Collaborative VR-based 3D Labeling of Live-captured Scenes by Remote Users

Domenic Zingsheim
University of Bonn

Patrick Stotko
University of Bonn

Stefan Krumpen
University of Bonn

Michael Weinmann
University of Bonn

Reinhard Klein
University of Bonn

Abstract—Previous work on interactive 3D labeling focused on improving user experience based on virtual/augmented reality and, thereby, speeding-up the labeling of scenes. In this paper, we present a novel interactive, collaborative VR-based 3D labeling system for live-captured scenes by multiple remotely-connected users based on sparse multi-user input with automatic label propagation and completion. Hence, our system is particularly beneficial in the case of multiple users that are able to label different scene parts from the respectively adequate views in parallel. Our proposed system relies on (1) the RGB-D capture of an environment by a user, (2) a reconstruction client that integrates this stream into a 3D model, (3) a server that gets scene updates and manages the global 3D scene model as well as client requests and the integration/propagation of labels, (4) labeling clients that allow an independent VR-based scene exploration and labeling for each user, and (5) remotely connected users that provide a sparse 3D labeling used to control the label propagation over objects and the label prediction to other scene parts. Our evaluation demonstrates the intuitive collaborative 3D labeling experience as well as its capability to meet the efficiency constraints regarding reconstruction speed, data streaming, visualization and labeling.
INTRODUCTION

The progress in Augmented Reality (AR) / Virtual Reality (VR) technology has a tremendous effect on various applications. Imagine people located at different physical places that are within the same (virtual) environment, where the latter could be simultaneously captured by one of the users, thereby resulting in a meeting in the respective user’s local environment. When talking about certain objects in the environment, we can mark them by touching their surfaces and drawing semantic labels to specify the respective object category. During such interaction with the scene, a learned model is empowered to get an improved understanding regarding the particular categories, thereby, allowing to cope with more complex variations in shape and appearance. Even, further, when seeing new instances of the same category in another part of the local environment, the model will transfer the learned knowledge and additionally refine the resulting scene labeling based on the user interaction. Instead of relying on offline pre-trained models, such simple and intuitive labeling of the scene seems possible purely based on online learning.

Indeed, 3D scene understanding is a central task in computer vision and graphics with numerous applications such as autonomous driving, urban planning, architecture, medical applications, design or education. The semantic decomposition of a scene into its respective semantic entities serves as a prerequisite for many follow-up tasks that rely on the geometric contextual relationships between these entities. In particular, time-critical scenarios such as disaster management require expert annotations already during acquisition for (almost) the entire scenes to guide follow-up tasks. E.g. in disaster management, we may have to label the terrain that can be accessed by a robot as well as the objects it can interact with (including smaller individual objects relevant for a task but also generally moveable objects like chairs or doors to reach certain locations) which therefore also represents the fundamental prerequisite for path planning. This means that we have to capture the current state of a certain scene and provide respective labels as soon as possible to allow further robots to access the scene (for bringing equipment, for rescue, etc.).

Most often, semantic segmentation of 3D scenes has been addressed after the acquisition of a scene in terms of images, RGB-D data or 3D models, typically with learning-based frameworks that exploit knowledge in terms of a-priori information in the form of labeled 3D scene datasets. As such automated semantic segmentation approaches may encounter problems in accurately reproducing object boundaries or the respective pre-trained models may not cover relevant categories for a considered scene, interactive scene segmentation has gained a lot of attention in recent years. Beyond respective screen-based annotation systems [1], [2], [3], [4], interactive semantic segmentation has also been addressed based on AR and VR systems [5], [6], [7], [8] to enhance the labeling experience of the user. While the aforementioned approaches where designed with the goal of improving user experience during labeling and thereby speeding-up the labeling of scenes, none of them addresses performing semantic 3D scene segmentation already during acquisition. Hence, these approaches are impractical for collaborative scenarios including remote consulting applications, safety applications, maintenance applications or disaster management.

In this paper, we present a novel interactive, collaborative VR-based 3D labeling method for live-captured scenes based on sparse multi-user input with automatic propagation and completion. Our system allows users to provide hints for the automatic 3D labeling based on brush strokes as well as refining the current labeling. This manual labeling is motivated by the fact that the individual characteristics of a scene are not necessarily represented in pre-recorded semantic datasets. In the presence of multiple users that are allowed to label scene parts from the respectively optimal views in parallel e.g. to allow the robot to enter the scene and interact with it at an earlier stage, our approach is particularly beneficial and results in a faster labeling of entire large-scale scenes where many different parts have to be simultaneously labeled as well as when using multiple devices for capture. Thereby, we address the major challenges of providing an intuitive scene interaction mechanism, that allows multiple remotely connected users to independently explore and label a scene, as well as meeting strong real-
time constraints regarding reconstruction speed, data streaming, visualization and labeling. Our evaluation demonstrates the intuitive 3D labeling experience in the live-captured scene as well as its capability to meet these constraints imposed by collaborative scenarios.

Related Work

3D Segmentation: Semantic segmentation refers to the task of decomposing a scene into its semantic entities and has gained a lot of attention in the past. The large majority of approaches exploits knowledge in terms of an ever growing number of 3D scene datasets with annotations [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] to follow a supervised learning strategy. After training models on the respective datasets, the models are used to predict semantic labels for novel data not included during training. However, obtaining corresponding datasets remains a challenging task. While annotations can be easily provided for synthetic datasets [11], the labeling of real-world datasets comes at a high manual effort which makes gathering annotations using cloud services such as Amazon Mechanical Turk (AMT) an expensive process. As suitable pre-trained models may typically not be available to automatically predict the desired labeling or not allow a perfect labeling, the annotation process is based on 2D images of the scene that are shown to individual subjects who draw a respective 2D labeling. This process may already take several minutes per image (depending on the complexity of the scene) and result in severe costs. In addition, the contours of the labeled 2D image regions may still be rather inaccurate. With our approach, we aim at allowing an intuitive joint interactive annotation experience for live-captured 3D datasets by multiple users, where the users only provide labels for certain surface points and these annotations are propagated over 3D object surfaces.

Interactive 3D Labeling: As automated semantic segmentation approaches may not accurately reproduce object boundaries or respective pre-trained models may not cover relevant categories for a considered scene, several approaches focus on interactive scene segmentation. To overcome the limited user experience of labeling 2D images of a scene, several approaches focused on providing a 3D modeling experience. Seminal work has been presented with SemanticPaint [1], [2], a framework that allows the user to interactively annotate a scene by touching scene surfaces directly within a classical 3D reconstruction pipeline. To improve the user experience, the coloring of large scene parts is achieved by propagating the provided user labels over surfaces based on region growing techniques. Further screen-based interactive segmentation approaches are based on the coupling of 2D and 3D information [3] as well as the exploration of depth information [4]. The Semantic PaintBrush technique [5] extends this approach by allowing the user to navigate through a respective environment based on an AR device and scene labeling is guided by a laser pointer interaction that is used to identify and label homogeneous areas. To improve the labeling quality, AR-based labeling based on 3D data and the tracking provided by AR systems has also been approached by directly marking object boundaries in world space [6]. Further work [7] explored a VR framework to explore reconstructed environments and provide respective semantic labels based on a fast, easy-to-use coloring mechanism in terms of shooting labels onto surfaces. Beyond semantic annotations, some approaches also considered interactive scene annotation in terms of placing shading and reflectance strokes onto real-world geometry in an AR setting [8]. In contrast to these approaches, we target the collaborative 3D labeling of live-captured scenes, which requires meeting stronger performance constraints while also allowing an intuitive (VR-based) experience of the scene.

Overview

In this paper, we present a collaborative reconstruction, labeling and annotation system where multiple users are able to jointly create a labeled and annotated 3D model for simultaneously captured scenes. Major challenges include the development of an intuitive scene interaction mechanism that allows users to independently explore and label a scene. Furthermore, the overall system performance has to meet strong real-time constraints regarding reconstruction speed, data streaming, visualization and labeling.

To meet these goals, our system as shown in
Figure 1 involves the following components:

- the RGB-D capture of a respective environment by a local user or robot,
- a real-time reconstruction client that integrates incoming RGB-D images into a 3D model which is streamed to the server component,
- a central server that is responsible for managing the global 3D scene model, handling client requests, and integrating as well as propagating the semantic labeling provided by users,
- labeling clients that request parts of the global 3D model for individual remotely connected users to allow an independent real-time VR-based scene exploration and labeling, and
- remotely connected users that independently explore and control the semantic labeling the respective scene.

In the following, we describe the major components in more detail.

**Reconstruction Client:** The key objective of the reconstruction client is given by the reliable real-time 3D reconstruction of a scene. Similar to previous work [19], we integrate incoming RGB-D images into a spatially-hashed, dense voxel-based 3D model and stream the volumetric data to the server component. Since only the currently visible scene parts is updated, the reconstruction performance is independent of the overall scene size.

**Central Server:** The central server component is responsible for maintaining the global 3D model as well as managing the states of connected clients and requests. For this purpose, we use a truncated signed distance function (TSDF) voxel block volume, like at the reconstruction client, as well as a second volume consisting of the bandwidth-efficient Marching Cubes (MC) voxel representation [19]. Since we are also interested in a semantic labeling of the scene parts, we extend both voxel data types with an 8-bit label value which is split into an ID (6 bits) and a label group (2 bits) [1]. Here, the ID is used to distinguish between different semantic classes whereas the label group denotes the source that generated the label, i.e. the three aforementioned types of labels. When either the reconstruction client sends updated scene data or the labeling clients send the user’s labeling action, the server integrates these updates into its global model. The list of updated voxel blocks is then added to the stream set of each connected labeling client which defines their individual state of updated 3D data.

**Labeling Client:** For a collaborative, efficient and intuitive 3D labeling and annotation, like in the case of physical touching or painting on surfaces, we allow remotely connected users simultaneously to enter and interact with a respective environment based on VR devices. This is particularly beneficial, as it decouples the scene exploration from the sensor used for scene capture and, in turn, allows an independent scene exploration for each user. We render a sphere on top of the virtual VR controller which provides feedback to the user regarding the position and size of the painted label stroke as well as regarding the currently selected label via its color as shown in Figure 2. We consider multiple types of labels, each representing a different trade-off between accuracy and automation. To achieve an interactive and intuitive experience, we only consider the currently visible scene parts for labeling which also has the benefit of being independent of the scene size. Furthermore, we give the user explicit control to switch between label types as well as starting/stopping the corresponding automatic labeling processes. Since a significant amount of computational resources is required...
for the visualization to provide a pleasant VR experience, the labeling client only sends the respective commands to the server and receives scene updates. This offloads the computational burden of the labeling process to the server.

**Interactive 3D Labeling**

In this context, we provide an overview of different label types used in our 3D labeling approach, that relies on sparse multi-user inputs (in the form of strokes) that are automatically propagated over individual objects and used for completing the labeling of the scene, and, hence, reduce the manual labeling effort. Furthermore, we discuss a refinement of the labeling by smoothing isolated (wrongly-predicted) labels.

**Considered Label Types**

Only relying on user-specified labels would limit the annotation speed and slow down the overall segmentation process as users would have to paint each surface part. Therefore, we consider various types of labels in the form of User Labels, Propagated Labels, and Predicted Labels.

**User Labels:** Labels of this type are directly provided by a user and are typically highly accurate. Therefore, we consider such labels as ground truth data, i.e. hard constraints, that cannot be changed to other label types. In our VR scenario, a user labels the scene with a 3D brush consisting of the selected label, the 3D scene position $p \in \mathbb{R}^3$, and the brush size $d \in \mathbb{R}$. When the server receives this update, it collects all voxels in the TSDF voxel block volume within the brush

$$
\mathcal{B} = \{ x \in \mathbb{R}^3 \mid \| x - p \|_2 < d/2 \}
$$

and sets their label to the user-specified label. Afterwards, all affected blocks in the MC voxel block volume are updated as well, similar to the integration procedure for the updates received by the reconstruction client [19]. Here, we collect the labels of all neighboring voxels contributing to the local surface defined by Marching Cubes and choose the dominant label in that region to enforce consistency for the streamed data.

**Propagated Labels:** In real-world scenarios, the user should not be forced to label the complete scene manually as this inherently limits the annotation speed and does not scale for larger scenes. Instead, the user-labeled parts should automatically propagate over the object’s surface up to its boundaries. We use an efficient image-based approach where label propagation is only applied to the currently visible parts of the scene. This avoids unnoticed misclassifications and allows the user to rapidly correct them. For this, we extract a low-resolution view of the 3D model from the user’s current pose using a raycasting operation. For each pixel that contains labeled scene data, we collect the corresponding voxels of the neighboring pixels and propagate their label if the surface is similar. Similarity is determined by comparing the position, color, and normal to avoid propagation across object discontinuities and boundaries. As scene reconstruction tends to smooth sharp edges due to sensor noise, using a low-resolution view is not only more efficient but also allows for better detecting these object boundaries.

**Predicted Labels:** In addition to propagating labels, we also consider label predication to automatically annotate similar reoccurring scene parts by exploiting knowledge of the already labeled parts. Here, the scene parts that contain labels defined either by the user or via propagation are used to continuously train a random forest in an online fashion during the reconstruction and segmentation process. We use *Voxel Oriented Patch features* (VOP) [2] as the descriptor of a given voxel which are rotation-invariant and incorporate the color information of adjacent voxels in LAB space, surface normals and the height of the voxel. The random forest itself consists of a set of $n \in \mathbb{N}$ binary decision trees $T_i$, each mapping the VOP feature space $\mathcal{F}_{\text{VOP}}$ to a discrete label space $\mathcal{L}$. In each training step, we sample a fixed number of voxels that are currently visible to the user. Here, we use the reservoir sampling technique...
of the Streaming Random Forest algorithm [2] to keep the number of training examples per label balanced and unbiased over multiple frames. This allows for a good training of the classifier on the CPU. After each training step, we copy the current state of the random forest to the GPU to accelerate the prediction step by performing the inference in parallel for each voxel. For a fixed number of uniformly sampled scene voxels, we compute their VOP features and traverse the random forest until we reach the leaf nodes of each decision tree. Finally, we average over the probability distribution of each leaf and choose the label

$$c^* = \arg \max_c \frac{1}{n} \sum_{i=1}^{n} p_i(c \mid v)$$ (2)

where $p_i(c \mid v)$ is the posterior distribution given a voxel $v$ stored at the leaf of tree $i$ [20].

Label Smoothing

In complex scenes, the random forest predictor will not always produce satisfactory results since the classifier is strongly dependent on already observed scene elements and the color of scene geometry. While our human-in-the-loop pipeline is designed to correct misclassifications interactively, single wrongly predicted labels might be noticeable and perceived as noise in the segmentation. Correcting these by hand would be tedious and slows down the segmentation process. Therefore, we handle such noisy classifications by an additional smoothing step which performs a majority voting in the TSDF voxel block volume. For each pixel in the raycasted view, we create a histogram of neighboring pixel labels and determine the label with the highest number of votes. If it is the dominant label in that local region, i.e. if it surpasses a fixed threshold $\delta = 6$ in our case, it is considered reliable and stored in the corresponding voxel. Since these single predictions are created by the random forest classifier, we only smooth labels that have the forest label group.

Evaluation

We tested our collaborative reconstruction and labeling system with multiple users and used a laptop and up to 3 desktop computers, each taking the role of a different component of our system. All computers were equipped with an Intel Core i7-8700K CPU (laptop) or an Intel Core i7-4930K CPU (desktop), 32 GB RAM and a NVIDIA GTX 1080 GPU with 8GB VRAM. We used a HTC Vive and a HTC Vive Pro for visualization with rendering resolutions of $1512 \times 1680$ and $2016 \times 2240$ respectively. Furthermore, we recorded an office scene with a Microsoft Azure Kinect ($640 \times 576$ at 30 Hz) and used the heating room scene [19] which was captured by a Microsoft Kinect v2 ($512 \times 424$ at 30 Hz). For volumetric reconstruction, we use a voxel size of 5 mm and a truncation region of 60 mm.

User Study

To demonstrate the benefits of our system, we conducted a user study with ten participants.
that were asked to fully label a large, unknown scene during reconstruction based on different conditions. Here, we compared 2D-based and VR-based labeling controls to get insights regarding the benefits on user experience as well as single- vs. multi-user labeling of another scene together with a more experienced user where audio communication was additionally included. All participants were naïve to the goals of the experiment, provided informed consent, reported normal or corrected-to normal visual and hearing acuity. Furthermore, an initial training based on a different scene was used to make users familiar with the respective controls and devices.

In Figure 3, we provide an overview on the user-provided ratings and p-values obtained by a paired t-test. In average, the users reported a higher degree of situation awareness and intuitiveness for interacting with and labeling the scene which resulted in an overall better experience. The perceived stress level to accurately and quickly finish the task was similar for all modes. Collaborative labeling and communication with a second user was considered extremely helpful and further enhanced the experience. The time needed to complete the task (see Figure 4) was considerably higher than the reconstruction time because inexperienced users tended to use the automatic label propagation and prediction less often. In case of conflicts, we observed that users directly corrected inconsistently set labels or clarified doubts via the audio communication. Furthermore, the users were able to finish the task in considerably less time indicating that our system is beneficial in large-scale scenarios. In future work, a larger user study would be beneficial for further investigating the statistical significance of the observed improved average performance of our system.

**Streaming Progress**

To illustrate the performance of our system, we measured the streaming progress and bandwidth requirements over time (see Figure 5). These quantities depend on the chosen number and frequency of requested voxel blocks, so each labeling client can independently adjust these parameters to trade streaming latency for bandwidth requirements. We used a fixed package size of 350 voxel blocks per request and a request rate of 60 Hz for all clients. This leads to moderate bandwidth requirements of up to 50 MBit/s per client in the limit when a large amount of data is streamed constantly over time. Since the scene updates are queued based on the user labeling actions, we can see common patterns in the graphs.
that can be assigned to the label types. In the first minute of the session, the users have labeled objects using the 3D brush which only has a small impact on the total streaming performance. Afterwards, they performed label propagation which results in a significant increase of scene updates and required bandwidth. In the second half of the session, they mostly performed label prediction to increase the automation of the labeling process. In contrast to the other two types, this operation updates all visible scene parts in every step leading to a constantly high and fluctuating number of updates. Reducing the streaming rate of the scene data summarizes such updates and reduces the bandwidth requirements, but also introduces more latency into the collaborative labeling process.

**Labeling Performance**

In addition to the streaming progress analysis, we also provide a comparison between the different label types to demonstrate their influence on the overall labeling performance. In Figures 6 and 7, we show the reconstructed, colored 3D scenes, their final labelings by the users as well as highlightings of each label type. Our system only requires few user annotations which are not required to cover the whole surface of an object. The label propagation step automatically extends these user strokes to the boundaries such that single objects can be quickly labeled. As shown in the aforementioned streaming performance results as well as in the accompanying video, after a sufficient initial user labeling and propagating, the labels of similar scene parts can be predicted with a good accuracy and only requires small corrections by the user.

**Limitations**

Like in any online real-time reconstruction system, artifacts and misalignments in the 3D model might occur due to abrupt camera movements or sensor limitations. The latter includes multi-path interferences in time-of-flight based depth cameras which makes detecting object edges and boundaries more challenging and labels may propagate across them. Furthermore, the random forest based approach with on-the-fly training to predict labels may not only misclassify single voxels but larger regions due to the limitations of the learned feature which is a compromise between compactness and accuracy. While, by design, our system allows to quickly correct these misclassifications, more sophisticated automatic approaches will further improve the accuracy and annotation speed. Finally, the labeling of distant scene parts or difficult-to-reach surfaces...
would benefit from adding a laser-pointer-like label shooting [5], [7] as an alternative to the purely touch-based labeling.

Conclusions

We presented a novel interactive, VR-based 3D labeling method for live-captured scenes based on sparse multi-user input with automatic label propagation and completion. Hence, our system is particularly beneficial in the case of multiple users that are able to label different scene parts from the respectively adequate views in parallel. Besides a VR-based intuitive and independent scene exploration and 3D labeling experience, we demonstrated our system to also meet the efficiency constraints regarding reconstruction speed, data streaming, visualization and labeling. In the future, we see a great potential of our approach for cooperative scenarios such as VR-based education, remote consulting and disaster management. With further increasing hardware capabilities, we also expect the performance of our system to be improved by more complex feature types or alternative prediction modules.

Acknowledgment

This work was supported by the DFG project KL 1142/11-1 (DFG Research Unit FOR 2535 Anticipating Human Behavior).

REFERENCES

Domenic Zingsheim studies Computer Science at the University of Bonn, Germany, where he received a BSc in 2020 and is currently pursuing the MSc degree. His research interests include real-time 3D reconstruction and rendering. Contact him at sdzo@uni-bonn.de.

Patrick Stotko studied Computer Science at the University of Bonn, Germany, where he received a BSc in 2014 and a MSc in 2016. He is currently working as a PhD student with the Visual Computing Group. His research focuses on interactive geometry and appearance reconstruction techniques with RGB-D cameras in the context of AR/VR applications. Contact him at stotko@cs.uni-bonn.de.

Stefan Krumpen studied Computer Science at the University of Bonn, Germany, where he received a BSc in 2014 and a MSc in 2016. Since 2016, he is working as a PhD student with the Visual Computing Group. His research focus lies on high quality reflectance representations using BTFs and VR applications. Contact him at krumpen@cs.uni-bonn.de.

Michael Weinmann studied Electrical Engineering and Information Technology at the University of Karlsruhe, Germany, where he received his degree Dipl.-Ing. in 2009. After joining the Visual Computing Group at the University of Bonn in 2010, he received his PhD in computer science in 2016. His research interests include machine learning, 3D reconstruction, reflectance reconstruction, scene understanding and visualization. Contact him at mwe@cs.uni-bonn.de.

Reinhard Klein studied Mathematics and Physics at the University of Tübingen, Germany, from where he received his MS in Mathematics (Dipl.-Math.) in 1989 and his PhD in computer science in 1995. In 1999 he received an appointment as lecturer (“Habilitation”) in Computer Science also from the University of Tübingen, with a thesis in Computer Graphics. In 1999 he became Associate Professor at the University of Darmstadt, Germany and head of the research group Animation and Image Communication at the Fraunhofer Institute for Computer Graphics. Since 2000 he is professor at the University of Bonn and director of the Institute of Computer Science II - Visual Computing.