J. I. Benavides, and N. Guil. Global motion estimation algorithm for video segmentation. In *Proceedings of SPIE, Visual Communications and Image Processing*, volume 5150, pages 1540–1550, 2003.

[154] H. Sanneck and G. Carle. A framework model for packet loss metrics based on loss runlengths. In *Proceedings of the SPIE, Multimedia Computing and Networking Conference*, volume 3969, pages 177–187, 2000.

[155] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. In *Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR '97*, pages 450–456, 1997.

[156] H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 18(8):814–830, 1996.

[157] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D&3D dominant motion estimation for mosaicing and video representation. In *Proceedings of IEEE International Conference on Computer Vision, ICCV'95*, pages 583–590, 1995.

[158] H. S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proceedings of European Conference on Computer Vision, ECCV '98*, volume 2, pages 103–119, 1998.

[159] R. R. Schultz and R. L. Stevenson. Improved definition video frame enhancement. In *Proceedings of IEEE International Conference of Acoustics, Speech, Signal Processing, ICASSP'95*, volume 4, pages 2169–2171, 1995.

[160] R. R. Schultz and R. L. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, 5(6):996–1011, 1996.

[161] H. Schwarz, D. Marpe, and T. Wiegand. Basic concepts for supporting spatial and snr scalability in the scalable H.264/MPEG4-AVC extension. In *Proceedings of IEEE International Workshop on Systems, Signals and Image Processing, IWSSIP'05*, 2005.

[162] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable H.264/MPEG4-AVC extension. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 161–164, 2006.

[163] C. Segall, R. Molina, A. K. Katsaggelos, and J. Mateos. Reconstruction of high-resolution image frames from a sequence of low-resolution and compressed observations. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'02*, volume 2, pages 1701–1704, 2002.

[164] C. A. Segall, R. Molina, and A. K. Katsaggelos. High-resolution images from low-resolution compressed video. *IEEE Signal Processing Magazine*, 20(3):37–48, 2003.

[165] C. A. Segall, R. Molina, A. K. Katsaggelos, and J. Mateos. Bayesian resolution enhancement of compressed video. *IEEE Transactions on Image Processing*, 13(7):898–911, 2004.

[166] M. I. Sezan. An overview of convex projections theory and its applications to image recovery problems. *Ultramicroscopy*, (40):55–67, 1992.

[167] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. In *Proceedings of European Conference on Computer Vision, ECCV'02*, volume 1, pages 753–768, 2002.

[168] E. Shechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, 2005.

[169] H. Shekarforoush and R. Chellappa. Data-driven multi-channel super-resolution with application to video sequences. *Journal of the Optical Society of America A*, 16(3):481–492, 1999.

[170] B. Shen, D. Li, and I. K. Sethi. HDH based compressed video cut detection. In *Proceedings of Visual'97*, 1997.

[171] K. Shen and E. J. Delp. A fast algorithm for video parsing using MPEG compressed sequences. In *Proceedings of IEEE International Conference on Image Processing, ICIP'95*, pages 2252–2255, 1995.

[172] Y. Q. Shi and H. Sun. *Image and Video Compression for Multimedia Engineering. Fundamentals, Algorithms, and Standards*. CRC Press, 1999.

[173] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Proceedings of International Conference on Computer Vision, ICCV'98*, pages 953–958, 1998.

[174] E. A. Silva, K. Panetta, and S. S. Agaian. Quantifying image similarity using measure of enhancement by entropy. In *Proceedings of SPIE, Mobile Multimedia/Image Processing for Military and Security Applications*, volume 6579, 2007.

[175] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval, MIR '06*, pages 321–330, 2006.

[176] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[177] J. Song and B. L. Yeo. Fast extraction of spatially reduced image sequences from MPEG-2 compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1100–1114, 1999.

[178] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *Journal of the Optical Society of America A*, 6(11):1715–1726, 1989.

[179] A. Stern, E. Kempner, A. Shukrun, and N. S. Kopeika. Restoration and resolution enhancement of a single image from a vibration-distorted image sequence. *Optical Engineering*, 39(9):2451–2457, 2000.

[180] A. Stern, Y. Porat, A. Ben-Dor, and N. S Kopeika. Enhanced-resolution image restoration from a sequence of low-frequency vibrated images by use of convex projections. *Applied Optics*, 40:4706–4715, 2001.

[181] M. Sugano, K. Hoaschi, K. Matsumoto, F. Sugaya, and Y. Nakajima. Shot boundary determination on MPEG compressed domain and story segmentation experiments for TRECVID'2003. In *TREC Video Retrieval Evaluation Online Proceedings, TRECVID'03,* `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`, 2003.

[182] J.-W. Suh and Y.-S. Ho. Error concealment techniques for digital TV. *IEEE Transactions on Broadcasting*, 48(4):299–306, 12 2002.

[183] H. Sun and W. Kwok. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Transactions on Image Processing*, 4(4):470–477, 4 1995.

[184] R. Szeliski. Image mosaicing for tele-reality applications. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 44–53, 1994.

[185] R. Szeliski. Video mosaics for virtual environment. *IEEE Computer Graphics and Applications*, 16(2):22–30, 1996.

[186] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31(Annual Conference Series):251–258, 1997.

[187] Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge. A new method for camera motion parameter estimation. In *Proceedings of IEEE International Conference on Image Processing, ICIP'95*, volume 1, pages 406–409, 1995.

[188] L. Teixeira and M. Martins. Video compression: The MPEG standards. In *Proceedings of European Conference on Multimedia Applications, Services and Techniques, ECMAST'96*, pages 615–634, 1996.

[189] A. M. Tekalp, M. K. Özkan, and M. I. Sezan. High-resolution image reconstruction from lower-resolution images sequences and space-varying image restoration. In *Proceedings of IEEE International Conference of Acoustics, Speech, Signal Processing, ICASSP'92*, volume 3, pages 169–172, 1992.

[190] B. C. Tom and A. K. Katsaggelos. Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images. In *Proceedings of the International Conference on Image Processing, ICIP'95*, volume 2, pages 2539–2542, 1995.

[191] B. C. Tom and A. K. Katsaggelos. Iterative algorithm for improving the resolution of video sequences. In *Proceedings of SPIE, Visual Communications and Image Processing*, volume 2727, pages 1430–1438, 1996.

[192] P. H. S. Torr and A. Zisserman. MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[193] H. J. Trussell and S. Fogel. Identification and restoration of spatially variant motion blurs in sequential images. *IEEE Transactions on Image Processing*, 1(1):123–126, 1992.

[194] R. Y. Tsai and T. S. Huang. *Advances in Computer Vision and Image Processing*, volume 1, chapter Multiframe image restoration and registration, pages 317–339. JAI Press Inc., 1984.

[195] S. Tsekeridou and I. Pitas. Error concealment techniques in MPEG-2. In *Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing, NSIP'97*, 1997.

[196] D. L. Tull and A. K. Katsaggelos. Iterative restoration of fast-moving objects in dynamic image sequences. *Optical Engineering*, 35(12):3460–3469, 1996.

[197] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. *Graphical Models and Image Processing*, 54(2):181–186, 1992.

[198] A. van Eekeren, K. Schutte, J. Dijk, D. J. J. de Lange, and L. J. van Vliet. Super-resolution on moving objects and background. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 2709–2712, 2006.

[199] A. Vega Garcìa. *Méchanismes de contrôle pour la transmission de l'audio sur l'Internet*. PhD thesis, Université de Nice-Sophia Antipolis, France, 1996.

[200] VXL. C++ Libraries for Computer Vision Research and Implementation. `http://vxl.sourceforge.net/`.

[201] Y. Wang and Q.-F. Zhu. Error control and concealment for video communications: A review. *Proceedings of the IEEE*, 86(5):974–997, 5 1998.

[202] Z. Wang and F. Qi. On ambiguities in super-resolution modeling. *IEEE Signal Processing Letters*, 11(8):678–681, 2004.

[203] E. W. Weisstein. Fourier transform–Gaussian. MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/FourierTransformGaussian.html`.

[204] L. Wu. *Segmentation spatio-temporelle d'images animées en vue d' un codage à fort taux de compression*. PhD thesis, University of Nantes, France, 1995.

[205] M. V. Wüst Zibetti and J. Mayer. Outlier robust and edge-preserving simultaneous super-resolution. In *IEEE International Conference on Image Processing, ICIP'06*, pages 1741–1744, 2006.

[206] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modelling of the temporal dependence in packet loss. Technical Report 98-78, Department of Computer Science, University of Massachusetts, 1998.

[207] Y. Yang and N. Galatsanos. Removal of compression artifacts using projections onto convex sets and line process modeling. *IEEE Transactions on Image Processing*, 6(10):1345–1357, 1997.

[208] B. L. Yeo and B. Liu. Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):533–544, 1995.

[209] Y. Yitzhaky and N. S. Kopeika. Identification of blur parameters from motion blurred images. *Graphical Models and Image Processing*, 59(5):310–320, 1997.

[210] D. C. Youla and H. Webb. Image restoration by the method of convex projections: Part 1 theory. *IEEE Transactions on Medical Imaging*, MI-1(2):81–94, 1982.

[211] J. Zhang, J. F. Arnold, and M. R. Frater. A cell-loss concealment technique for MPEG-2 coded video. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(4):659–665, 2000.

[212] W.-Y. Zhao. Super-resolution with significant illumination change. In *Proceedings of International Conference on Image Processing, ICIP'04*, pages 1771–1774, 2004.

[213] W.-Y. Zhao. Super-resolving compressed video with large artifacts. In *Proceedings of International Conference on Pattern Recognition, ICPR'04*, volume 1, pages 516–519, 2004.

[214] W. Y. Zhao and H. S. Sawhney. Is super-resolution with optical flow feasible? In *Proceedings of European Conference on Computer Vision, ECCV'02*, volume 1, pages 599–613, 2002.

[215] A. Zomet and S. Peleg. Efficient super-resolution and applications to mosaics. In *Proceedings of International Conference on Pattern Recognition, ICPR'00*, volume 1, pages 579–583, 2000.

[216] A. Zomet, A. Rav-Acha, and S. Peleg. Robust super resolution. In *Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR'01*, volume 1, pages 645–650, 2001.

Publications

Journal Papers

[1] P. Krämer, O. Hadar, J. Benois-Pineau, and J.-P. Domenger. *Super-resolution mosaicing from MPEG compressed video*. Signal Processing: Image Communication, 2007, Elsevier. To appear.

[2] P. Krämer, J. Benois-Pineau, and J.-P. Domenger. *Scene similarity measure for video content segmentation in the framework of rough indexing paradigm*. International Journal of Intelligent Systems, special issue on Intelligent Multimedia Retrieval, 21(7):765–783, July 2006, Wiley.

International Conferences with Proceedings

[3] P. Krämer, O. Hadar, J. Benois-Pineau, and J.-P. Domenger. *Use of motion information in super-resolution mosaicing*. In Proceedings of the International Conference on Image Processing (ICIP'06), pages 357–360, October 2006.

[4] P. Krämer, O. Hadar, J. Benois-Pineau, and J.-P. Domenger. *Super-resolution mosaicing from MPEG compressed video*. In Proceedings of the International Conference on Image Processing (ICIP'05), volume 1, pages 893–896, September 2005.

[5] P. Krämer, J. Benois-Pineau, and M. Gràcia Pla. *Indexing camera motion integrating knowledge of quality of the encoded video*. In Proceedings of the 1st International Conference on Semantic and Digital Media Technologies (SAMT'06), volume 233, `http://CEUR-WS.org/Vol-233/`, December 2006.

[6] P. Krämer, J. Benois-Pineau, and J.-P. Domenger. *Scene similarity measure for video content segmentation in the framework of rough indexing paradigm*. In Proceedings of the 2nd International Workshop on Adaptive Multimedia Retrieval (AMR'04), pages 141–155, August 2004.

[7] J. Ćalić, P. Krämer, U. Naci, S. Vrochidis, S. Aksoy, Q. Zhang, J. Benois-Pineau and A. Saracoglu, C. Doulaverakis, R. Jarina, N. Campbell, V. Merzaris, I. Kompatsiaris, E. Spyrou, G. Koumoulos, Y. Avrithis, A. Dalkilic, A. Alatan, A. Hanjalic, and E. Izquierdo *COST292 experimental framework for TRECVID*

2006. In TREC Video Retrieval Evaluation Online Proceedings (TRECVID'06), `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`, November 2006.

[8] P. Krämer and J. Benois-Pineau *Camera motion detection in the rough indexing paradigm*. In TREC Video Retrieval Evaluation Online Proceedings (TRECVID'05), `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`, November 2005.

[9] L. Primaux, J. Benois-Pineau, P. Krämer, and J.-P. Domenger. *Shot boundary detection in the framework of rough indexing paradigm*. In TREC Video Retrieval Evaluation Online Proceedings (TRECVID'04), `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`, November 2004.

International Conferences without Proceedings

[10] P. Krämer, O. Hadar, J. Benois-Pineau, and J.-P. Domenger. *Increasing resolution in MRI using motion-based restoration and super-resolution techniques*. The 1st France-Israel Bi-National Workshop on NanoBioPhotonics (NanoBio-Photonics'05), December 2005.

Technical Reports

[11] P. Krämer, J. Benois-Pineau, and J.-P. Domenger. *Construction of spatio-temporal mosaics in the framework of rough indexing paradigm*. Technical report 1328-04, LaBRI, University of Bordeaux 1, `http://www.labri.fr/perso/lepine/Rapports_internes/`, June 2004.