ImmersiveIML – Immersive interactive machine learning for 3D point cloud classification: the neural network at your fingertips

Maxime Cordeil, Thomas Billy, Nicolas Mellado, Loïc Barthe, Nadine Couture, Patrick Reuter

ImmersiveIML – Immersive interactive machine learning for 3D point cloud classification: the neural network at your fingertips

Maxime Cordeil*
University of Queensland

Thomas Billy
University of Queensland

Nicolas Mellado†
CNRS, IRIT, Université de Toulouse

Loïc Barthe‡
CNRS, IRIT, Université de Toulouse

Nadine Couture§
Univ. Bordeaux, ESTIA Institute of Technology

Patrick Reuter¶
Univ. Bordeaux, Inria, LaBRI, ESTIA

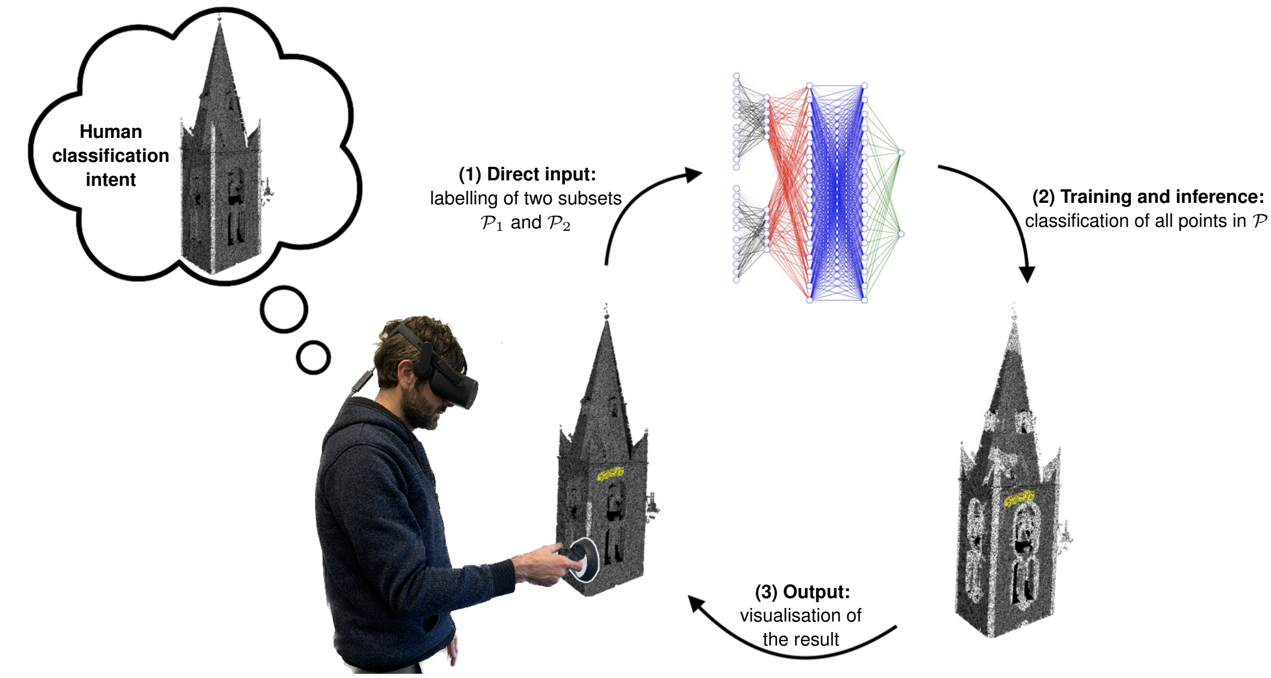


Figure 1: Starting from a human intent, the iterative human-in-the-loop process is composed of (1) 6-DOF controllers to allow direct interaction in large 3D point clouds via a minimal brushing technique, (2) a fast trainable machine learning model, and (3) the direct visual feedback of classification results. In the shown iteration, the classification result does not correspond to the user's intent, so they refine the input to eventually converge to an acceptable classification of the data, iteratively.

Abstract

We propose an initial exploration of an interactive machine-learning (IML) dialogue in immersive, interactive Virtual Reality (VR) for the classification of points in 3D point clouds. We contribute *ImmersiveIML*, an Immersive Analytics tool which builds on human-machine learning trial-and-error dialogue to support an iterative classification process of points in the 3D point cloud. The interactions in ImmersiveIML are designed to be both expressive and minimal; we designed the iterative process to be supported by (1) 6-DOF controllers to allow direct interaction in large 3D point clouds via a minimal brushing technique, (2) a fast trainable machine learning model, and (3) the direct visual feedback of classification results. This constitutes an iterative human-in-the-loop process that eventually converges to a classification according to human intent. We argue that this approach is a novel contribution that supports a constant improvement of the classification model and fast tracks classification tasks with this type of data, in an immersive scenario. We report on the design and implementation of ImmersiveIML and demonstrate its capabilities with two emblematic application scenarios: edge detection in 3D and classification of trees in a city LiDAR dataset.

Index Terms: Human-centered computing—Visualization—Visualization techniques; Human-centered computing—Interaction techniques—Gestural input; Human-centered computing—Interaction paradigms—Virtual reality; Computing methodologies—Neural networks

*e-mail: m.cordeil@uq.edu.au
†e-mail: nicolas.mellado@irit.fr
‡e-mail: loic.barthe@irit.fr
§e-mail: n.couture@estia.fr
¶e-mail: preuter@labri.fr

1 Introduction

3D point clouds are essential in many domains, such as self-driving cars, the mining industry, digital twins, engineering inspection and in biodiversity. In all these domains, automatically analysing 3D point clouds with machine learning is an active and challenging topic [12]. One of the challenges is the classification of subsets of 3D points as a first step to determine objects at a semantic level; for example, automated methods are needed to classify a given set of points that defines a tree or a car. So far, bespoke machine learning techniques demonstrate spectacular results to tackle these challenges. However, during a learning process, humans are often left out of the loop, and the lack of control to obtain a classification that matches the exact user needs (e.g., users need high accuracy, fast processing and domain-agnostic models) need to be addressed. Hence, there is a need for human input to help and verify classification output all at the same time. Human-in-the-loop machine learning approaches consist of tightly coupled interactions between a classification model and user input to refine the results [15]; with fast updates as an immediate response to incremental user input, the approach is called interactive machine learning [1]. In these approaches, it is important that users' input represents human intent, and that the human is efficiently and transparently integrated to take control of the learning process. Usually, user input for classification of 3D point clouds consists of fully manual interactive selections of the subsets of points of interest. Traditionally, these interactions are performed on flat screens with mouse and keyboard interfaces, and consist of drawing 3D bounding boxes or polygons around a set of 3D points. These interactions face usability issues due to the mismatch of the 3D nature of the data and the 2D user input and visualisation. First, users have to visually locate the boundaries of an object emerging from a point cloud, which can be occluded by other points in a cluttered scene. Second, because these interactive visualisation tasks are performed on 2D setups, they require significant user interaction to zoom, pan and rotate the 3D views to select the points, and very often with multiple views.

This work contributes an Immersive Analytics [5] solution to the intersection of two identified problems: (1) the user interface and interaction with the 3D data and (2) the interaction with a learning algorithm, with a human-in-the-loop interactive machine learning approach. Our ultimate goal is to provide a neural-network-at-the-finger-tips type of interface to the user, in an immersive way. In particular, we contribute the first exploration of a new pathway to fast track 3D point cloud classification by enabling an immersive dialogue between an embodied, immersive 3D point cloud and a fast, light weight neural network. Our work features the design of small but expressive, immersive visualisation and interactions techniques that allow users to communicate their intent to the neural network following a quick-and-dirty approach, in order to label, and ultimately classify, multiple points with minimum input, in an iterative manner. We demonstrate this initial exploration through the implementation of ImmersiveIML; we explain its design rationale and showcase its use in two scenarios: edge detection in a mechanical engineering part, and classification of trees in a LiDAR point cloud of a city.

2 Rationale and Design

Selection techniques for 3D point clouds mainly use two metaphors: 3D bounding box selections and 2D polygon selections. Those methods are costly in terms of interaction input and cognitive load as they require multiple views, but are also limited to 3D data were depth occlusion is minimal. On the other hand, techniques such as structure-aware lassos [21] or neural network based lassos like Lassonet [6] for 3D point clouds require small strokes input to select data. However, in the case of Lassonet, this selection requires to be trained on thousands of lasso records. This is obviously limited as those networks are too specialised on domains (e.g., buildings, mechanical engineering) and cannot be reused for different applications, without having to record tremendous amounts of strokes again. The use of VR and embodied interaction [7] with tracked controllers for point cloud annotation has been studied before [20]. Very recent work shows that higher levels of immersion better support the annotation tasks, eventually in combination with pre-trained machine learning models [9]. In contrast, ImmersiveIML includes the training stage of the machine learning model itself, embedded in the interaction dialogue. This allows for an AI-supported annotation of point clouds that can be specifically tailored to the user's intent. The trained network may also be transferred for annotating other point clouds.

Following the guidelines from Dudley and Kristensson [10], we designed ImmersiveIML with minimal, expressive human input in mind, and we focus on the human-in-the-loop classification dialogue with a light-weight neural network that can be used for generic point cloud classification.

2.1 Expressivity and minimal input for selection

Expressivity in terms of interaction has many definitions in HCI [4]. In our context, expressivity follows Benyon et al.'s definition [2]: we mean to seize the opportunities of gesture-based interactions afforded by 3D immersive VR setups, to facilitate human-input in the 3D space. The expressivity of movements provided by immersive VR controllers and human movements enables direct manipulation with the visualisation, and has shown to be advantageous in spatial tasks [18]. For a more expressive point selection and define points *intended* to be classified, that goes beyond specifying a 3D bounding box with VR Controllers, or drawing and extruding a polygon, we designed a free brush interaction, attached to the VR 3D controller. The user presses the trigger of the controller to paint and tag a set of points to define a class of objects. Our brush has an additive and a subtractive mode, which allows refining selected points in complex shapes (e.g., curved edges, convex shapes or objects that are less accessible due to clutter). In our interaction model, we envision a minimal brushing approach to determine points of interest and discriminate them against the rest of the data points. This brushing approach allows us to substantiate human intent into input, and it is somehow the 3D equivalent to minimal brushing interaction in regions of interest on 2D images used in AI image manipulation [3].

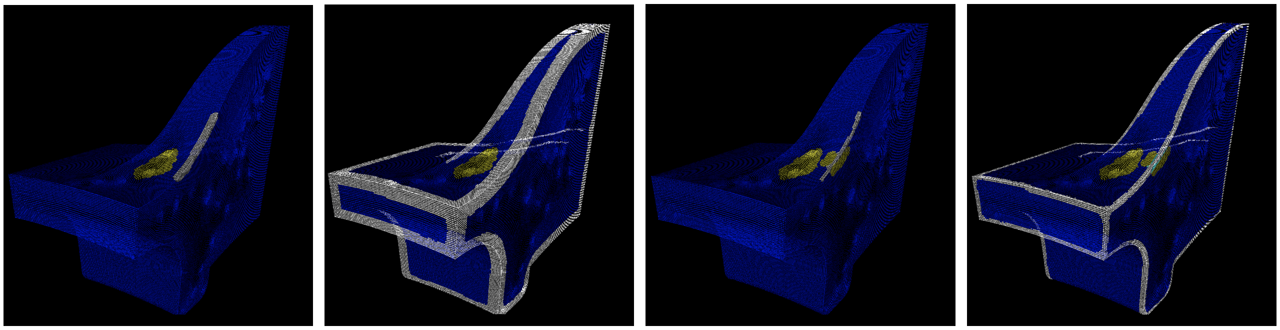
2.2 Interactive Machine Learning Dialogue

We define an interaction paradigm based on a simple dialogue between human point selection and the classification of the data points. This paradigm is based on interactive machine learning [19] where users are engaged to build a classifier by iteratively providing input until reaching a consensus. Note that compared to active learning [16], where the model would ask a user to label additional points, in interactive machine learning, the user is in control of the process and provides labeled input iteratively [15]. To achieve a high-quality dialogue we require to use a very fast model, both for training and inference. We chose PCEDNet [13], which provides both training on thousands of points and classification of million of points in seconds. This model is thus a lightweight neural network fast enough for supporting an interactive human/system dialogue.

From a formal point of view, we model the dialogue as follows: given a point cloud $\mathcal{P} = (p_1, ..., p_n)$, the user has a classification **intent** for the points in mind, that we denote $\mathcal{C}_{intent} = (c_1, ..., c_n)$, with $c_i \in \{1, ..., d\}$ for a classification into d classes and c_i is the class of the point p_i. In this paper, we present our approach with two classes ($d = 2$). According to their intent, the user iteratively labels two subsets $\mathcal{P}_1$ and $\mathcal{P}_2$ as class 1 and 2, respectively. Having provided these subsets, the user wants the network to learn the classification $(\mathcal{P}, C_{learn})$ for all points, so that $\mathcal{C}_{learn} = \mathcal{C}_{intent}$.

At a higher level, we define the human dialogue between the user and PCEDNet with the following user input:

- *[Input 1]: the user defines which points belong to class 1 of objects, according to their intent ($\mathcal{P}_1$).*
- *[Input 2]: the user defines which points belong to class 2, i.e., the 3D points that should not be classified as belonging to class 1, again according to their intent ($\mathcal{P}_2$).*



(a) Initial selection (316 white points and 732 yellow points brushed)

(b) First inference

(c) Refinement of the selection (390 white points and 449 yellow points brushed)

(d) Second inference

Figure 2: Brush selection of points and learning on a sampled version of the fandisk model (106,468 points)

Points that are neither in $\mathcal{P}_1$ nor $\mathcal{P}_2$ are ignored during training, and their classes will then be inferred by the network. In a generic manner, in ImmersiveIML's interface, these inputs are translated by brush colours; we use white for Input 1, yellow for Input 2. The user then enters in an iterative human-in-the-loop dialogue with PCEDNet responses:

- *[Output 1]*: PCEDNet performs the classification in interactive time and ImmersiveIML presents the results visually to the user — the entire point cloud is coloured according to the inferred classes.
- *[Output 2]*: It may happen that the neural network classifies some points in a different class than the one labeled by the user. For example, it may infer a point in $\mathcal{P}_1$ to belong to class 2 in C_{learn}. We present these points in cyan color; reasons for this misclassification include a user's accidental wrong or contradictory input or the neural network provided a wrong classification.

After visualising Ouput 1 and 2, if not satisfied with the classification (i.e., $\mathcal{C}_{learn} \neq \mathcal{C}_{intent}$), the user can redefine Input 1 and 2 (by adding or deleting points from $\mathcal{P}_1$ and $\mathcal{P}_2$ with boolean brush operations). The user repeats these actions until $\mathcal{C}_{learn} = \mathcal{C}_{intent}$. Note that, in each iteration, the user can also decide whether the neural network model should be trained from scratch with random model parameters, or whether it should continue the training with the model parameters from the last training iteration. The latter is particularly interesting when the user was globally satisfied with the former training result and made only slight changes in the labeling.

At this stage, ImmersiveIML can display the result of classification with two modalities: for each class, the colours of the original point cloud are either replaced with their respective class colours; or the user retains only one class to visualise and the other classes are filtered out. The dialogue between the user and PCEDNet via ImmersiveIML is a converging sequence of Input 1, Input 2 and Output 1, Output 2 to a satisfying classification.

2.3 Implementation

ImmersiveIML integrates two main components: a 3D immersive and interactive point cloud visualisation and selection tool, and the lightweight point classification neural network PCEDNet.

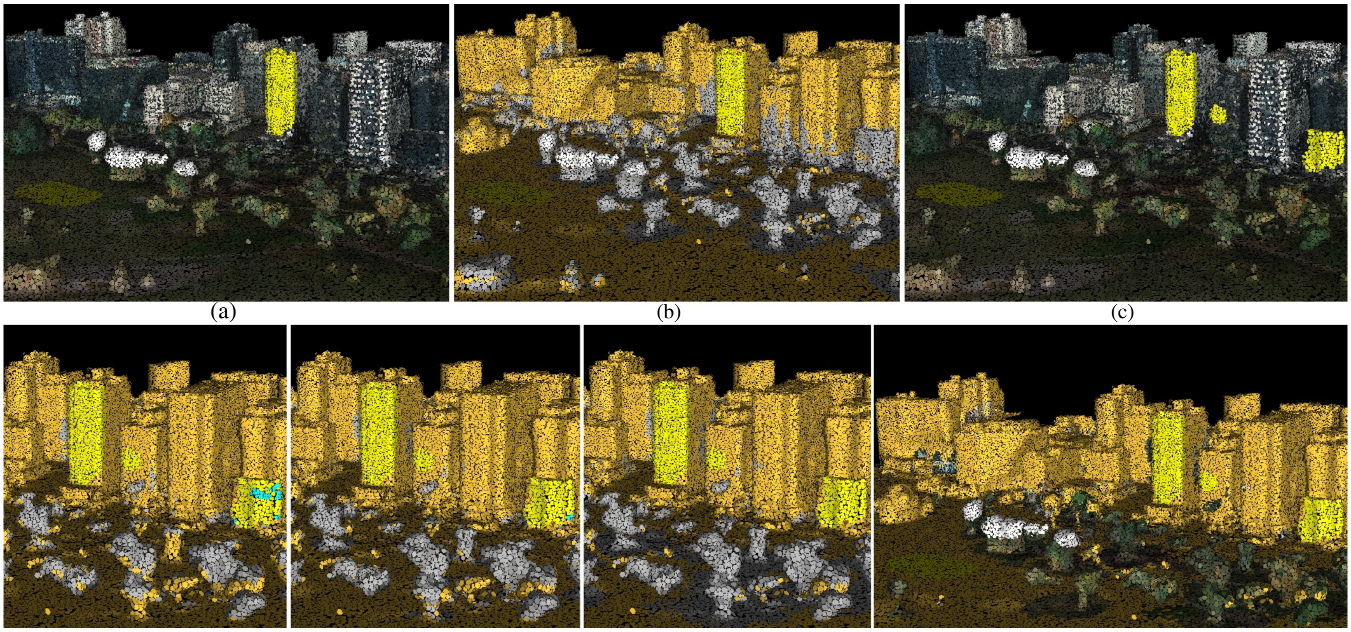
The 3D interactive visualisation is built with the Immersive Analytics Toolkit (IATK) [8]. IATK uses the Unity[1] game engine, and enables real time, high FPS visualisation of large 3D point clouds. The toolkit also provides a fast 3D brush based on compute shaders. We adapted IATK to be able to brush multiple classes of points with different colours. The brush interactions are supported via the 3D controller prefabs position in the space. Enabling a brush is performed by pulling the trigger of the VR controller. Switch buttons allow to change the class of the brush, so that it corresponds to the classification intent. All the examples in this paper require only two classes, and we used the colours white and yellow, respectively.

For learning the classification of points in a point cloud, we rely on PCEDNet that is based on the concept of a scale-space analysis put into a lightweight neural network, and that is very fast to train and able to classify millions of points in seconds. Although initially designed for point cloud edge detection, PCEDNet performs also well on more general classification tasks; it operates on the pre-computed differential information of the surrounding shape of each point in the point cloud at different scales. Scale is defined by the size of the neighborhood that is taken into account, and for each point $p_i \in \mathcal{P}$ and scale $t \in 1, ..., T$, a geometric feature vector $X_i^t \in R^6$ is associated that consists of 6 scalar values, as for example the signed curvature information defined by the local reconstructed shape [11, 14]. In PCEDNet, $T = 16$ scales are used, and so the input layer of the neural network consists of 96 input values. Hence, for the classification into d classes, the involved neural network learns a function $f : R^{96} \rightarrow \{1, ..., d\}$, by training only a few thousands of parameters. For more details on PCEDNet and more information about the timings for pre-computation, training, and classification, we refer the reader to [13].

In order to plug PCEDNet in ImmersiveIML, starting from the points p_i in the labeled subsets $\mathcal{P}_1$ and $\mathcal{P}_2$ and their associated pre-calculated geometric feature vectors X_i^t, we simply train the network on the labeled function values $f(X_i^1, ..., X_i^T) = 1$ for all points in $\mathcal{P}_1$ and $f(X_i^1, ..., X_i^T) = 2$ for all points in $\mathcal{P}_2$. Then, the network can infer the classification on all points in $\mathcal{P}$. By learning on the point's associated geometric feature vectors instead of the point positions, PCEDNet guarantees translation, rotation, and scale invariance.

For the integration into the IATK in Unity, we developed a client-server approach, based on a communication module with PCEDNet via the http POST and http GET commands in Unity. After selecting the points of interests for each class, the GPU returns instantly the ids of the selected points, which are then posted to the PCEDNet server. The PCEDNet server then returns a classification of the point cloud back to Unity. The classes are then read and the visualisation encodings updated — the 3D points belonging to the intended classified objects are fully opaque, the other points are fully transparent. The user can switch back to the initial visualisation encodings by pushing a button on the VR controller.

[1] `https://unity.com/`



(a) (b) (c) (d) (e) (f) (g)

Figure 3: The user intent is to classify all the trees in a small tile of a Melbourne city LiDAR model (205,745 points). (a) the user brushes trees (intended class, white, $\mathcal{P}_1$) and what is NOT a tree ($\mathcal{P}_2$, bright yellow, here, a building). (b) the machine learning model learns, creates an inference and ImmersiveIML displays the first inference of classified points (gray points are the inferred trees, orange points are considered not to be trees). (c) the user adds points to $\mathcal{P}_2$ by brushing parts of two other buildings, to improve the results. (d,e,f) The user triggers successive learning epochs to refine the result, which are assessed visually with the updated visualisations. Note that the numbers of points in cyan color (i.e., points that the network considers to belong to trees) are decreasing. (g) the results are considered satisfactory.

3 Scenarios

Data workers need to clean, organise and label their data. We envision ImmersiveIML to be used at multiple stages of the data wrangling process for analytics and learning. We focus on two scenarios that involve 3D point clouds and in which the user has different goals: (1) a low-level aim to determine geometric properties and (2) a higher-level task to define a class of objects from 3D point sets.

(1) Determining edges on a mechanical part – 3D scanners are used in engineering to retrieve the 3D structure of mechanical components, for repair and maintenance re-engineering. It is essential to detect faces, holes and edges on those point clouds in order to transform them into higher level data structures (e.g., triangular meshes) for further processing, e.g., 3D printing. In this scenario, the user needs to specify the edges of a mechanical part (here the fandisk model); using ImmersiveIML the user brushes some points that are perceived to belong to an edge, and brushes a small patch of points of a perceived uniform surface to explicit what is not an edge. After a first classification of the model is returned, the user wants to refine the classification of edges, as the strip near the conceptual edge is too wide; the user excludes some points of the thick edge with the brush (Figure 2). After the second iteration, the model returns satisfying thinner edges on the fandisk. ***Observation:*** *at a low-level and straight utilization of PCEDNet, this example demonstrates the usefulness of an immersive tool to explicitly define geometric properties with small, iterative and expressive spatial interaction.*

(2) Classifying trees in a slice of Melbourne's city centre – In this scenario, the user wants to quickly label the 3D points that belong to the trees, for example to determine the proportion of tree vegetation in a city. With ImmersiveIML, the users explicit their intent to classify the trees by performing 4 small brush strokes on different trees with the 3D controller. They then switch the brush to explicit other objects that should not be classified as a tree. When they release the trigger, PCEDNet performs the classification and returns a first visual result. The user perceives the result as being good but requires more precision, they explicitly tell the neural network that there are some remaining points that should not belong to the intended classified points. To do this, they brush parts of two other buildings. After running a few more epochs, the final iteration displays a satisfying classification result (Figure 3). ***Observation:*** *with minimal brush input and iterative dialogue, the scenario demonstrates how to fast track an initial point cloud classification and label many points as belonging to a conceptual tree class. Further refinements are needed which go beyond the scope of this initial exploration.*

4 Conclusion

We contributed ImmersiveIML, the first immersive and interactive machine learning interface for 3D point cloud classification. Initial exploration with exemplar datasets demonstrate the potential of the interactive machine learning based dialogue and the use of PCEDNet to interactively learn and classify sets of points.

Future work includes more design and research efforts to improve expressivity for point set classification; in particular we want to integrate more 3D point cloud properties in the dialogue, such as visualisations of histograms of point colours, density and other colour space and geometric properties for the user, to help the classification process, in combination with the intelligent brush approach. ImmersiveIML is at an initial exploration stage; in future work we would like to formally explore the benefits of such an interactive machine learning dialogue with controlled user studies. We also plan to assess whether a visualisation of the underlying neural network improves the user efficiency, and whether this leverages explainability of the machine learning model, following [17]. Moreover, we want to explore how other 3D point cloud specialised machine learning models could be suited for interactive machine learning to support an even wider range of selection scenarios. Ultimately, we envision ImmersiveIML to be a comprehensive 3D point cloud analysis tool that will support up to segmentation tasks.

References

[1] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[2] D. Benyon, K. Höök, and L. Nigay. Spaces of interaction. In *Proceedings of the 2010 ACM-BCS Visions of Computer Science Conference*, ACM-BCS '10, 2010.

[3] K. Blomqvist, L. Ott, J. J. Chung, and R. Siegwart. Baking in the feature: Accelerating volumetric segmentation by rendering feature maps. *CoRR*, abs/2209.12744, 2022. doi: 10.48550/arXiv.2209.12744

[4] M. Bruns, S. Ossevoort, and M. G. Petersen. Expressivity in interaction: A framework for design. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2021.

[5] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive analytics. In *2015 Big Data Visual Analytics (BDVA)*, pp. 1–8, 2015. doi: 10.1109/BDVA.2015.7314296

[6] Z. Chen, W. Zeng, Z. Yang, L. Yu, C.-W. Fu, and H. Qu. LassoNet: Deep lasso-selection of 3D point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):195–204, 2020. doi: 10.1109/TVCG.2019.2934332

[7] M. Cordeil, B. Bach, A. Cunningham, B. Montoya, R. T. Smith, B. H. Thomas, and T. Dwyer. Embodied axes: Tangible, actuated interaction for 3D augmented reality data spaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–12, 2020. doi: 10.1145/3313831.3376613

[8] M. Cordeil, A. Cunningham, B. Bach, C. Hurter, B. H. Thomas, K. Marriott, and T. Dwyer. IATK: An immersive analytics toolkit. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 200–209, 2019. doi: 10.1109/VR.2019.8797978

[9] A. Doula, T. Güdelhöfer, A. Matviienko, M. Mühlhäuser, and A. S. Guinea. PointCloudLab: An environment for 3D point cloud annotation with adapted visual aids and levels of immersion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11640–11646, 2023. doi: 10.1109/ICRA48891.2023.10160225

[10] J. J. Dudley and P. O. Kristensson. A review of user interface design for interactive machine learning. *ACM Trans. Interact. Intell. Syst.*, 8(2), jun 2018. doi: 10.1145/3185517

[11] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM Trans. Graph.*, 26(3):23–es, jul 2007. doi: 10.1145/1276377.1276406

[12] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021. doi: 10.1109/TPAMI.2020.3005434

[13] C.-E. Himeur, T. Lejemble, T. Pellegrini, M. Paulin, L. Barthe, and N. Mellado. PCEDNet: A lightweight neural network for fast and interactive edge detection in 3D point clouds. *ACM Trans. Graph.*, 41(1), Nov. 2021. doi: 10.1145/3481804

[14] N. Mellado, G. Guennebaud, P. Barla, P. Reuter, and C. Schlick. Growing Least Squares for the Analysis of Manifolds in Scale-Space. *Computer Graphics Forum*, 31(5):1691–1701, July 2012. doi: 10.1111/j.1467-8659.2012.03174.x

[15] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and A. Fernández-Leal. Human-in-the-loop machine learning: A state of the art. *Artif. Intell. Rev.*, 56(4):3005–3054, aug 2022. doi: 10.1007/s10462-022-10246-w

[16] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[17] S. Teso, Alkan, W. Stammer, and E. Daly. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, 6, 2023. doi: 10.3389/frai.2023.1066049

[18] W. Usher, P. Klacansky, F. Federer, P.-T. Bremer, A. Knoll, J. Yarch, A. Angelucci, and V. Pascucci. A virtual reality visualization tool for neuron tracing. *IEEE transactions on visualization and computer graphics*, 24(1):994–1003, 2017.

[19] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.

[20] F. Wirth, J. Quehl, J. Ota, and C. Stiller. PointAtMe: Efficient 3D point cloud labeling in virtual reality. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019. doi: 10.1109/IVS.2019.8814115

[21] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. Efficient structure-aware selection techniques for 3D point cloud visualizations with 2dof input. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2245–2254, 2012.