Interactive Semi-Automatic Labeling of Point Clouds Using Transformer-Based Descriptors

Patryk Najgebauer
Czestochowa University of Technology
Faculty of Computer Science and Artificial Intelligence
Czestochowa, Poland *patryk.najgebauer@pcz.pl*

Rafał Scherer
Czestochowa University of Technology
Faculty of Computer Science and Artificial Intelligence
Czestochowa, Poland
AGH University of Krakow
Faculty of Computer Science
and Center of Excellence in Artificial Intelligence
Krakow, Poland *rafal.scherer@pcz.pl*

Jakub Walczak
Lodz University of Technology
Institute of Information Technology
Lodz, Poland *jakub.walczak@p.lodz.pl*

Adam Wojciechowski
Lodz University of Technology
Institute of Information Technology
Lodz, Poland *adam.wojciechowski@p.lodz.pl*

Abstract

This paper presents a descriptor-based method for labeling point clouds using a two-stage transformer architecture. The first stage consists of an encoder that extracts descriptors from point cloud fragments. The second stage, a decoder, assigns labels to these fragments based on both the descriptor of the current fragment and an earlier predefined pattern descriptor. This approach functions as an interactive labeling tool similar to a brush, with the ability to reinforce or weaken the pattern through direct manipulation of its descriptor.

Keywords: point clouds, transformers, descriptors, encoder decoder

1. Introduction

The advent of large language models has enabled the solution of many tasks, such as point-cloud labeling, more effectively than ever before. However, given their size, these models often operate like brute-force methods, and handling huge datasets can become cumbersome. In this work, we focus on labeling raw data acquired with the Leica BLK360 3D laser scanner.

In practical scenarios, a scene may include several dozen scans of millions of points, and it is rarely possible to limit processing to a fixed, predefined set of classes. The scanning context, especially outdoors, can vary widely or even be unique. The context of a similar place may change with season, weather, or lighting conditions. Despite this variability, most of the existing methods address only a narrow range of classes, mainly due to the availability of standardized benchmarks that facilitate comparison and evaluation of different network architectures.

2. Related Work

Numerous methods for point cloud classification and registration are closely related to our approach. Much research has focused on designing effective feature extraction methods for point clouds. Early graph-based methods include PointNet and its extensions [2]. Projection-based approaches project the point cloud onto 2D surfaces and apply 2D convolutions like examples of KPConv and RangeNet++ [1]. Voxelization and clustering methods subdivide space into voxels or superpoints like MVX-Net [4] and Submanifold Sparse Convolutions [3] to leverage 3D convolutional architectures efficiently. Recently, transformer-based models have achieved the best performance on point cloud tasks. Their ability to handle irregular data makes them particularly well suited to point clouds [5], [6]. These advances motivate our descriptor-based, two-stage transformer framework for scalable point cloud labeling.

3. Proposed Semi-Labeling Method for Point Clouds

The core idea of our approach is an interactive "magic brush" that assists the user in manual labeling only the points that match a chosen pattern. The user begins by roughly marking a few keypoint regions of interest. Using these examples, the tool filters user selection during manual labeling to label only points that appear similar. The user remains in full control of the process, and if the user notices mislabeling or unwanted points, they can mark additional positive examples to reinforce the pattern or mark negative examples to suppress it.

3.1. Vision Transformer Model

Our method is based on the Vision Transformer (ViT) architecture, originally developed for image classification, with two key modifications. First, we replace several dropout layers with batch normalization to enhance the stability and repeatability of the resulting descriptor. Second, instead of using image patches, we construct point patches by extracting the k nearest neighbors of each point. For each patch, we compute a feature vector that includes RGB values, pairwise distances between points, and direction vectors relative to the local best-fit plane. To ensure consistent feature ordering for the transformer, we sort the neighboring points according to their projection order onto this plane. The overall system consists of two independent networks (Fig. 1): an encoder and a decoder. The encoder produces both a global descriptor for each point cloud fragment and a local descriptor for every point. During the creation of the labeling pattern, the descriptors of the user-selected fragments are aggregated into a single normalized global pattern descriptor. Users may also mark unwanted fragments to suppress those features. The decoder is only invoked during the labeling phase. It takes as input the local point descriptors generated by the encoder and the global pattern descriptor, and computes similarity scores that determine which points should be labeled.

3.2. Model Training

Our method operates directly on raw, unlabeled data from the 3D scanner, so we also apply the unsupervised learning method. We begin by sampling random fragments from these scans and create perturbed counterparts by adding noise to the data to alter the selection of k nearest neighbors. To prevent the encoder from exploiting point order or fragment centroids, we randomly shuffle points within each fragment. Both the original and perturbed fragments are passed through the encoder, and a similarity loss is optimized to align their global and local descriptors. Next, we train the decoder while fine-tuning the encoder by mixing descriptor data across batches. Each batch consists of multiple point-cloud fragments, each supplying its local point descriptors along with a single global pattern descriptor. These descriptors are shuffled to form positive and negative pairs for the decoder. Although fragments with similar context often

generate false negative examples, the model learns to distinguish true matches effectively.

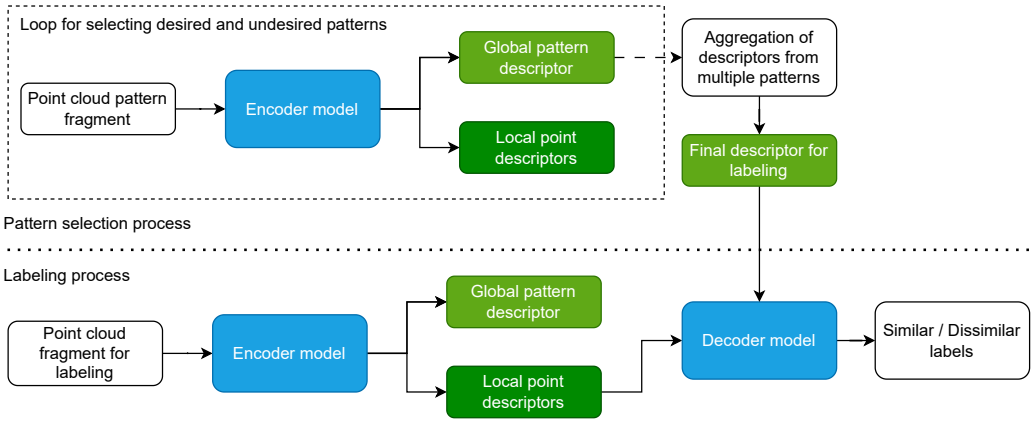

Fig. 1. Diagram of the labeling process and pattern selection.

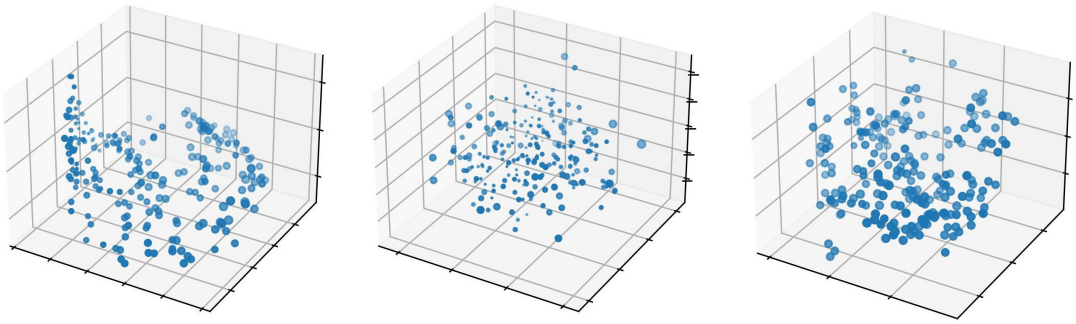

Fig. 2. Example of the attention strength per points.

4. Example results

Figure 3 presents several examples of labeling produced by our method. The process begins with the user selecting a pattern of interest. The encoder generates a corresponding global descriptor. During labeling, the user paints over fragments of the point cloud which are encoded and labeled by the decoder based on their similarity to the selected pattern. If the model incorrectly labels regions from other objects, especially near class boundaries, the user can mark them as negative examples to suppress their influence. This usually improves accuracy. However, using larger brush sizes increases the influence of the global descriptor over local descriptors, which may lead to boundary mislabeling. The encoder's attention mechanism, shown in Figure 2, highlights salient features such as edges or color outliers and distributes attention more evenly across flat regions. As a result, points near class boundaries contribute more strongly to the global descriptor, which can intensify labeling errors when using larger brushes. The method was tested on a laptop with an NVIDIA RTX 3050 GPU (4 GB) and 32 GB of RAM. This setup supported loading and displaying up to eight scenes with about 14 million points, rendered using the integrated Intel GPU. The model was run on the NVIDIA GPU. Larger brushes increased data transfer between the rendering and prediction components, becoming a performance bottleneck. The most effective configuration used a brush that covered around 1,000 points. Additional tests with a single GPU improved rendering performance, but required limiting brush size and

the number of scenes due to memory constraints. Moreover, data transfer between OpenGL and PyTorch remained a limiting factor.

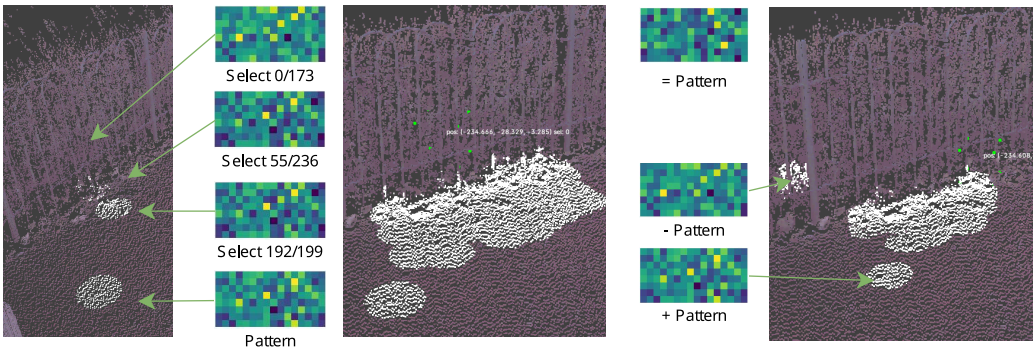

Fig. 3. Interactive labeling example.

5. Conclusions

The proposed method enables interactive, semi-automatic labeling of arbitrary point clouds through descriptor-based pattern recognition. It provides users with intuitive control over the labeling process and allows refinement via positive and negative example selection. While the approach offers effective and user-friendly labeling, certain challenges arise when larger brush sizes are used. These include the increased influence of the global descriptor on boundary points and the additional overhead caused by data transfer between the rendering (OpenGL) and prediction (PyTorch) devices. Despite these limitations, the method remains a practical assistive tool, offering users full control and the ability to manually refine labeling results as needed.

References

[1] Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 4213–4220. IEEE (2019)

[2] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)

[3] Schmohl, S., Sörgel, U.: Submanifold sparse convolutional networks for semantic segmentation of large-scale als point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4, pp. 77–84 (2019)

[4] Sindagi, V.A., Zhou, Y., Tuzel, O.: Mvx-net: Multimodal voxelnet for 3d object detection. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7276–7282. IEEE (2019)

[5] Sun, J., Qing, C., Tan, J., Xu, X.: Superpoint transformer for 3d scene instance segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2393–2401 (2023)

[6] Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler faster stronger. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4840–4851 (2024)