Multi-Scale Superpatch Matching using Dual Superpixel Descriptors

Rémi Giraud[a], Merlin Boyer[a], Michaël Clément[b,*]

[a]*Bordeaux INP, Univ. Bordeaux, CNRS, IMS, UMR 5218, F-33400 Talence, France*
[b]*Bordeaux INP, Univ. Bordeaux, CNRS, LaBRI, UMR 5800, F-33400 Talence, France*

Abstract

Over-segmentation into superpixels is a very effective dimensionality reduction strategy, enabling fast dense image processing. The main issue of this approach is the inherent irregularity of the image decomposition compared to standard hierarchical multi-resolution schemes, especially when searching for similar neighboring patterns. Several works have attempted to overcome this issue by taking into account the region irregularity into their comparison model. Nevertheless, they remain sub-optimal to provide robust and accurate superpixel neighborhood descriptors, since they only compute features within each region, poorly capturing contour information at superpixel borders. In this work, we address these limitations by introducing the dual superpatch, a novel superpixel neighborhood descriptor. This structure contains features computed in reduced superpixel regions, as well as at the interfaces of multiple superpixels to explicitly capture contour structure information. A fast multi-scale non-local matching framework is also introduced for the search of similar descriptors at different resolution levels in an image dataset. The proposed dual superpatch enables to more accurately capture similar structured patterns at different scales, and we demonstrate the robustness and performance of this new strategy on matching and supervised labeling applications.

1. Introduction

In many computer vision related applications, such as image classification and segmentation, there is a important need for fully automated results. To this end, a commonly employed strategy is to take inspiration from other data, in a supervised way when ground truth annotations are available. In this context, non-local methods have provided accurate results on various applications. In these methods, image regions are independently considered, generally using a square patch defined for each pixel, capturing a local pattern [4]. For exemplar-based classification or segmentation, matching algorithms are usually used to search for similar patterns in the available data, to then transfer the associated information, at the pixel or image scale after a global decision process.

This search for similar patterns is generally performed for each image patch, and fast matching algorithms have been developed *e.g.*, PatchMatch [3], TreeCANN [24] or FLANN [22], to efficiently exploit a large number of example images in a reduced computational time. To find similar content with such matching algorithms, it is common to extract features from dense image patches or local interest points. These features are usually designed to be robust to transformations such as scaling, viewpoint or illumination changes. Standard descriptors include features based on gradient information such as SIFT [20] or HoG [7, 8], or binary patterns, *e.g.*, BRIEF [5].

In the last few years, convolutional neural networks have also been able to extract particularly relevant features, and have shown promising results in many applications related to image processing [18]. However, although some recent architectures such as U-Net may learn from relatively small datasets [26], these methods rely on costly supervised learning strategies, and often require very large annotated datasets to be trained efficiently. Besides, the results of these models are generally difficult to interpret, and can be very sensitive to small perturbations of their inputs [21]. In many cases, such as medical imaging, these drawbacks can limit the potential of automated processing pipelines. Therefore, there is still an important need for methods that can perform without learning steps, as well as with limited training data and computing power.

In this context of fast image search and matching requirements, many works have first focused on hierarchical approaches using prior image over-segmentation into regular grids, *e.g.*, [17]. To go further, other methods proposed to group similar pixels into connected components of homogeneous colors called superpixels, drastically reducing the number of elements to process while preserving contours and spatial structure [28]. Therefore, a process applied at such over-segmentation scale can be close to the optimal pixel-wise result. Several works have used superpixels in non-local frameworks, *e.g.*, [12, 29], or in unsupervised learning-based superpixel matching approaches using random forests [6, 16]. Nevertheless, the geometrical irregularity of such decompo-

---

*Corresponding author: Tel.: +33(0)540003552;
Email address: `michael.clement@labri.fr` (Michaël Clément)

sitions [11] (*i.e.*, in terms of shape, adjacency or contour smoothness) can become an issue, since neighborhood information is crucial to compute accurate matches in terms of context.

Other approaches have attempted to use superpixel neighboring information, *e.g.*, [25, 27]. Among them, the Super-PatchMatch (SPM) framework [10] partially addresses this issue with a superpixel neighborhood structure called superpatch and a dedicated metric to compare two structures having different geometry and number of elements. However, SPM remains sub-optimal in terms of computational complexity and matching accuracy. The framework enables the search for matches at the same superpatch scale and it only computes features within each superpixel region, poorly capturing contour information. Several works indeed highlighted the need for accurate superpixel-wise features [23, 31, 30], while most image descriptors are locally computed on a regular square neighborhood.

Contributions

In this paper, we address the limitations of previous non-local methods only focusing on intra-region information within superpixels or superpixel neighborhoods [10], by introducing a novel dual superpixel neighborhood descriptor called dual superpatch (DSP), containing two independent descriptor sets (see Sec. 3).

First, intra-superpixel features capture color or texture information within cropped superpixel regions in order to avoid influence of pixel contours or inaccurate superpixel borders (see Sec. 3.1). Then, to capture structure information, for instance in terms of contour orientations, we extract a relatively regular grid of specific descriptors at superpixel interfaces (see Sec. 3.2). To efficiently compare such irregular dual descriptors, having different geometry and number of elements, we also propose new distances and optimizations, significantly reducing the computational complexity.

The SuperPatchMatch (SPM) search algorithm [10] with our accurate dual superpatch (DSPM), performs more relevant superpixel matching. To go further, we also extend DSPM to the search of matches at multiple scales (see Sec. 4), and propose a framework to perform automatic labeling using exemplar-based images with ground truth labels. In our framework, the comparison of DSP at different scales can be easily performed since we consider reduced spatial information, *i.e.*, sets of barycenter positions. This way, we are able to match similar objects at different sizes in heterogeneous datasets.

Finally, to show the robustness of our framework, especially compared to [10], we consider several matching and exemplar-based labeling experiments on a standard face dataset [14] (see Sec. 5).

2. The SuperPatchMatch Framework

In this section, we first recall the SuperPatchMatch (SPM) framework initially introduced by [10], which constitutes the

Figure 1: Superpatch definition. For a superpixel S_i (filled blue), neighboring superpixels (blue contours) having their barycenters (red dot) into a radius r, centered on X_{S_i} the barycenter of S_i, are part of the superpatch $\mathbf{S_i}$.

basis of our approach.

2.1. The SuperPatch Structure

To generalize standard patch-based frameworks to irregular image decompositions, [10] proposed the *superpatch* structure. As for square patches of pixels defined around each pixel, a superpatch $\mathbf{S_i}$, associated to a superpixel S_i, contains the adjacent neighbors of a superpixel S_i with respect to a fixed radius r. The proximity is simply computed according to the superpixel spatial barycenters such that:

$$\mathbf{S_i} = \{S_{i'}, \text{ such that } \|X_{S_i} - X_{S_{i'}}\|_2 \leq r\}, \tag{1}$$

where $X_{S_i} = [x_{S_i}, y_{S_i}]$ and $X_{S_{i'}} = [x_{S_{i'}}, y_{S_{i'}}]$ respectively denote the spatial barycenters of superpixels S_i and $S_{i'}$. This way, the superpatch structure only includes the most significantly neighboring superpixels, using reduced spatial information. In Fig. 1, we show a superpatch example, defined for a superpixel S_i, containing its adjacent superpixels to provide a superpixel neighborhood descriptor.

2.2. SuperPatch Comparison Distance

A comparison distance is also proposed in [10] to measure the similarity between two superpatches. The main issue to design such distance is that the two structures are very likely to have different geometry and number of elements. Hence, there is no one-to-one association between superpixels, contrary to pixels within regular patches. To preserve the ability to compare patterns, the spatiality must be taken into account, and [10] proposed to simply consider the proximity of superpixel barycenters after registration on the central superpixels. In the following, we consider two superpatches $\mathbf{S_i}$ and $\mathbf{S_j}$, for instance in two images A and B. A weight $w(X_{S_{i'}}, X_{S_{j'}}) = \exp(-\|X_{S_{j'}} - (X_{S_i} - X_{S_j}))\|_2^2/\sigma_1^2)$, measures the relative displacement between the registered barycenters $X_{S_{i'}}$ and $X_{S_{j'}}$ of superpixels $S_{i'} \in \mathbf{S_i}$ and $S_{j'} \in \mathbf{S_j}$, with respect to central superpixels S_i and S_j, and σ_1 is a scaling parameter set to $1/2 \sqrt{|I|/K}$, with $|I|$ and K the respective number of pixels and superpixels in the image. This way a superpixel $S_{i'}$ only compares to the closest ones in $\mathbf{S_j}$.

2

Figure 2: Comparison process of two superpatches $\mathbf{S_i}$ and $\mathbf{S_j}$ from Eq. (2). Superpatches are registered according to the barycenters X_{S_i} and X_{S_j} of their central superpixel. The weights w in Eq. (2) favor the comparison to closest superpixels after registration and the weight values corresponding to the bottom superpixel of $\mathbf{S_i}$ (filled blue) are represented within each compared superpixel of $\mathbf{S_j}$ (darker meaning a higher w weight).

The distance D between two superpatches $\mathbf{S_i}$ and $\mathbf{S_j}$ is finally defined as:

$$D(\mathbf{S_i},\mathbf{S_j}) = \frac{\sum\limits_{S_{i'} \in \mathbf{S_i}} \sum\limits_{S_{j'} \in \mathbf{S_j}} w(X_{S_{i'}}, X_{S_{j'}}) w_s(X_{S_{i'}}) w_s(X_{S_{j'}}) d(F_{S_{i'}}, F_{S_{j'}})}{\sum\limits_{S_{i'} \in \mathbf{S_i}} \sum\limits_{S_{j'} \in \mathbf{S_j}} w(X_{S_{i'}}, X_{S_{j'}}) w_s(X_{S_{i'}}) w_s(X_{S_{j'}})},$$

(2)

where $w_s(X_{S_{i'}})$ also weights the influence of $S_{i'}$ according to its spatial distance to S_i such that $w_s(X_{S_{i'}}) = \exp(-\|X_{S_{i'}} - X_{S_i}\|_2^2/(2r^2))$, and d is the distance between the superpixel features $F_{S_{i'}}$ and $F_{S_{j'}}$. Note that any distance d and feature F can be considered in Eq. (2).

The comparison process between two superpatches having different number of element and geometry is illustrated in Fig. 2. The weight w in Eq. (2) weights the feature distance d between a superpixel in $\mathbf{S_i}$ and a superpixel in $\mathbf{S_j}$ after registration on their central barycenters. In this Figure, the weights w corresponding to the bottom superpixel of $\mathbf{S_i}$ are represented within each superpixel of $\mathbf{S_j}$.

2.3. SuperPatchMatch Correspondence Algorithm

Non-local methods have soon highlighted the need for fast patch-based matching algorithms to perform the search of correspondences within large areas, *e.g.*, library of example images. A significant breakthrough has been obtained with PatchMatch (PM) [3], a fast partly random matching algorithm, providing for each patch of an image A, a match in an image B. PM has very interesting properties such as no requirements for learning or preprocessing steps, and its complexity only depends on the size of the image to process A, enabling to search for matches in a large set of example images.

The PM algorithm starts from random associations and iteratively refines them with a sequential processing of all image patches. The refinement process is mainly based on the fast propagation of good matches from spatially adjacent neighbors. Large regions are indeed very likely to correspond between images. According to the scan order, which is reversed at each iteration, the shifted correspondences of two

Figure 3: Core of the SPM algorithm. Full lines indicate current best matches. The direct adjacent neighbors of the blue superpixel are considered to propose new match candidates (dotted lines). The relative orientations between superpixels in A tend to be respected in B, *e.g.*, the yellow superpixel remains on top of the blue one in both images. Remaining adjacent neighbors (gray) that have not yet being processed during this iteration will be considered at the next one, processing superpixels in the reverse order.

spatially adjacent patches are considered as new candidates. Also, random patches are tested near the current best match in B. Note that the PM process being partly random, several processes carried out in parallel can potentially provide different matches.

The SuperPatchMatch (SPM) method generalizes PM to superpatches [10], to provide a fast matching algorithm of superpixels. The method mainly requires to adapt the propagation of matches based on adjacent neighbors since there is no more consistent geometry between adjacent superpixels, contrary to the standard pixel grid case. The propagation step of SPM is illustrated in Fig. 3 in the case of two images A and B. The adjacent neighbors are considered to lead to new matches while respecting the relative orientation between superpixels in A and B, to favor the matching of larger regions.

Limitations

Nevertheless, the default SPM distance in Eq. (2) has a quadratic complexity, *i.e.*, each superpixel of a superpatch is compared to all the ones of the other superpatch, and may result in important computational time. Besides, this framework only considers intra-superpixel descriptors. Although, they may be sufficient to capture information within regions, the superpatch does not explicitly focus on capturing contours or gradient information. This may be an issue since such information generally lies at the border of a superpixel, and can be shared between two regions, reducing the relevance of the descriptor. Finally, the SPM framework does not consider multi-scale information that would allow to capture objects of different sizes.

We address all these issues in the following sections with the proposed multi-scale superpatch matching framework that uses new dual superpixel descriptors.

3. Dual Superpixel Descriptors

In this section, we propose an approach to relevantly extract superpixel neighborhood descriptors. We introduce a dual descriptor to efficiently capture around a superpixel, both feature region content and contour information, potentially

3

Figure 4: Illustration of the dual superpatch (DSP) descriptor. Intra-region information ($R_{i'}$) within each superpixel $S_{i'}$ with an offset of $\beta = 3$ pixels from the border (blue regions) are considered along with information at the superpixel interfaces $I_{i'}$ (green squares) within the same r radius.

lying at their borders. Features are computed within super-pixel regions and additional contour features are extracted at multiple interfaces between adjacent superpixels. The proposed dual descriptor of superpixel neighborhood is called *Dual SuperPatch* (DSP), is denoted $\bar{\mathbf{S}}_{\mathbf{i}}$ for a superpixel S_i, and is represented in Fig. 4 on the same decomposition example used in Fig. 1. In the following, we present the extraction approach for region (R), *i.e.*, intra-superpixel, and interfaces (I) descriptors, and we propose a general framework to compare different DSP.

3.1. Superpixel Region Descriptors

The superpatch formulation of [10] considers features computed on each whole superpixel region contained into the neighborhood structure. However, superpixels tend to capture homogeneous regions, so pixels at thin contours can be arbitrarily associated to superpixels, leading to altered descriptors. A reduced block area or spatial weighting from the superpixel barycenter could not be applied since these approaches do not guarantee to relevantly extract information when superpixels have very irregular shapes [23].

In this work, we propose to consider the superpixel region information with an offset of β pixels to its borders. This way, we take into account almost all the region, while being robust to inaccurate superpixel borders or contour information, that will be considered in another specific interface descriptors within our DSP (see Sec. 3.2). In Fig. 4, the considered regions $R_{i'}$ for superpixels $S_{i'}$ are represented in blue. To each region $R_{i'}$, feature $F_{R_{i'}}$ and spatial $X_{R_{i'}}$ information are considered, so a dual superpatch contains a set of tuples $\mathbf{R}_{\mathbf{i}} = \{F_{R_{i'}}, X_{R_{i'}}\}$.

To demonstrate the issue of considering the whole superpixel region to extract features, we consider two decompositions of images containing regions with 16 different oriented textures (see Fig. 5). The superpatch radius is set to $r = 0$ to only consider intra-region information, where HoG descriptors [7] are computed. For each superpixel of the left image, we compute in an exhaustive manner its closest match in terms of superpixel content in the right image. If the texture

Figure 5: Two synthetic images containing 16 oriented textures and decomposed into superpixels with [9].

Table 1: Impact of border offset β for intra-region descriptor extraction. Texture matching results between images in Fig. 5 for different values of β and Gaussian noise levels using both superpixel decompositions from [9] and ground truth ones. Best and second results are respectively bold and underlined.

	Superpixel decompositions				Ground truth decompositions			
Noise variance	0	50	100	125	0	50	100	125
$\beta = 0$	0.526	0.634	0.516	0.366	**1.000**	**1.000**	0.980	0.907
$\beta = 1$	0.616	0.654	**0.558**	0.419	**1.000**	**1.000**	**0.993**	**0.913**
$\beta = 2$	<u>0.679</u>	<u>0.665</u>	<u>0.558</u>	**0.482**	**1.000**	**1.000**	<u>0.987</u>	<u>0.913</u>
$\beta = 3$	**0.711**	**0.675**	0.521	<u>0.482</u>	**1.000**	**1.000**	0.953	0.900

are similar, we consider the matching as accurate (1), otherwise inaccurate (0). We report in Tab. 1 the average matching accuracy on all superpixels of the left image, according to different β values and several levels of Gaussian noise applied to both images after decomposition. We perform this evaluation using the decompositions obtained with a texture-aware method [9], and also the ground truth ones, perfectly capturing texture changes. This experiment highlights the need to restrict the area to extract superpixel information since superpixel decompositions may be not perfectly accurate. Moreover, even on perfectly fitting decompositions, the inaccurate gradient information lying on superpixel borders is captured with $\beta = 0$ and may impact the results with a noise variance superior to 50, while we do not take into account texture in other superpixels with $\beta > 0$.

Fast comparison distance. The comparison between two sets of superpixel region descriptors can be performed in a more computationally efficient manner than with Eq. (2). We propose to only select one superpixel $S_{j'} \in \mathbf{S_j}$ to compare for each superpixel $S_{i'} \in \mathbf{S_i}$. To do so, the superpixel barycenter $X_{S_{i'}}$ is first registered by the displacement between central superpixels S_i and S_j, and we denote this new position by $X_{S_{j(i')}}$ which is computed such that $X_{S_{j(i')}} = X_{S_{i'}} - (X_{S_i} - X_{S_j})$. In Fig. 2, these correspond to the red superpixel barycenters. Then, we project the registered barycenters on the decomposition of the image from where $\mathbf{S_j}$ is extracted. In Fig. 2, the black superpixel containing a red dot $X_{S_{j(i')}}$ would be selected to compare to superpixel $S_{i'}$ of region $R_{i'}$. This corresponding superpixel containing $X_{S_{j(i')}}$ in the compared image, is denoted $S^j(i')$, and its associated intra-region is denoted $R^j(i')$. This way, we significantly reduce the distance complexity, while potentially increasing the comparison accuracy (see Sec. 5). The comparison between two region descriptors

(a) Matching using only color information at intra-superpixel regions ($\alpha = 1$)

(b) Matching using only contour features at superpixel interfaces ($\alpha = 0$)

Figure 6: Impact of the α parameter in the DSP distance (5). (a) Only region descriptors $R_{i'}$ are used ($\alpha = 1$), with average color information. This result corresponds to the one obtained using [10]. (b) Only interface HoG [7] descriptors $I_{i'}$ are used ($\alpha = 0$) and enable to capture structure information. Radius r is set to 25, *i.e.*, approximately capturing the first adjacency ring.

$\mathbf{R_i}$ and $\mathbf{R_j}$ is defined using barycenter projections such that:

$$d_p(\mathbf{R_i}, \mathbf{R_j}) = \frac{\sum\limits_{R_{i'} \in \mathbf{R_i}} w_s\left(X_{R_{i'}}\right) d\left(F_{R_{i'}}, F_{R^j(i')}\right)}{\sum\limits_{R_{i'} \in \mathbf{R_i}} w_s\left(X_{R_{i'}}\right)}. \tag{3}$$

Note that barycenters falling outside the image limits are projected to select the closest superpixel on the image boundary.

A similar projected distance was suggested in [10], in a non-symmetric formulation. In our dual superpatch comparison model, we consider a symmetric projected distance D_p on intra-region descriptors defined as:

$$D_p(\mathbf{R_i}, \mathbf{R_j}) = \frac{1}{2}\left(d_p\left(\mathbf{R_i}, \mathbf{R_j}\right) + d_p\left(\mathbf{R_j}, \mathbf{R_i}\right)\right). \tag{4}$$

3.2. Superpixel Interface Descriptors

To efficiently capture image contour information, we propose to also consider specific descriptors at superpixel interfaces. These can be easily extracted with a low complexity by considering the presence of at least three superpixels in a 3×3 pixels neighborhood. To avoid over detection, a larger area can be neglected after selection of an interface point. This way, we directly obtain a relatively regular grid of potential interest points in terms of contours, without introducing further scaling or thresholding parameters. In Fig. 4, these interface regions denoted $I_{i'}$ are represented as green squares. On these regions, specific contour descriptors can be computed, *e.g.*, HoG [7].

Acceleration of quadratic distance. Since interface regions do not provide a dense decomposition of the image domain, the distance Eq. (3) using projections cannot be used to fastly compare two sets of interface descriptors. A quadratic one-to-many distance such as Eq. (2) could be used, but at the expense of important computational cost. To address this issue, we propose a one-to-one association for each interface descriptor $I_{i'}$. Each $I_{i'}$ is only compared to the spatially closest one $I_{j'}$ in the other dual superpatch. This way, the framework only requires exhaustive spatial distances between interface barycenters. The distance is computed as for Eq. (3), where $I^j(i')$, the selected interface descriptor in $\mathbf{I_j}$ for $I_{i'}$ is defined as $I^j(i') = \underset{I_{j'}}{argmin}\|X_{I_{i'}} - X_{I_{j'}}\|_2$. Finally, as for Eq. (4), the distance is also computed from $\mathbf{I_j}$ to $\mathbf{I_i}$ to obtain a symmetric distance.

3.3. General Dual SuperPatch Comparison Framework

Our dual superpatch (DSP) $\bar{\mathbf{S}}_\mathbf{i}$, for a superpixel S_i, is described by a set of intra-superpixel regions $\mathbf{R_i} = \{F_{R_{i'}}, X_{R_{i'}}\}$ and superpixel interfaces $\mathbf{I_i} = \{F_{I_{i'}}, X_{I_{i'}}\}$ descriptors such that $\bar{\mathbf{S}}_\mathbf{i} = [\mathbf{R_i}, \mathbf{I_i}]$. Note that $\mathbf{R_i}$ and $\mathbf{I_i}$ can have a different number of elements. To relevantly measure the similarity of two DSP $\bar{\mathbf{S}}_\mathbf{i}$ and $\bar{\mathbf{S}}_\mathbf{j}$ having different geometry and number of elements, we propose the following general DSP comparison distance:

$$D(\bar{\mathbf{S}}_\mathbf{i}, \bar{\mathbf{S}}_\mathbf{j}) = \alpha\, D_p(\mathbf{R_i}, \mathbf{R_j}) + (1 - \alpha)\, D_p(\mathbf{I_i}, \mathbf{I_j}), \tag{5}$$

with D_p the fast distance on descriptors, using barycenter projections Eq. (4) for intra-region R, and selection of closest descriptor for interfaces I. and $\alpha \in [0, 1]$ a setting parameter. Note that any feature can be considered in $F_{R_{i'}}$ or $F_{I_{i'}}$. Therefore, α can have an intuitive tuning using the same or normalized descriptors for both R and I.

In Fig. 6, we show matching results obtained with our generalized model using average RGB color as intra-region features $F_{R_{i'}}$ (Fig. 6(a), $\alpha = 1$) and HoG descriptors as interface features $F_{I_{i'}}$ (Fig. 6(b), $\alpha = 0$). Hence, we highlight the general aspect of our framework that allows to either focus on intra-region (Fig. 6(a)) or interface information (Fig. 6(b)). In Sec. 5, we further demonstrate the performances obtained using these complementary descriptors.

4. Multi-Scale Dual SuperPatchMatch

In this section, we propose to extend the SPM framework with our dual descriptor (DSPM), to perform the search of DSP at multiple scales. We first show how to compare two DSP of different sizes, then we propose a multi-scale fusion strategy.

4.1. Dual SuperPatch Rescale

In Sec. 3, we showed how to compare two DSP extracted with the same radius size r. Nevertheless, the proposed distance Eq. (5) can easily adapt to DSP of different sizes, since the spatial information is only measured by barycenter positions denoted with X. We consider two DSP $\bar{\mathbf{S}}_\mathbf{i}$ and $\bar{\mathbf{S}}_\mathbf{j}$, with different DSP extraction radius r^i and r^j in Eq. (1). To compare them, all spatial information contained in $\bar{\mathbf{S}}_\mathbf{j}$ can be adjusted according to the ratio between the radiuses, such that:

$$\bar{\mathbf{S}}_\mathbf{j} = [\mathbf{R_j}, \mathbf{I_j}] = \left[\left\{\left(F_{R_{j'}}, X_{R_{j'}}\frac{r^i}{r^j}\right)\right\}, \left\{\left(F_{I_{j'}}, X_{I_{j'}}\frac{r^i}{r^j}\right)\right\}\right]. \tag{6}$$

5

Figure 7: Example of matching result without (gray lines) and with (green lines) a multi-scale search for matches. See text for more details.

Figure 8: Influence of radius r (a) and trade-off coefficient α (b) in the proposed DSPM framework compared to SPM. We report the average distance between each superpixel barycenter and the one of its closest match in another decomposition of the same image. See text for more details.

This way, similar DSP can be searched at various scales, *e.g.*, in example images. Note that features $F_{R_{j'}}$ and $F_{I_{j'}}$ remain unchanged by this scaling transformation.

4.2. Multi-Scale Exemplar-based Framework

Most non-local methods perform a search for similar content in a heterogeneous dataset, with no prior information on the targeted object size. In [10], no multi-scale strategy is proposed since it considers an exemplar-based labeling experiment on linearly registered images [14].

Here, we introduce a generalized multi-scale exemplar-based framework allowing to search for similar DSP of different radiuses, in order to capture objects of different sizes. The proposed DSP structure indeed enables to perform a simple automatic rescale, presented in Sec. 4.1. Hence, multiple DSP sizes can be considered in a set of examples images **B**. A set $\mathbf{r^B} = \{r^B\}$ is considered for setting the DSP radius in **B**.

In [10], a supervised labeling framework based on the non-local means algorithm [4] was introduced to merge the information of multiple superpatch matches computed in a library of example images for an image A to process. A label map $L_m^{r^B}(S_i)$ is computed for a superpixel S_i, for all M different labels m, such that:

$$L_m^{r^B}(S_i) = \frac{1}{W} \sum_{S_j \in \mathbf{B}_i^{m,r^B}} \omega(S_i, S_j), \qquad (7)$$

where $\mathbf{B}_i^{m,r^B}$ is the set of matches for S_i computed at scale r^B and having a ground truth label m, and W is the normalization factor $W = \sum_{m=1}^{M} \sum_{S_j \in \mathbf{B}_i^{m,r^B}} \omega(S_i, S_j)$, with ω a weight depending on the DSP similarity [10]. The final label of a superpixel S_i is computed as $\mathcal{L}(S_i) = \underset{m \in \{1,...,M\}}{\mathrm{argmax}} \left(\underset{r^B \in \mathbf{r^B}}{\mathrm{argmax}} \left(L_m^{r^B}(S_i) \right) \right)$.

In Fig. 7, we represent matching results obtained without (gray lines) and with (green lines) our multi-scale strategy. We can see that the best match between searches at scales $\mathbf{r^B} = [0.5, 1, 2, 4] \times r^A$ enables to catch the larger flower with similar colors, instead of the one at the same scale.

5. Experimental Validations

In this section, we present several quantitative experiments to demonstrate the interest of the proposed DSPM framework. We first validate the behavior of our model on standard images with respect to the method parameters. Then, we propose larger scale segmentation experiments on a standard face dataset.

5.1. Parameter Settings

The proposed method was implemented with MATLAB using C-MEX code on a standard Linux computer with 4 cores at 1.90 GHz and 16 GB of RAM. The number of DSPM iterations is set to 5, and we use a ℓ_2-norm as distance d between features F as in [10]. To avoid over detection, interfaces are detected at least each 4 pixels. Default parameters are set such that $\beta = 1$, the border offset for region features R, $\alpha = 0.5$, the trade-off parameter between intra-region and interface distances in Eq. (5), and superpatch radius $r = 50$. The considered descriptors in R and I are reported according to the application.

5.2. Influence of Parameters

To demonstrate the interest of each contribution, we consider a matching experiment on standard images *Baboon, Barbara, House, Lena* and *Peppers*, each decomposed with two superpixel methods: SLIC [1] and SNIC [2]. For each superpixel in a given decomposition, we compute the closest DSP match in the other one. A robust descriptor should indeed be robust to variations in the segmentations. In terms of features, we compute normalized cumulative RGB histogram with 9 bins per canal on intra-regions $F_{R_{j'}}$, and HoG [7] on a local 9×9 pixels window for interface region descriptors $F_{I_{j'}}$.

We evaluate the matching accuracy by the average distance between the superpixel barycenters and the one of their match in the other decomposition. In Fig. 8(a) and Fig. 8(b), we respectively report the average distance with respect to the radius parameter r and with respect to the α parameter for $r = 50$. On the first hand, Fig. 8(a) shows that the accuracy logically increases with the superpatch radius, and with each contribution, *i.e.*, symmetrical projected distance (Eq. (4)), offset to border ($\beta > 0$), and interface descriptors. On the other hand, Fig. 8(b) illustrates the interest of using interface descriptors in Eq. (5) in conjunction with cropped regions for $\alpha > 0.2$, and that a balanced trade-off parameter $\alpha \approx 0.5$ provides the best matching accuracy. In Fig. 9, we also show an example of matching result for DSPM with default parameters compared to SPM.

5.3. Segmentation and Labeling Experiments

Validation framework. We evaluate the proposed DSPM approach on the same face labeling experiment than [10]. The

(a) SLIC	(b) SNIC	(c) SPM	(d) DSPM

Figure 9: Matching accuracy of DSPM vs SPM. (a) and (b): decompositions with SLIC and SNIC. (c) and (d): SPM and DSPM matching results (r=50). Distances between superpixel barycenters are shown with standard optical flow representation (stronger colors represent larger displacements).

Table 2: Labeling accuracy for the multi-scale experiment. Training images have been either downsampled or upsampled by a factor of 1.5 or 2 and r^A is set to 50. The argmax columns correspond to the fusion strategy proposed in Sec. 4.2 for different combined scales.

Radius r^B	50*	25*	33*	75*+	100*+	argmax*	argmax+
w/o (6)	94.08%	94.05%	93.93%	94.11%	94.07%	94.25%	94.16%
w/ (6)	94.08%	94.22%	94.13%	94.87%	94.80%	94.78%	**94.96**%

considered Labeled Faces in the Wild (LFW) dataset [14], contains 1500 training and 927 testing images of 250×250 pixels, linearly registered with [13], and already decomposed into approximately 250 superpixels. LFW contains decompositions and labeling ground truths, so comparisons with state-of-the-art methods do not depend on the superpixel decompositions. Note that to fairly compare with [10], we use the same HoG implementation on a regular grid [8] and compute 50 DSP matches by 50 independent DSPM processes for each superpixel, merged in Eq. (7).

Multi-scale validation. In this experiment, the goal is to validate the interest of our proposed multi-scale matching strategy introduced in Sec. 4. To this end, we have manually applied random scaling transformations to the registered training images. Each image and its corresponding decomposition has been either downsampled or upsampled randomly by a factor of 1.5 or 2 (with no interpolation). As a result, faces depicted in the images can appear up to twice as big or small compared to their initial scales. Hence, the dataset contains face patterns at different scales that would not be efficiently captured using the same DSP radius in A and B.

We apply DSPM with r^A=50 and $r^B \in \{25, 33, 50, 75, 100\}$. In Tab. 2, we report the labeling accuracy for each radius and for the multi-scale label fusion proposed in Sec. 4.2. From these results, when $r^A \neq r^B$, we can observe that the performance for smaller or larger radiuses is always better after applying the rescale strategy, *i.e.*, with Eq. (6). The multi-scale fusion averaging the result over the different radius sizes performs better than the default $r^B = 50$ and than most single scales. Moreover, by considering only larger scales, *i.e.*, $r^B = 75, 100$, thus more precise DSP comparisons, we obtain the highest labeling accuracy, demonstrating the framework ability of the framework to merge information from multi-scale matching. In future works, we plan to apply this ap-

Figure 10: Influence of the number of selected k-ANN for SPM [10] and DSPM on face labeling accuracy.

Table 3: Exemplar-based labeling accuracy on the LFW dataset.

Method	Superpixel accuracy	Pixel accuracy
Spatial CRF [15]	93.95%	*not reported*
CRBM [15]	94.10%	*not reported*
GLOC [15]	94.95%	*not reported*
DCNN [19]	*not reported*	95.24%
SPM [10] w/ (2)	91.88%	92.21%
SPM [10] w/ (3)	95.08%	95.43%
DSPM	**95.24**%	**95.59**%

proach on non-registered datasets where objects might naturally appear at different scales.

Comparison to state-of-the-art methods. In Fig. 10, we first compare the influence of the number k of selected ANN for each superpixel, then merged in the label fusion process Eq. (7) for the proposed DSPM method and SPM [10] using Eq. (3). For all ANN numbers, DSPM provides the best results, and is already reaching 95.13% of accuracy with only $k = 20$ ANN. Note that for both methods, a plateau is reached around $k = 50$ ANN.

In Tab. 3, we also compare the performance of the proposed DSPM method with the results of state-of-the-art ones, mostly based on supervised (deep) learning approaches. In [15], several approaches are used to label the LFW dataset such as a spatial conditional random field (CRF) and a conditionnal restricted Boltzmann machine (CRBM). The GLOC (GLObal and LOCal) method [15] is also proposed to jointly use both CRF and CRNM approaches to introduce global shape priors in the training process. Finally, in [19], a deep convolutional neural network (DCNN) is proposed and dedicated to the face labeling application. Note that the results in Table 3 are the ones reported by the authors. For SPM, the results correspond to the initial framework using costly quadratic comparisons with Eq. (2), and the results reported by the authors using non-symmetric projected distances with Eq. (3). DSPM reports the best compared labeling accuracy at both superpixel and pixel-wise level. Labeling examples compared to SPM are also represented in Fig. 11. The proposed DSP enables to relevantly capture the context of a superpixel neighborhood in terms of texture and struc-

Figure 11: Examples of labeling results with superpixel accuracy obtained with the proposed DSPM approach on the LFW dataset, compared to the initial SPM method.

ture. Moreover, without further optimizations on non fully multi-threaded code, DSPM performs in less than 3s per subject, against 45s for SPM using Eq. (2). Note that compared state-of-the-art approaches may provide faster computational times but at the expense of previous costly learning-based processes.

Our method is particularly interesting due to its simplicity of use, parameter tuning, and interpretability compared to learning-based approaches, while providing more accurate results than SPM. Besides, any feature can be directly used in the method, even more advanced descriptors, *e.g.*, [31, 30], eventually based on previously trained deep learning architectures.

6. Conclusion

In this work, we addressed some important limitations of existing superpixel matching frameworks, in terms of robustness and computational complexity. We introduced the dual superpatch, a new superpixel neighborhood descriptor containing both intra-region and interface information that are respectively robust to the inaccuracy of superpixel borders and capture contour structures. We also proposed optimized distances and a multi-scale framework to search for similar dual superpatches in an image dataset. Our validations showed an accuracy improvement with our method on matching and exemplar-based labeling applications. The relevance of the proposed dual approach should benefit to all superpixel-based non-local approaches and future works will focus on applying the method to heterogeneous computer vision and medical datasets, and tackling other applications such as superpixel-based image editing.

References

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. on Pattern Analysis and Machine Intelligence 34, 2274–2282.

[2] Achanta, R., Süsstrunk, S., 2017. Superpixels and polygons using simple non-iterative clustering, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 4895–4904.

[3] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B., 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Trans. on Graphics 28.

[4] Buades, A., Coll, B., Morel, J.M., 2005. A non-local algorithm for image denoising, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 60–65.

[5] Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. BRIEF: Binary robust independent elementary features, in: European Conf. on Computer Vision (ECCV), pp. 778–792.

[6] Conze, P.H., Tilquin, F., Noblet, V., Rousseau, F., Heitz, F., Pessaux, P., 2017. Hierarchical multi-scale supervoxel matching using random forests for automatic semi-dense abdominal image registration, in: International Symposium on Biomedical Imaging (ISBI), pp. 490–493.

[7] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 886–893.

[8] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part based models. IEEE Trans. on Pattern Analysis and Machine Intelligence 32, 1627–1645.

[9] Giraud, R., Berthoumieu, Y., 2019. Texture superpixel clustering from patch-based nearest neighbor matching, in: European Signal Processing Conf. (EUSIPCO).

[10] Giraud, R., Ta, V.T., Bugeau, A., Coupé, P., Papadakis, N., 2017a. SuperPatchMatch: An algorithm for robust correspondences using superpixel patches. IEEE Trans. on Image Processing 26, 4068–4078.

[11] Giraud, R., Ta, V.T., Papadakis, N., 2017b. Evaluation framework of superpixel methods with a global regularity measure. Journal of Electronic Imaging (JEI) .

[12] Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D., 2008. Multi-class segmentation with relative location prior. Int. Journal of Computer Vision 80, 300–316.

[13] Huang, G.B., Jain, V., Learned-Miller, E., 2007a. Unsupervised joint alignment of complex images, in: IEEE Int. Conf. on Computer Vision (ICCV).

[14] Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E., 2007b. Labeled faces in the wild: A database for studying face recognition in unconstrained environments, in: Technical Report 07-49, Univ. of Massachusetts, Amherst.

[15] Kae, A., Sohn, K., Lee, H., Learned-Miller, E., 2013. Augmenting CRFs with Boltzmann machine shape priors for image labeling, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2019–2026.

[16] Kanavati, F., Tong, T., Misawa, K., Fujiwara, M., Mori, K., Rueckert, D., Glocker, B., 2017. Supervoxel classification forests for estimating pairwise image correspondences. Pattern Recognition (PR) 63, 561–569.

[17] Lazebnik, S., Schmid, C., Ponce, J., 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2169–2178.

[18] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep Learning. Nature 521, 436–444.

[19] Liu, ., Yang, J., Huang, C., Yang, M.H., 2015. Multi-objective convolutional learning for face labeling, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 3451–3459.

[20] Lowe, D.G., 2004. Distinctive image features form scale-invariant keypoints. Int. Journal of Computer Vision 60, 91–110.

[21] Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P., 2016. Universal Adversarial Perturbations, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 86–94.

[22] Muja, M., Lowe, D.G., 2014. Scalable Nearest Neighbor Algorithms for High Dimensional Data. IEEE Trans. on Pattern Analysis and Machine Intelligence 36, 2227–2240.

[23] Neubert, P., Protzel, P., 2015. Benchmarking superpixel descriptors, in: European Signal Processing Conf. (EUSIPCO), pp. 614–618.

[24] Olonetsky, I., Avidan, S., 2012. TreeCANN-kd tree coherence approximate nearest neighbor algorithm, in: European Conf. on Computer Vision (ECCV), pp. 602–615.

[25] Pei, S.C., Chang, W.W., Shen, C.T., 2014. Saliency detection using superpixel belief propagation, in: IEEE Int. Conf. on Image Processing (ICIP), pp. 1135–1139.

[26] Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234–241.

[27] Sawhney, R., Li, F., Christensen, H.I., 2014. GASP: Geometric association with surface patches, in: Int. Conf. on 3D Vision (3DV).

[28] Stutz, D., Hermans, A., Leibe, B., 2018. Superpixels: An evaluation of the state-of-the-art. Computer Vision and Image Understanding 166, 1–27.

[29] Tighe, J., Lazebnik, S., 2010. SuperParsing: Scalable nonparametric image parsing with superpixels, in: European Conf. on Computer Vision (ECCV), pp. 352–365.

[30] Tilquin, F., Conze, P.H., Pessaux, P., Lamard, M., Quellec, G., Noblet, V., Heitz, F., 2018. Robust supervoxel matching combining mid-level spectral and context-rich features, in: Int. Work. on Patch-based Techniques in Medical Imaging (Patch-MI, MICCAI), pp. 39–47.

[31] Zhang, Y., Pei, Y., Guo, Y., Ma, G., Xu, T., Zha, H., 2018. Consistent correspondence of cone-beam ct images using volume functional maps, in: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 801–809.