ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Shooting Labels: 3D Semantic Labeling by Virtual Reality

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Pierluigi Zama Ramirez, Claudio Paternesi, Luigi Di Lella, Daniele De Gregorio, Luigi Di Stefano (2020). Shooting Labels: 3D Semantic Labeling by Virtual Reality. Institute of Electrical and Electronics Engineers Inc. [10.1109/AIVR50618.2020.00027].

Availability:
This version is available at: https://hdl.handle.net/11585/806564 since: 2021-02-25

Published:

DOI: http://doi.org/10.1109/AIVR50618.2020.00027

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

10 November 2026

This is the final peer-reviewed accepted manuscript of:

P. Z. Ramirez, C. Paternesi, L. D. Luigi, L. Lella, D. D. Gregorio and L. D. Stefano, "Shooting Labels: 3D Semantic Labeling by Virtual Reality," 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), Utrecht, Netherlands, 2020, pp. 99-106

The final published version is available online at ***https://dx.doi.org/10.1109/AIVR50618.2020.00027***

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Shooting Labels: 3D Semantic Labeling by Virtual Reality

Pierluigi Zama Ramirez[1], Claudio Paternesi[1], Luca De Luigi[1], Luigi Lella[1], Daniele De Gregorio[2], Luigi Di Stefano[1]
[1]Department of Computer Science and Engineering (DISI) - University of Bologna, Italy
[2]Eyecan.ai
{pierluigi.zama, luca.deluigi4, luigi.distefano}@unibo.it
{claudio.paternesi, luigi.lella}@studio.unibo.it
daniele.degregorio@eyecan.ai

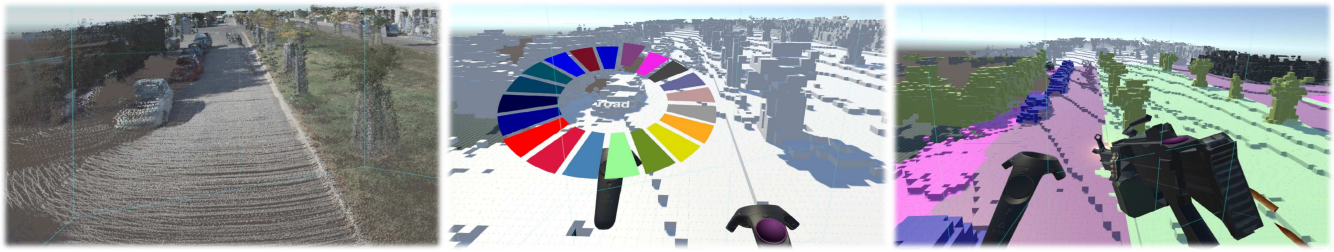

Fig. 1: **V**irtual **R**eality view of a 3D reconstruction from KITTI sequence. Left image: RGB visualization of the virtual world. Middle image: label palette and empty voxelization. Right image: partially labeled environment.

***Abstract*—Availability of a few, large-size, annotated datasets, like ImageNet, Pascal VOC and COCO, has lead deep learning to revolutionize computer vision research by achieving astonishing results in several vision tasks. We argue that new tools to facilitate generation of annotated datasets may help spreading data-driven AI throughout applications and domains. In this work we propose Shooting Labels, the first 3D labeling tool for dense 3D semantic segmentation which exploits Virtual Reality to render the labeling task as easy and fun as playing a video-game. Our tool allows for semantically labeling large scale environments very expeditiously, whatever the nature of the 3D data at hand (e.g. point clouds, mesh). Furthermore, Shooting Labels efficiently integrates multi-users annotations to improve the labeling accuracy automatically and compute a label uncertainty map. Besides, within our framework the 3D annotations can be projected into 2D images, thereby speeding up also a notoriously slow and expensive task such as pixel-wise semantic labeling. We demonstrate the accuracy and efficiency of our tool in two different scenarios: an indoor workspace provided by Matterport3D and a large-scale outdoor environment reconstructed from 1000+ KITTI images.**

I. Introduction

Two major leitmotifs in nowadays computer vision are: *Convolutional Neural Networks have surpassed human performance in classification tasks* [1] and *The success of the modern Deep Neural Networks (DNNs) is ascribable to the availability of large datasets* [2].

As for the latter, one might just consider the dramatic advances brought in by large annotated datasets like ImageNet [3] and Pascal VOC [4] in the fields of image classification and object detection, as well as by KITTI [5] and Cityscapes [2] in the realm of dense scene understanding. Indeed, the key issue in modern computer vision deals more and more with how to speed-up and facilitate acquisition of large annotated datasets. Innovative companies, like Scale.ai[1], Superannotate.ai[2] and many others, are gaining increasing popularity thanks to their advanced image labeling tools. This suggests data annotation to play a pivotal role alongside with the development of data-driven AI techniques.

The annotation process is notoriously tedious and expensive. Moreover, the more complex the visual perception task, the slower and more costly becomes the required annotation work. For instance, labeling a single image for 2D semantic segmentation, one of the most complex annotation tasks together with instance segmentation, can take several hours per image. Thus, as proposed in [6]–[9], directly annotating a 3D reconstruction of the scene and then projecting the 3D labels into 2D images can facilitate the data generation process significantly. On the other hand, the annotation process for 3D Dense Semantic Segmentation requires expertise in 3D modeling and several consecutive hours of manual labeling, which makes it an error-prone process, also due to inherent difficulties in handling 3D data (occlusions, 3D boundaries, ambiguities introduced by the 3D reconstruction process). Thus, existing 3D labeling tools [7], [8], [10] employ several annotators to iteratively refine the labeled data and attain the so-called *ground truth*, which, nonetheless, turns out far from perfect. For instance, in Figure 2 we can see exemplar annotation errors (e.g. a window labeled as a table). The majority of annotated datasets, both 2D and 3D, are obtained through similar procedures and thus

[1]https://scale.ai/
[2]https://superannotate.ai/

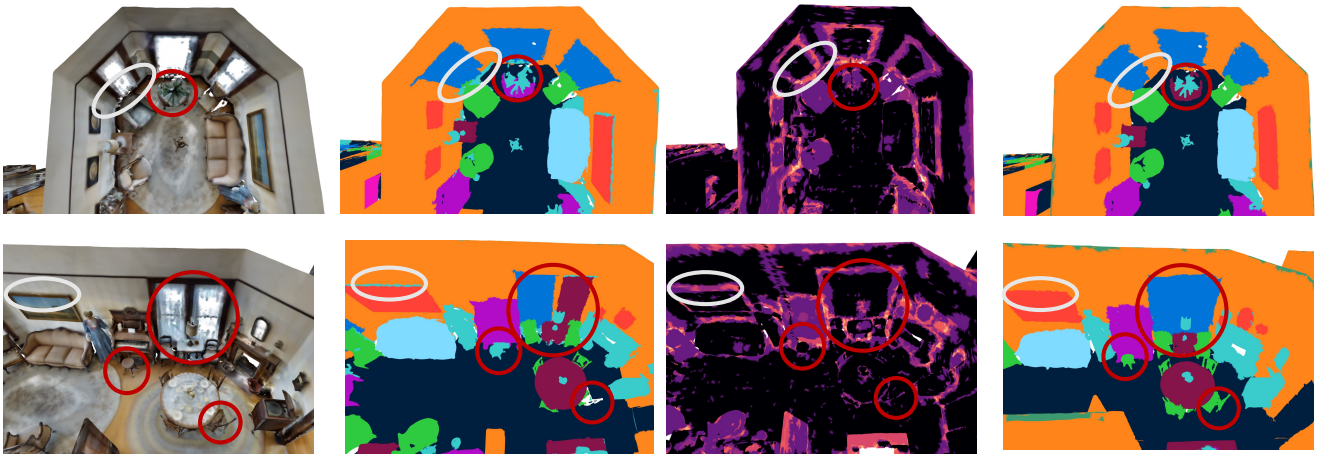

Fig. 2: Matterport 3D dataset. From left to right: RGB mesh, ground-truth provided by Matterport, Uncertainty map by multi-user integration, best results obtained with our tool. Circled in red we highlight the errors in the Matterport ground-truth (table as furniture, windows as table) avoided by labeling with our tool. Circled in white we show high uncertainty labels (e.g. 3D boundaries).

may show some mistakes, leading to biased rankings when employed to benchmark the accuracy of algorithms.

Based on these considerations, in this work we propose *Shooting Labels*, a novel tool based on Virtual Reality (VR) to ease the dense 3D semantic labeling, so as to gather 3D and 2D data endowed with semantic annotations. To the best of our knowledge, ours is the first system which allows for handling efficiently large-scale 3D semantic labeling processes, such as labeling whole city blocks. Moreover, by exploiting Virtual Reality to make the task of labeling as easy and fun as playing a video-game, our approach remarkably reduces the expertise necessary to work with 3D semantic labeling tools. The immersive experience provided by VR technologies allows the user to physically move around within the scenario she/he is willing to label and interact with objects in a natural and engaging way. The user is transported into a large virtual environment represented as 3D meshes, where surfaces can be "colored" semantically in a highly captivating way (see. Figure 1 for some in-game visualizations).

The full fledged gamification of our tool empowers a larger community to undertake this type of activity and enables the possibility to obtain much more annotated data. For this reason, our tool features a multi-player post-processing procedure wherein we integrate results of several annotators to both improve accuracy and compute a labeling uncertainty map which provides information about the reliability of the produced ground truth.

The novel contributions of this paper can thus be summarized as follows:

- To the best of our knowledge, we propose the first 3D dense labeling tool based on Virtual Reality.
- Our immersive game-style interface enables users without specific expertise to label 3D data expeditiously.
- Our tool can work with the most common 3D data structures and with large scale scenarios.
- We propose an approach to integrate multi-player results in order to extract information about the reliability of the generated ground truth. Thus, unlike existing tools, our provide both semantic labels and their associated uncertainty.
- We label a whole sequence of KITTI and an entire house of Matterport3D and release annotations dealing with both 3D data and 2D images.

Our open-source framework is based on Unity[3], Blender[4] and open3D [11]. Project page: https://github.com/pierlui92/Shooting-Labels.

II. Related Works

We review the literature concerning Semantic Segmentation datasets and the user experience associated with 3D dense annotation.

Semantic Segmentation Datasets Several datasets featuring 2D images annotated with semantic labels are available. The most popular are KITTI [5] and Cityscapes [2], which, yet, contain a relatively small number of images semantically annotated by hand. They were two of the first datasets proposed in this area (urban outdoor), so the focus was more on the data than on how to generate them. On the other hand, the Mapillary dataset [12] includes many more images, though the labeling was still performed manually image by image. The same is true for some indoor datasets, such as [13] and [14], in which, although smart graphical tools are used to produce frame-by-frame annotations, the order of magnitude of the available images is only slightly higher. Conversely, in [6] an efficient pipeline for indoor environments is proposed, which allows for scanning a room, reconstruct it in 3D and

[3]https://unity.com/
[4]https://www.blender.org/

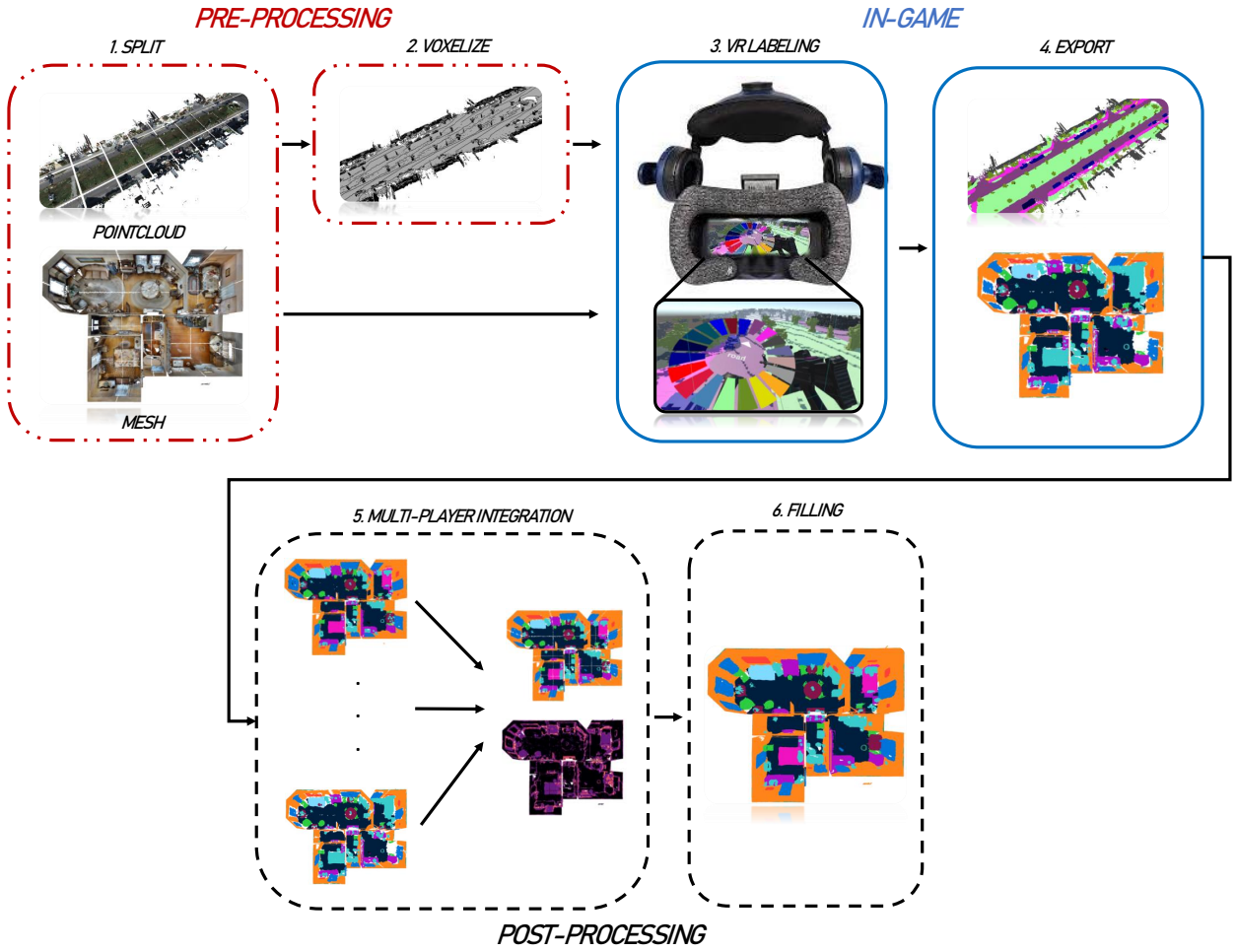

Fig. 3: The six steps of the *Shooting Labels* pipeline. 1- Splitting 3D data (a mesh or point cloud) into chunks to optimize visualization in the VR environment. 2- In case of point clouds, voxelization enables real-time rendering and reduces memory footprint 3- Semantic labeling by Virtual Reality. 4- Exporting annotated data into the original format 5- Integrating multi-player results to improve labeling and assess the reliability of the labels (optional). 6- Filling of unlabeled elements (optional).

then easily label the gathered 3D data rather than each single image. In [7] such procedure is formally extended with a projection module which, based on known camera poses, brings the 3D labels into 2D. The label projection approach was then exploited in other datasets, such as [15] and [9]. It can be observed that, leveraging on 3D reconstruction and camera tracking to facilitate labeling, may be thought of as shifting the cost of labeling each individual image toward the complexity of the requirements necessary to obtain a suitable dataset (tracked camera). This benefit is even more evident in synthetic datasets, such as [8], [16], [17], where obviously both camera tracking and 3D reconstruction are no longer external elements but inherent to the rendering engine. Recently [18]–[20] have proposed large urban outdoor 2D-3D datasets where the annotation task is performed on point clouds and semantically labeled images are attained by projection.

User Experience As for "how to generate the data", the user experience associated with 3D Dense Annotation is rarely addressed in literature. Indeed, this procedure typically involves professional software and 3D modeling experience. Some authors have expressly addressed this by proposing elegant, efficient solutions. In [21] the authors have proposed an interactive procedure by means of which the user can physically touch the scene within a classical 3D reconstruction pipeline, so as to "color" large parts of the scene by exploiting region growing techniques. In [22], instead, the authors build a physical device able to reproduce the pipeline while the user navigates the environment in Augmented Reality, using a laser pointer to identify the homogeneous areas of the scene and assign them a correct label. Differently, our proposal introduces a Virtual Reality framework to navigate within the reconstructed environments, providing the user with a series of gamification tools to "color" the world in a fast and intuitive

manner. To the best of our knowledge, ours is the first method that allows for labeling very-large-scale scenes in a short time by a VR approach.

III. VR Labeling Tool

In this section we describe the key features of our tool. *Shooting Labels* works with the most popular 3D representations, such as point cloud and meshes, which can be obtained by any kind of 3D reconstruction technique. Moreover, with our tool we can also load a 3D scene pre-labeled by any other technique (e.g. a CNN for 3D semantic segmentation) in order to refine it. As shown in Figure 3, our pipeline can be summarized into 6 main steps grouped into 3 stages, with the first stage dealing with *Pre-processing* of the input 3D Data, the second with *In-Game* labeling and the third with *Post-processing* of the labeled 3D data.

A. Pre-Processing of the Input 3D Data

Meshes and point clouds obtained by 3D reconstruction techniques typically consist of millions of vertices which can hardly be rendered in real-time in a VR environment. For this reason, with both meshes and point clouds we employ a Level Of Detail strategy to mitigate the computational demand. As illustrated in the first step of Figure 3, we split meshes and point clouds in several chunks, saving each chunk at 3 different resolutions. During VR labeling session, objects close to the player are loaded at a higher resolution than those far away. Furthermore, as point clouds cannot be managed by the Unity gaming engine, we voxelize them (see Figure 3) in order to obtain a friendly visualization both in terms of light computation and user-experience during navigation. To perform voxelization we set a discretization step and for each position of the dense 3D grid build a cube mesh if that volume contains a minimum number of points (e.g. > 5).

B. In-Game Labeling

3D meshes are loaded into the Virtual World and the user can explore and label the environment. The player can teleport or physically move around the scene to reach each portion of the environment. The following features have been implemented to enhance and simplify the user experience:

- Geometric and RGB visualization
- Unlabeled Face Visualization
- Level of Detail (LOD)
- Labeling Granularity
- Export of Final Results

Geometric and RGB Visualization To assign a semantic label to a mesh, the user paints on the geometric view of the object (Figure 1 middle picture). However, in the 3D reconstruction objects may be difficult to disambiguate without color cues. To address this, in the case of meshes, we directly visualize the RGB version of the mesh if available. As for point clouds, we noticed that coarse RGB voxelizations can lead the user to misunderstand the scene. Thus, we visualize directly the RGB point cloud, building a mesh object for each point to enable visualization of this type of data also within the Unity rendering system (Figure 1 left image). We did not employ this kind of visualization during labeling because the interaction with this type of data can be extremely slow. However, we obtain smooth rendering performances for only the visualization.

Unlabeled Face Visualization Reaching some portion of the 3D space can be hard (e.g small hidden faces), or the user might wish to visualize the progress of its labeling. Thus, we keep track of the faces labeled by the user and, at any moment, allow the user to visualize only the faces still unlabeled.

Level of Detail As already mentioned, we implement a Level Of Detail (LOD) optimization to enable real-time rendering of large-scale scenarios. For each chunk obtained by splitting the mesh we keep 3 versions at different LOD and dynamically load at high resolution only the meshes within an action range. The user can interact only with the meshes at the highest resolution, those closest to him, thereby significantly alleviating the overall computational burden.

Labeling Granularity A user may require different labeling resolution degrees so as to, e.g., colour either large surfaces or small details. Therefore, she/he can choose between a pool of different *weapons* which feature different action ranges, thereby enabling either a more precise or faster labeling. The user chooses the current label from a color palette (Figure 1, central picture) and when shooting toward a direction we color each face within the weapon action range of the first hit face. When hitting a face, in Unity we know only the hit face and we must find each face up to a range. As analyzing all faces of the scene can be extremely slow, thus impractical for real-time rendering, we search between the faces belonging only to the same object chunk.

Exporting Final Results At the end of VR Labeling phase we can save the progress or export the annotated mesh. During export chunks are merged together into the original mesh. In case the source data was a point cloud, we assign to each 3D point the label of the corresponding voxel.

C. Post-Processing of the Labeled 3D Data

Once the labeled data have been exported, the tool offers some optional post-processing step.

Multi-Player Integration The gamification process potentially enlarge the pool of possible users of our tool. Thus, we can exploit the redundancy of labeling to predict the most confident label for each face or point. Let us denote as $\mathcal{S}$ our mesh or point cloud, composed of several elements, e, i.e. faces or points respectively. For each point or face $e \in \mathcal{S}$ we define as $\mathcal{H}$ its corresponding histogram of labels, as assigned to it by different players. We can assign to each element e its most confident label by simply finding the most frequent label:

$$e = argmax(\mathcal{H}), e \in \mathcal{S} \tag{1}$$

Label Uncertainty Since the annotation process by a single user may contain errors, we might wish to assess upon the uncertainty associated with each label.

Player	Bed	Ceiling	Chair	Floor	Furniture	Object	Picture	Sofa	Table	Wall	Window	Perc.Labeled	mIoU
1	74.63	**95.19**	69.35	84.16	78.83	58.66	74.74	85.99	55.14	86.86	71.84	**97.04**	75.94
2	76.37	90.29	72.45	**88.16**	**80.87**	60.76	**77.36**	82.39	**62.20**	85.65	69.43	94.56	76.90
3	74.15	93.53	71.19	80.28	67.14	50.12	66.80	80.43	56.56	86.31	70.02	94.31	72.41
4	-	90.07	70.74	80.12	58.78	44.89	64.32	80.60	38.79	84.72	68.93	88.44	62.00
5	77.69	89.10	35.99	77.76	56.09	41.87	46.90	33.36	40.49	80.95	44.25	91.76	56.77
Integr.	**81.46**	94.80	**77.34**	87.02	79.46	**64.27**	75.35	**89.68**	57.70	**89.83**	**74.93**	94.47	**79.26**

TABLE I: Comparison between single player and multi-player results on the Matterport 3D dataset [9]. Best results in bold.

Player	Bed	Ceiling	Chair	Floor	Furniture	Object	Picture	Sofa	Table	Wall	Window	Perc.Labeled	mIoU
1	73.77	**95.18**	68.09	82.94	78.04	56.28	73.79	**85.56**	54.29	86.13	71.58	100	75.06
2	75.90	90.28	65.55	**84.69**	**78.31**	55.44	**76.00**	67.00	**58.60**	84.05	68.19	100	73.09
3	73.21	93.23	67.00	77.54	65.27	48.08	65.57	75.25	52.64	84.46	68.68	100	70.09
4	-	88.86	57.47	73.92	54.73	39.56	59.50	68.01	30.52	80.93	65.67	100	56.29
5	71.56	88.82	35.13	73.79	54.49	38.55	46.13	32.30	38.50	78.59	43.89	100	54.71
Integr.	**80.33**	94.70	**70.27**	83.00	77.90	**57.98**	74.51	82.94	53.86	**87.85**	**73.85**	100	**76.11**
Integr.0.5	80.36	94.71	70.30	83.02	77.94	57.99	74.58	82.86	53.95	87.87	73.86	100	76.13
Integr.0.65	77.97	94.85	70.15	82.13	77.84	57.24	74.71	85.28	50.65	87.91	74.78	100	75.77
Integr.0.8	75.17	94.40	68.62	81.10	77.91	55.62	73.21	81.91	47.40	87.13	70.27	100	73.89

TABLE II: Comparison between filling single users, multi-player integration and multi-player integration based on label uncertainty on the Matterport 3D dataset [9]. Best results of filling without uncertainty in bold. Best result of filling integration using uncertainty in red.

Given n annotators we can easily get the label probability distribution $\mathcal{P} = \frac{\mathcal{H}}{n}$ for each element e. From the probability distribution we can calculate its entropy:

$$\mathcal{E} = -\sum p \log p \tag{2}$$

The entropy of that distribution can be treated as the uncertainty of the labeling for that element, u_e. We can leverage this uncertainty to decide which points should be considered noisy ground truth in the annotation process. Moreover, we could exploit it to refine only the high entropy elements both manually or by means of suitable algorithms.

Filling Some users may decide to label only partially the whole scenario. Moreover, during the pre-processing we may lose information about few faces where no labels will be available. For these reasons, *Shooting Labels* provides a function for automatic filling missing elements based on their neighborhood. We define $\mathcal{S}_{unlabeled}$ the set of elements without any label assigned and $\mathcal{S}_{labeled}$ the set of elements with a label assigned such as $\mathcal{S} = \mathcal{S}_{labeled} \cup \mathcal{S}_{unlabeled}$. Given one point $s_{unlabeled} \in \mathcal{S}_{unlabeled}$, we can find its K closest elements $s_{labeled} \in \mathcal{S}_{labeled}$ and their labels, and build the histogram H of labels of its neighborhood. Then we can easily infer its label:

$$y = argmax(\mathcal{H}) \tag{3}$$

In a multi-player setting we can leverage the uncertainty information to further improve the precision and accuracy of the labeling by semantically filling also high uncertainty elements by means of their neighborhood. More precisely, given a fixed uncertainty threshold th_u, we consider unlabeled elements that have their uncertainty u_s above th_u:

$$\hat{\mathcal{S}}_{labeled} = \{s \in \mathcal{S}_{labeled} | u_s > th_u\} \tag{4}$$

$$\hat{\mathcal{S}}_{unlabeled} = \mathcal{S}_{unlabeled} \cup \hat{\mathcal{S}}_{labeled} \tag{5}$$

For each $\hat{s}_{unlabeled} \in \hat{\mathcal{S}}_{unlabeled}$ we can find its neighborhood composed of its K closest elements e_k. For each e_k we know its labels y_k and its uncertainty information u_k. Thus, we can build a weighted histogram of labels for the considered neighborhood, $\mathcal{H}_u$, collecting the votes for each label multiplied by their own uncertainty.

At this point the label of $\hat{s}_{unlabeled}$ will be $y = argmax(\mathcal{H}_u)$.

Obtaining 2D Segmentations We leverage the 3D segmentation to produce 2D segmentations of known RGB images. If a specific set of images comes along its intrinsic and extrinsic camera parameters we can seamlessly render the segmentation through the Blender render engine.

IV. Experimental Results

A. Efficiency and Accuracy of the Tool

To evaluate the efficiency and performance of our tool we tested it on the Matterport 3D dataset [10]. To perform the evaluation we considered the labeling provided by Matterport as our ground truth. Their labeling has been attained through a

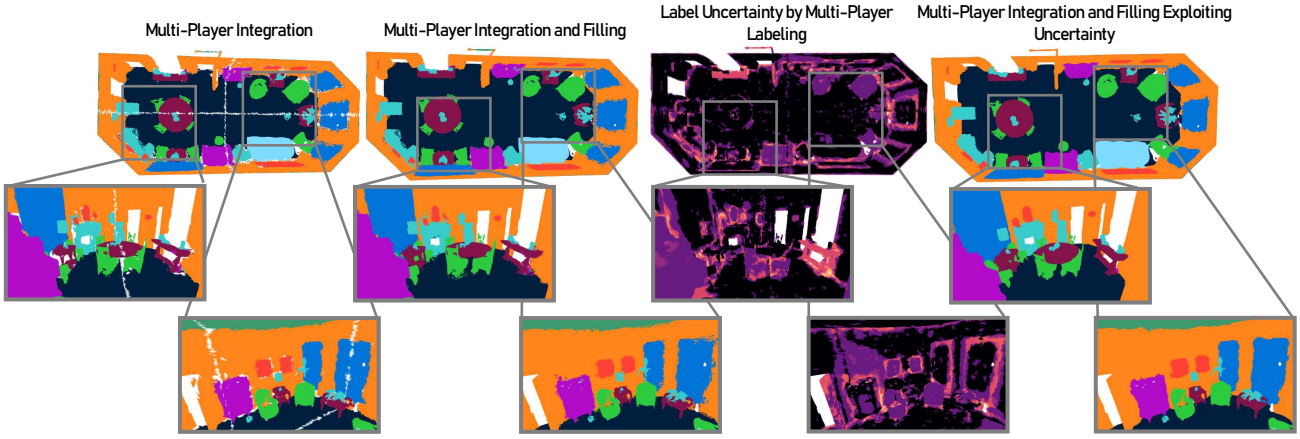

Fig. 4: Qualitative comparison of different combination of post-processing steps. From left to right: results after only the multi-player integration step; multi-player integration and filling steps without using label uncertainty; label uncertainty map; multi-player integration and uncertainty aware filling. We can notice that the fourth image has much smoother edges than the second one thanks to exploitation of the uncertainty map.

series of refinement steps based on several different tools and expertises. They first produced a coarse annotation with a first tool for planar surface labeling, then they used the ScanNet crowd-sourcing interface by Dai et al. [7] to "paint" triangles and name all object instances of the house. Finally a team of 10 expert annotators refined, fixed and verified the quality of the final labeling. In ours tests we labeled an entire Matterport house[5] made out of 6 rooms based on the 13 classes of objects defined in [23] (eigen13 categories). We exploit the mapping provided by Matterport from their labeling to eigen13 to obtain ground truths used for testing. They provide face-wise labels since ground truth are meshes. Therefore, we evaluated our results using a mean intersection over union weighted on the area of the faces:

$$IoU^c_{faces} = \frac{A_{TP}}{A_{TP} + A_{FN} + A_{FP}} \quad \text{where} \quad c \in Cl \tag{6}$$

$$mIoU_{faces} = \frac{1}{N} \sum_{c \in Cl} IoU^c_{faces} \tag{7}$$

where Cl is the set of classes, N is the total number of classes, A_{TP}, A_{FN}, A_{FP} represent the total area of the true positive, false negative and false positive faces respectively for a class c. Furthermore, we provide the percentage of area of faces annotated $A_{labeled}$ over the total area of the labeled ground truth A_{total}.

$$Perc.Area = \frac{A_{labeled}}{A_{total}}\% \tag{8}$$

Single Player and Multi-Player Integration Results We evaluated the annotation of 5 different players without any expertise in 3D modeling. The annotators are 16 to 60 years old, with little or no background in VR systems. We compared their results with the ground truth provided by Matterport. The results are shown in Table I. The average time needed for the labeling was approximately 2.5 hours. Even though there are users who achieved low labeling performances (player 4 and 5), we notice that integrating results of all players yield the best overall performances of 79.26% mIoU, surpassing the accuracy of each single user. As we wanted to analyze what are the most common errors in labeling, we inspected qualitatively each single player results noticing that most frequent errors are correlated with 3D object boundaries and ambiguous object. Therefore, we manually investigate also the GT provided by Matterport finding the same types of errors. In Figure 2, we circled in white the errors on object boundaries while in red the completely mismatching object between our labeling and the Matterport ground truth. In the second row we note that a window (Blue Label) was labeled as a Table (Red Label) in the Matterport GT while with our tool we did not encounter that error. We also found a case of an ambiguous object where a stool has been labeled as an Object (Light Blue Label) by the Matterport ground truth while as a chair by our users (Green Label). These ambiguities and errors in both labeling might be the main cause of achieving lower mIoU score in our labeling and therefore, discarding them with our uncertainty information should lead to better overall performances.

Uncertainty Map Evaluation By integrating the results of several users we computed the label uncertainty map shown in the third column of Figure 4 and Figure 2. As we wish to evaluate the quality of our labeling, we analyzed the performances of our labeled mesh at different uncertainty thresholds. In Figure 5 we show the mIoU and the Perc.Area labeled at different uncertainty thresholds. For each threshold we evaluated the mIoU only on the elements with lower uncertainty than the threshold. We see that the higher the threshold the lower the mIoU, symptom of a good uncertainty map. In Figure 2 the third column are the uncertainty maps of the

[5]Matterport House ID: `2t7WUuJeko7`

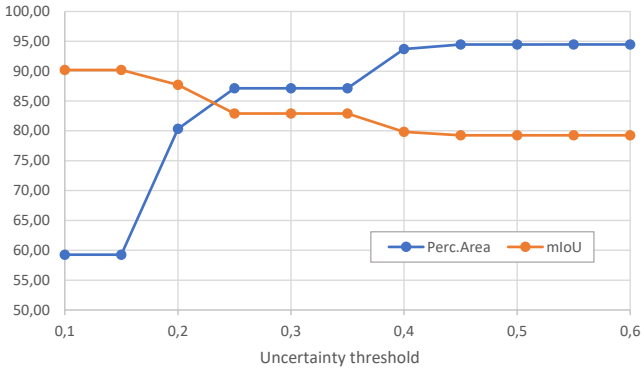

Fig. 5: Evaluation at different threshold of uncertainty. The more noisy labels we discard the higher mIoU wrt Matterport3D ground truth.

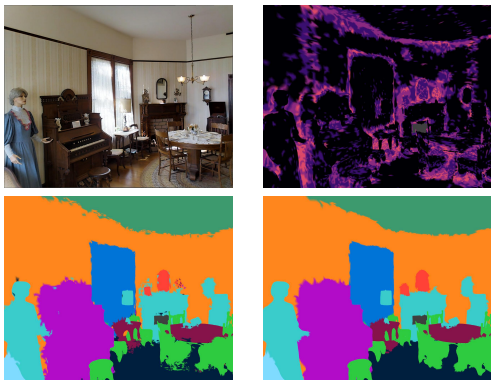

Fig. 6: Matterport3D: projection of 3D labels into 2D labels. Top Left: RGB image with known camera pose. Bottom Left: 2D labels from a semantically filled mesh without exploiting uncertainty. Top Right: 2D uncertainty map. Bottom Right: 2D labels from a semantically filled mesh by exploiting uncertainty.

labeling where warmer color represents higher uncertainty. We notice that while white circled error are always correlated to high uncertainty, red circled errors might have low uncertainty. This happens because there are object that the majority of the annotators labeled in the same way while they are labeled different in Matterport ground truth which is correlated to an error in the Matterport ground truth (Windows labeled as a Table).

Filling Results w/ or w/o Uncertainty Table II shows the results of the filling step in various settings. The first five rows report the results obtained by applying filling immediately after the single player annotation. These rows highlight that we were able to fill all the unlabeled faces obtaining Perc.Area of 100% and slightly lower performance. We can see a similar trend also in multi-player integration results where we score a 76.11% mIoU while gaining a +5.53% on the Perc.Area labeled and losing only the 3.15% in mIoU with respect to the results without filling. The last three rows report the results of filling by exploiting the uncertainty map and using different threshold levels, i.e. 0.5, 0.65 and 0.8 respectively, with the best results of 76.13% mIoU attained with threshold 0.5. The decreasing trend with higher threshold can be explained thinking that using a higher threshold corresponds with considering good a lot of uncertain elements making filling too difficult and noisy. Figure 4 depicts a qualitative comparison between filling strategies. From left to right we can see the labeled mesh before the filling, the mesh filled without uncertainty, the uncertainty map and filling exploiting uncertainty. We highlight how, though the increase in performance in uncertainty aware filling is small, the qualitative results highlight a much smoother labeling.

3D to 2D Projection Figure 6 illustrates qualitative results of the 2D labels obtained by projecting 3D labels. Given the availability of the RGB image (top left image) and its associated camera intrinsic and extrinsic parameters, we can configure the Blender rendering engine and position the virtual camera in order to obtain the 2D render of the scene. Peculiarly to our tool, we can also provide a 2D uncertainty map by projecting the 3D uncertainty map. Bottom left and right images are rendered from the filled mesh with (right) or without (left) exploiting uncertainty. We can notice that the right image has smoother edges and several blobs are less noisy. The render took place in approximately 1 second on a GTX 1080 Ti, much less than the hours needed by manual pixel-wise annotation.

Large Scale Outdoor Labeling We evaluated the effectiveness of our tool in a challenging outdoor scenario: the Kitti Odometry Dataset [5]. We used the provided 3D Lidar data of a static sequence[6], consisting of more than 1000 images equipped with ground truth camera poses. We reconstructed the point cloud, then voxelized and labeled it by our tool. Then, we were able to annotate the whole sequence in approximately 8 hours, a much shorter time compared to other non-VR tool such as [20] which needed about 51 hours for each sequence. Moreover, we could obtain the 2D semantic segmentation associated with the 1000 RGB input images in a few minutes of rendering. Figure 7 reports qualitative results dealing with our 3D labeling and examples of projected 3D labels.

V. Conclusions and Future Works

We have proposed the first 3D semantic labeling tool based on Virtual Reality (VR). Our tool exploits VR alongside with gamification to ease and expedite 3D semantic labeling of large scale scenarios and enlarge the pool of possible annotators to people without any knowledge about 3D modeling. The tool works with the most popular 3D data structures, such as meshes and point clouds. Moreover, we have shown how to integrate results from multiple users in order to achieve an overall better performance as well as uncertainty map of the labeling process. We have also demonstrated how to integrate the uncertainty map in the labeling process in order to further improve the results.

[6]Kitti Sequence `2011_09_30_drive_0020_sync`

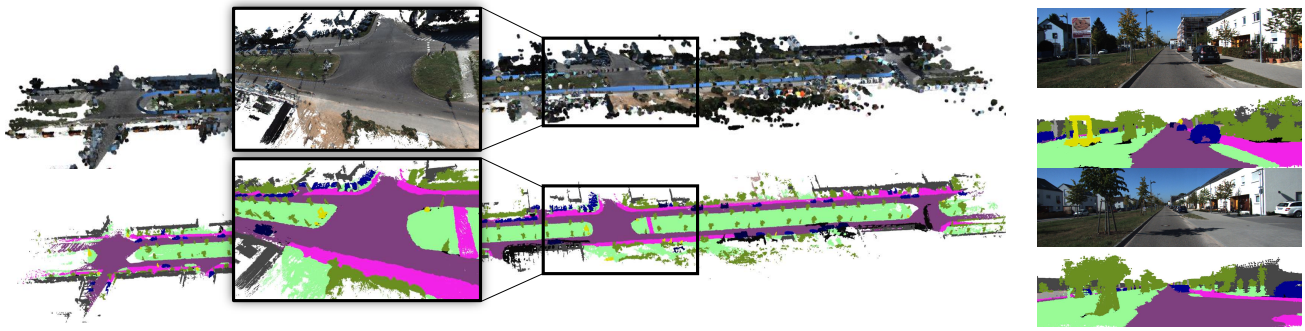

Fig. 7: 3D and 2D Labeling from a Kitti sequence. Top Left: RGB point cloud. Bottom Left: labeled point cloud obtained by using our tool. Right: RGB images and projected semantic labels

We argue that the label uncertainty information may also be leveraged while training deep neural networks for semantic segmentation, e.g. so as to weight the labels in the loss based on their associate uncertainty, as proposed in some recent works dealing with stereo vision [24]. Moreover, availability of uncertainty maps may foster the design of novel performance evaluation metrics which would take into account the uncertainty of labels.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[4] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[6] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "Scenenn: A scene meshes dataset with annotations," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 92–101.

[7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[8] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2678–2687.

[9] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[10] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 667–676.

[11] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[12] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4990–4999.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[14] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[15] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *arXiv preprint arXiv:1702.01105*, 2017.

[16] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4077–4085.

[17] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," in *British Machine Vision Conference (BMVC)*, 2018.

[18] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[19] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, 2017, pp. 91–98.

[20] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "A dataset for semantic segmentation of point cloud sequences," *arXiv preprint arXiv:1904.01416*, 2019.

[21] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr, "Semanticpaint: Interactive 3d labeling and learning at your fingertips," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 5, p. 154, 2015.

[22] O. Miksik, V. Vineet, M. Lidegaard, R. Prasaath, M. Nießner, S. Golodetz, S. L. Hicks, P. Pérez, S. Izadi, and P. H. Torr, "The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3317–3326.

[23] C. Couprie, C. Farabet, L. Najman, and Y. Lecun, "Indoor semantic segmentation using depth information," in *International Conference on Learning Representations (ICLR2013), April 2013*, 2013.

[24] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised adaptation for deep stereo," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1605–1613.