Semantic Object Reconstruction via Casual Handheld Scanning

RUIZHEN HU, Shenzhen University
CHENG WEN, Shenzhen University
OLIVER VAN KAICK, Carleton University
LUANMIN CHEN, Shenzhen University
DI LIN, Shenzhen University
DANIEL COHEN-OR, Shenzhen University and Tel Aviv University
HUI HUANG*, Shenzhen University

We introduce a learning-based method to reconstruct objects acquired in a casual handheld scanning setting with a depth camera. Our method is based on two core components. First, a deep network that provides a semantic segmentation and labeling of the frames of an input RGBD sequence. Second, an alignment and reconstruction method that employs the semantic labeling to reconstruct the acquired object from the frames. We demonstrate that the use of a semantic labeling improves the reconstructions of the objects, when compared to methods that use only the depth information of the frames. Moreover, since training a deep network requires a large amount of labeled data, a key contribution of our work is an *active self-learning framework* to simplify the creation of the training data. Specifically, we iteratively predict the labeling of frames with the neural network, reconstruct the object from the labeled frames, and evaluate the confidence of the labeling, to incrementally train the neural network while requiring only a small amount of user-provided annotations. We show that this method enables the creation of data for training a neural network with high accuracy, while requiring only little manual effort.

CCS Concepts: • **Computing methodologies** → **Shape modeling**;

Additional Key Words and Phrases: 3D scanning, object registration, semantic reconstruction, active learning

ACM Reference Format:
Ruizhen Hu, Cheng Wen, Oliver van Kaick, Luanmin Chen, Di Lin, Daniel Cohen-Or, and Hui Huang. 2018. Semantic Object Reconstruction via Casual Handheld Scanning. *ACM Trans. Graph.* 37, 6, Article 219 (November 2018), 12 pages. https://doi.org/10.1145/3272127.3275024

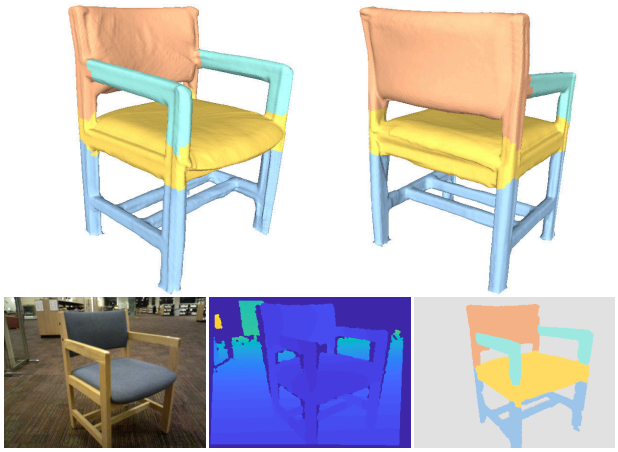

Fig. 1. A semantic reconstruction of an object obtained with our method (top), using a semantic labeling of frames (one example shown in the bottom-right) computed for RGB and depth input images (bottom-left and middle).

1 INTRODUCTION

In recent years, we have witnessed a huge advance in the development of scanning technologies. Yet, the acquisition and reconstruction of three-dimensional content remains a challenging task, especially for creating a massive amount of high-quality models. A simple approach to scan a 3D object is to work in an uncalibrated setting, where the scanner is hand-held and casually rotated around the object. However, it is often difficult to obtain an accurate reconstruction in such a casual setting, since the scans are not accurately tracked during the acquisition, and noise in the data may easily lead to incorrect registration of the scans.

Due to these challenges, the advances in scanning technologies have been accompanied by research to improve the quality of scan reconstruction. Methods focused on simultaneous localization and mapping (SLAM) typically use tracking methods to perform the alignment of scanned sequences [Fuentes-Pacheco et al. 2015; Thrun 2002]. Thus, these methods rely mainly on the geometric similarity and temporal coherence between subsequent frames, which may not be reliable in the casual setting described above. A few methods combine segmentation and registration to improve the alignment based on semantic information. However, these methods are either trained on a sparse set of examples with class-specific priors [Häne et al. 2016], or assume that the registration is sufficiently reliable with objects being abstracted as bounding boxes [Xiao et al. 2013], which can be unreliable for objects with detailed geometry. Moreover, all of these methods are designed to reconstruct 3D scenes based on object labels instead of a 3D object based on part labels.

*Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Authors' addresses: Ruizhen Hu, College of Computer Science & Software Engineering, Shenzhen University, ruizhen.hu@gmail.com; Cheng Wen, Shenzhen University; Oliver van Kaick, Carleton University; Luanmin Chen, Shenzhen University; Di Lin, Shenzhen University; Daniel Cohen-Or, Shenzhen University and Tel Aviv University; Hui Huang, Shenzhen University.

0730-0301/2018/11-ART219 $15.00
https://doi.org/10.1145/3272127.3275024

In this paper, we present a learning-based method to reconstruct a 3D object, which consists of segmenting, registering and reconstructing RGBD frame sequences acquired in a casual handheld scanning setting. The premise of our work is that a semantic segmentation of RGBD frames leads to a significant improvement of their registration and, as a consequence, improves the reconstruction of the object. Shapes can have many local patches with similar geometry, which may lead to ambiguities in the matching. Thus, the advantage of a semantic labeling is to break the ambiguity in such cases, indicating that similar geometry should only be matched if belonging to regions with the same semantics. With the use of a semantic labeling, the method is more accurate than traditional reconstruction techniques, and yields as a byproduct a segmentation of the reconstructed object; see Figure 1 for an example result. Our method is based on two core components: (i) A deep network that labels the pixels of each frame into different semantic parts, and (ii) A reconstruction method that employs the semantic labeling to register the RGBD frames and reconstruct a 3D object.

Training a deep network for segmentation requires a large amount of training data in the form of multiple labeled scanning sequences. Thus, a key contribution of our work is an *active self-learning* framework that simplifies the creation of training data, relieving users from considerable manual work (Figure 2). The *active learning* is an iterative process where the frames that should be labeled by the users are strategically selected to minimize the amount of required annotations. Specifically, the user annotates a few frames of multiple sequences, which are used to train a deep network that predicts a semantic segmentation and labeling of the remaining frames of all the sequences. The process is then repeated by asking users to annotate frames from sequences labeled with low confidence, and retraining the network, until all the sequences are labeled with high confidence or the method reached the maximum number of iterations. To estimate the labeling confidence of a sequence, we reconstruct 3D models of the objects acquired in each sequence from all the labeled frames, while fusing the frame labels into consensus segmentations for each model. Then, we compute the agreement between the labels of each frame and the segmentations of the 3D models, which provides a measure of the labeling consistency of each frame. Sequences are deemed to be of high-confidence if most of their frames have high consistency.

Since requesting user input is the laborious part of the active learning, which we seek to minimize, we introduce a *self-learning* approach that exploits the use high-confidence sequences as much as possible. Specifically, after retraining the network with user annotations, several sequences are labeled with high confidence based on the reconstruction consensus. Thus, we automatically sample frames from these sequences to serve as new training data to retrain the network. This self-learning cycle is repeated until no new high-confidence sequences are generated. Then, the active learning continues and the user is asked to annotate a new batch of frames from low-confidence sequences. The result of the active learning is a trained network that is able to label frames accurately without any user assistance.

Moreover, to obtain a reconstructed model, another key contribution of our work is a rigid registration method based on the semantic labeling of the frames. We iteratively build a 3D model by registering frames to the model, according to the correspondence between regions of the frames and the model with the same label. Note that this semantic registration is a crucial component for the estimation of the confidence of the labeling used in the active self-learning.

Although there has been recent work [Häne et al. 2016; McCormac et al. 2017; Salas-Moreno et al. 2013; Xiao et al. 2013] and datasets [Dai et al. 2017] aimed at semantically labeling indoor scenes at the object level, our work is the first to create a *dense* set of labeled training sequences in the context of single object reconstruction with part labels. The training data is obtained with an active learning framework that lowers the amount of manual work needed to create such dense sequences of data. With the use of our trained deep network, we are able to aggregate information coming from multiple training sequences to label new sets of frames and reconstruct labeled 3D objects with higher accuracy.

An important message of our work is that semantic segmentation improves the reconstruction of scanned sequences. We demonstrate this point by evaluating our method in terms of both segmentation and reconstruction quality. We show segmentation results of the RGBD scans and reconstructed models using our method. Moreover, we compare the results obtained with our semantic reconstruction method to results obtained with a standard method that considers mainly the depth information of the frames.

2 RELATED WORK

We organize the discussion of related works into three general topics: reconstruction, segmentation, and their combination.

2.1 Object and scene reconstruction

The traditional pipeline for object reconstruction from multiple scanned views involves the alignment of the scans followed by the reconstruction of a surface, e.g., via fitting an implicit function. The alignment step can be performed by matching feature points according to descriptors, or by performing an alignment with local techniques such as iterative closest points (ICP) or more global methods such as transformation sampling and voting [van Kaick et al. 2011]. These techniques require a sufficient amount of overlap and similarity between the scans, which can be achieved usually only in a controlled setting.

One group of recent approaches for object reconstruction create models by reusing and fitting parts from existing shapes. Xu et al. [2011] fit a 3D model to an image by taking parts from a dataset of shapes. Shen et al. [2012] fit parts from a dataset to reconstruct an object given in a single RGBD image, where the dataset allows to handle scans with considerable missing data. In addition, Huang et al. [2015] reconstruct objects from single views with an analysis that jointly considers entire datasets of images and 3D shapes. Xu et al. [2016] and Lin et al. [2018] focus on recovering functional mechanical assemblies. These works assume the availability of parts that are similar to their projections in the 2D frames. Thus, the fidelity of the 3D reconstruction is bounded by the density of the shape database. In practice, the reconstructed 3D models only approximate the original objects.

Moreover, another group of methods aim to reconstruct entire scenes from multiple scans. In the field of robotic mapping, simultaneous localization and mapping (SLAM) is one of the main approaches used to guide a robot through an environment. SLAM techniques typically involve the acquisition and reconstruction of the 3D environment sensed by the autonomous agent [Fuentes-Pacheco et al. 2015; Thrun 2002]. The reconstruction is obtained by aligning multiple scanned views and fusing them into a single model of the environment. Recent work derives the views from a video sequence of RGBD scans [Chen et al. 2015; Morell-Gimenez et al. 2014]. Notable examples of methods working on this type of input reconstruct the scene with live tracking of depth and normals [Newcombe et al. 2011], alignment of depth and color maps [Kähler et al. 2015; Nießner et al. 2013; Whelan et al. 2015], or alignment of primitives larger than points, such as scene fragments [Choi et al. 2015]. A few works also leverage tracking for scanning of objects with deformable or inaccessible parts [Dou et al. 2015; Yan et al. 2014].

Note that these works use mainly geometry tracking and alignment to guide the reconstruction. Additional structural information or semantics about the input are not extracted. The goal of our work is to use a semantic segmentation of an object and labeling of its parts to provide additional cues for reconstruction.

2.2 Object and scene segmentation

Our method generates a semantic segmentation of the reconstructed shape as a byproduct of the analysis. Numerous methods have been developed to semantically segment 3D shapes [Shamir 2008]. In recent years, methods based on learning have also been introduced for segmenting shapes, including supervised [Guo et al. 2015; Kalogerakis et al. 2010], unsupervised [Shu et al. 2016; Sidi et al. 2011], and semi-supervised approaches [Wang et al. 2012; Yi et al. 2016].

The offline training stage of our approach is based on an active learning framework. However, unlike in the works of Wang et al. [2012] and Yi et al. [2016], our input is not a complete reconstructed model, but a sequence of RGBD scans. Recently, deep learning has been used to segment 3D models, assigning semantic labels to mesh triangles [Guo et al. 2015; Shu et al. 2016]. These methods also assume clean input 3D models, while we use a deep network to label portions of RGBD frames and then reconstruct the models from the frame sequences.

Our method also shares similarities with the work of Wang et al. [2012] and Fish et al. [2016] in using a projective analysis for 3D segmentation. Wang et al. transfer labels from available 2D images by selecting and back-projecting the inferred labels onto a 3D shape. Fish et al. propagate 3D semantic labels via corresponding 2D projections and integrate the labels in the target 3D model. In our work, the integration of 2D segmentations into a 3D model is performed in an uncalibrated setting, where the views appear in a sequence of RGBD frames.

Furthermore, there is a large body of work addressing the segmentation of video sequences. Recent methods in this area perform the segmentation of entire scenes acquired with RGBD videos with user assistance. For example, Wong et al. [2015] introduce an interactive system to facilitate the annotation of raw RGBD images. The system predicts segmentations and labels of objects, and the user mainly needs to refine them with scribbles and select among the hypotheses suggested for each object. Differently from our approach, these methods focus on providing automatic or semi-automatic solutions, and do not perform reconstruction combined with segmentation.

2.3 Reconstruction + segmentation

A more specialized class of methods combine both segmentation and reconstruction into a single approach, especially to enhance the reconstruction step with semantic information given by the segmentation. A common approach is to first learn object classifiers from isolated objects or manually-labeled scans, and then apply the classifiers to a scan to segment and label the objects that appear in the scene. Kim et al. [2012] first carefully scan objects of interest that are expected to appear in the acquired scenes, and then learn structural models to detect these objects. Similarly, Salas-Moreno et al. [2013] use a database of carefully scanned objects for adding semantics to the scanned scenes. Nan et al. [2012] and Shao et al. [2012] learn classifiers from examples of individual objects, and then apply the classifiers to label input scans. Both approaches populate the scans with models from a database to create a clean output, since the sparse number of scans used by these methods does not allow to fully reconstruct the scanned geometry. Stückler et al. [2015] segment RGBD sequences with random forest classifiers, and then combine the segmented frames into a 3D semantic map with camera pose localization and label fusion. The segmentation and registration are performed independently. McCormac et al. [2017] use a convolutional neural network trained on segmented images to label scanned frames. Kundu et al. [2014] combine semantic cues with a CRF to infer a volumetric semantic map of a scene, while Rünz and Agapito [2017] employ both motion and semantic cues to segment RGBD sequences at the scene-level.

The methods more closely related to our work enable more interaction between the segmentation and reconstruction components. Xiao et al. [2013] proposed an interactive tool for constructing a labeled dataset of scanned 3D scenes. Their method performs a joint optimization where the segmentation and reconstruction steps benefit from each other. The reconstruction is performed with a structure-from-motion method that uses the segment labels to improve the correspondences, while the labels of segmented frames are propagated to corresponding views. However, it is up to the user to traverse the entire video and locate segmentation errors that need to be manually corrected. Moreover, to incorporate object labels into the reconstruction error, their method abstracts each object with a bounding box of fixed size based on an object prior, and optimizes the camera pose so that each object is always inside the predicted box. For object parts with more detailed geometric and structural variations, such an approximation is not reliable.

Moreover, Häne et al. [2016] also optimize segmentation and reconstruction simultaneously, but their method applies object classifiers automatically, without incorporating any user input to correct potentially mislabeled objects. Their method considers a set of class-specific geometric priors, while in our case, the input is a RGBD sequence of object scans. Besides, the methods of Xiao et al. [2013] and Häne et al. [2016] label scenes with object labels, while we consider part labels for the reconstruction of single objects.

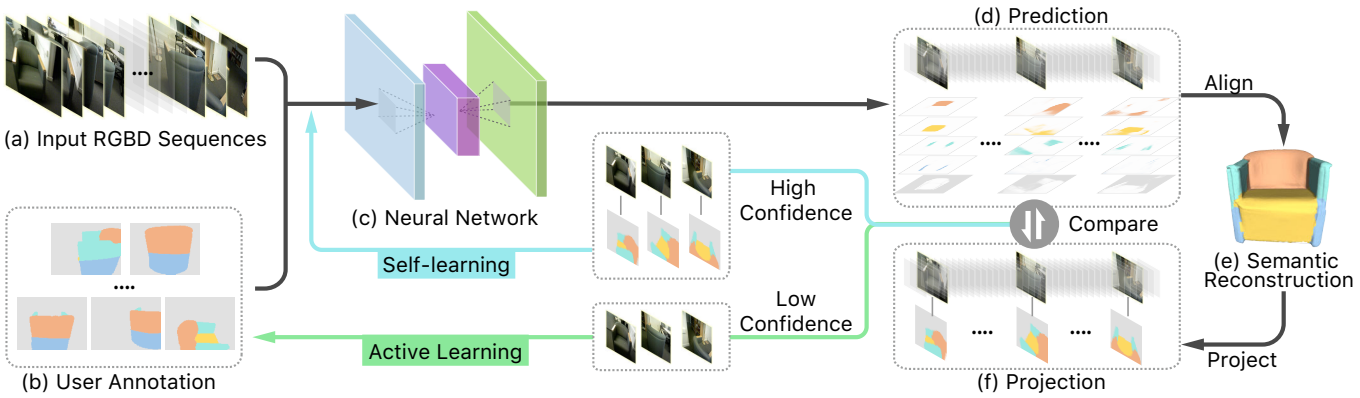

Fig. 2. Overview of our active self-learning method for object reconstruction. We learn how to segment and label a sequence of RGBD frames (a), to improve the quality of object reconstruction (e). Specifically, we employ an *active self-learning* approach to create the necessary data for the learning while involving minimal user effort. The active learning asks for user input on strategically-selected frames (green arrow) and then invokes a *self-learning* component on the annotated frames. The self-learning is an automatic learning approach consisting of cycles of prediction, reconstruction, and confidence estimation for creating additional training data from the remaining frames in the sequence (black + blue arrows). Please refer to Section 3 for details on these steps.

3 OVERVIEW

Our goal is to train a neural network to semantically segment and label RGBD sequences of a single object, and use the labeling to obtain an improved frame alignment and consequently improved object reconstruction. The training data is composed of multiple RGBD sequences for different objects of the same category, where the individual frames are segmented and labeled. To create this training data with minimal user involvement, we introduce an active self-learning framework, illustrated in Figure 2.

3.1 Active self-learning

During the active learning process, the user is asked to annotate a sparse set of frames with labels, denoted as (a) and (b) in Figure 2, which are used to train a deep network (c) that provides a semantic segmentation and labeling of all the frames (d). The labeled frames are then aligned and fused together to reconstruct a 3D model (e). During the fusion, we also transfer the labels from all the frames to the model with a voting scheme. We then estimate the consistency of the labeling of each frame. The consistency is derived from the agreement between the labels predicted by the network (d) and the labels obtained by projecting labels from the model back onto the frame via the frame's alignment (f). A sequence is considered to be of high-confidence if it does not contain any subsequence longer than 30 frames with all the frames being of low-consistency.

Once all the sequences have been processed, we select frames from the high-confidence sequences as additional training data to fine-tune the network and improve its labeling accuracy (blue arrow in the figure). This *self-learning* process involving prediction, reconstruction, and confidence estimation is iterated until the confidence of the reconstructions cannot be further improved.

Then, the user is asked to annotate frames sampled from the sequences that remained labeled with low confidence (green arrow), and we repeat the self-learning process. We repeat the active learning combined with self-learning until all the sequences are reconstructed with high confidence or we reached a maximum number of iterations. We demonstrate that this approach leads to the creation of high-quality training data to successfully train the deep network while requiring minimum manual labeling, since the user is only asked to annotate frames from sequences with persistently low confidence.

3.2 Semantic reconstruction

To reconstruct a 3D model, we iteratively align the frames to an evolving model based on their semantic labeling. The model is represented as volume. We start with an empty volume and project the first frame onto the model based on the identity transformation. Next, we align and project each subsequent frame to the model. For each frame, sets of pixels in the frame with a common label are matched to sets of voxels in the model with the same label. Note that we only consider pixels and voxels with high labeling confidence in the matching.

The matching determines a candidate transformation that aligns the frame to the model. Since each label determines a different transformation, we combine all the candidates into a single transformation by optimizing for a transformation that best approximates all of the candidates. The optimized transformation is used to align the frame to the model, which is then projected onto the volume to update the model. The labeling of pixels of the frame is also accumulated onto the labeling of voxels of the model.

After obtaining the reconstructed model and its labeling, we perform a refinement of the labeling. Specifically, we divide the reconstructed model into super-voxels and eliminate super-voxels corresponding to the background. We then optimize a graph-cut energy to refine the granularity of the labeling. We show that the use of semantic labels and the label refinement improve the quality of the registration and reconstruction, when compared to approaches oblivious to the semantics of the objects.

4 NETWORK FOR FRAME SEGMENTATION

Our neural network belongs to the category of fully convolutional networks (FCNs) that has been successfully applied to perform regression of dense label maps for semantic segmentation [Long et al. 2015]. We model our FCN following the architecture of ResNet-101 [He et al. 2016]. However, our network is modified to accept frames with depth information as input, since the original ResNet-101 is based on RGB images. Thus, similarly to Shen et al. [2016], we append a depth channel to each parametric kernel of the first convolutional layer of the network. We train the network with the back-propagation algorithm using stochastic gradient descent.

During the learning phase, the input to the network is a set of RGBD images and their segmentations with part labels. Since the amount of training data in the first iterations of the method can be quite small, we pre-train the network on existing datasets. Given that there are no large datasets of segmented RGBD images available, we pre-train our network with the dataset of object scans provided by Choi et al. [2016] for object classification from RGBD frames, where we are given a set of training images $\{I_i\}$ along with their object labels $\{y_i\}$, e.g., chair, bench, etc. Although the problem of object classification is distinct from image segmentation, we remark that a similar approach has been successfully applied to fine-tune RGB part segmentation networks [Tsogkas et al. 2015; Xia et al. 2015]. Features learned for object classification tend to also be relevant for part segmentation, as object classification often depends on the appearance of individual object parts. Thus, the pre-trained features for object classification form a useful prior for further learning a network for segmenting RGBD frames.

In more detail, we pre-train the network to find the set of network parameters W_s that minimize the *object classification loss*:

$$\text{Loss}(\{I_i\},\{y_i\},W_s) = -\frac{1}{N}\sum_{i=1}^{N}\log P(y_i|I_i,W_s), \tag{1}$$

where N is the number of training images, and $P(y_i|I_i,W_s)$ is the probability of image I_i having the ground-truth object label y_i according to the network.

After the pre-training, we fine-tune the network parameters W_s using the segmented and labeled RGBD images obtained during the active self-learning. The fine-tuning objective is to minimize the *pixel-level classification loss* defined as:

$$\text{Loss}(I,m,W_s) = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|I_i|}\sum_{j=1}^{|I_i|}\log P(m_{ij}|I_i,p_{ij},W_s), \tag{2}$$

where I_i is an input image, p_{ij} is the j-th pixel of I_i, m_{ij} is the ground-truth part label of p_{ij}, $|I_i|$ is the number of pixels in I_i, and $P(m_{ij}|I_i,p_{ij},W_s)$ is the probability of $p_{ij} \in I_i$ having the ground-truth label m_{ij}, according to the network. We minimize the sum of pixel-wise errors for all pixels coming from all training images.

In the testing phase, for each pixel p of a given image I', we use the learned network parameters to compute the label probabilities of the pixel:

$$P(m_k|p) = P(m_k|I',p,W_s), \text{ for all } k \in \{1,\ldots,K\}, \tag{3}$$

where K is the number of possible labels, and m_k is the k-th label.

5 SEMANTIC RECONSTRUCTION

5.1 Improved reconstruction with semantic information

Our reconstruction method iteratively builds a 3D model $\mathcal{M}_t$ from a sequence of depth frames $\{\mathcal{F}_1,\ldots,\mathcal{F}_t\}$. The model is maintained as a volumetric representation, which is accessed with a hash function for efficiency [Nießner et al. 2013]. Specifically, we use a volumetric, truncated signed distance function (TSDF) representation, where we also store the label distribution for each voxel. We register the frames to this volume by mapping the depth maps to the voxels and updating the TSDF values with the corresponding values from the depth map, as in the method of Nießner et al. [2013].

We start with an empty volume where the TSDF is zero and register the first frame to the volume with the identity transformation. An intermediate iteration of our method then consists in registering a frame $\mathcal{F}_t$ to an existing model $\mathcal{M}_{t-1}$, built from previously registered frames $\{\mathcal{F}_1,\ldots,\mathcal{F}_{t-1}\}$, to obtain an updated model $\mathcal{M}_t$. For the registration, we use a method similar to the sensor pose estimation method of Newcombe et al. [2011], based on an objective function involving the depth information of the frames. However, we extend this method to also consider the semantic labeling of frames. The pixels of $\mathcal{F}_t$ and voxels of $\mathcal{M}_{t-1}$ are partitioned into semantic groups, which are then mapped to each other and used to derive a transformation for registering $\mathcal{F}_t$ to $\mathcal{M}_{t-1}$.

First, we group pixels in $\mathcal{F}_t$ into semantic sets S_i, one for each possible part label i. In each set, we keep only the pixels with high labeling confidence. A pixel is defined as having high confidence if the information entropy $H(P)$ of the label probability distribution P of the pixel is below a threshold $\theta = 0.15$, where the information entropy is computed in the usual manner:

$$H(P) = -\sum_{i=1}^{K} P_i \log P_i, \tag{4}$$

with P_i being the probability of the pixel having label i. Similarly, we group the voxels of $\mathcal{M}_{t-1}$ into semantic groups S'_i, which contain only high confidence voxels.

Next, we compute candidate transformations for the semantic sets by aligning each set S_i in $\mathcal{F}_t$ to its corresponding set S'_i in $\mathcal{M}_{t-1}$ with the Iterative Closest Points (ICP) method, which provides a rigid transformation T_i that aligns the two sets. Then, we obtain the optimal transformation between $\mathcal{F}_t$ and $\mathcal{M}_{t-1}$ with an optimization involving all candidate transformations.

Given the set of transformations $\{T_i\}$ for all part labels, our goal is to find a global transformation T that best combines all the transformations in $\{T_i\}$. Since the part labels are obtained by the network prediction, and different parts may have different geometric properties, the transformations corresponding to different semantic sets should have different importance in the optimization. Towards this goal, we define a weight w_i for each set S_i as:

$$w_i = \text{conf}_i + \text{size}_i + \text{var}_i, \tag{5}$$

where conf_i is the average confidence in the prediction of the label of S_i, given by the average of $H(P)$ for all pixels in S_i, size_i is the percentage of pixels assigned with the label of S_i within $\mathcal{F}_t$, and var_i is the variation in the angles between each pair of normal vectors of the voxels of the set S'_i corresponding to S_i in the reconstructed

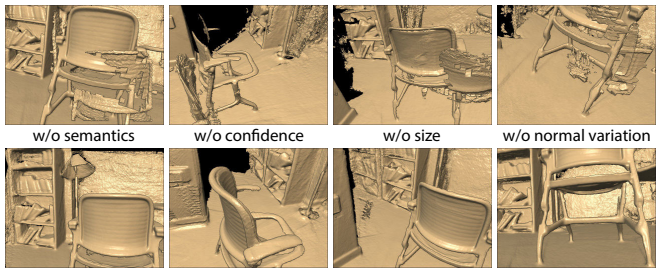

Fig. 3. Effect of different terms of the reconstruction objective. Top row: result without specified term. Bottom row: result obtained with our full method. Note the misalignments present in the top row compared to bottom.

model. Thus, the weight captures the idea that sets with higher labeling confidence, larger size, and more variation in the normals should influence more the global transformation. The rationale for preferring high normal variation is that pixels with small normal variation tend to provide more mismatches, e.g., the matching for pixels on a flat table top is ambiguous, as adding a translation before mapping any such pixel results in the same loss value.

With the weight defined per set, the global transformation T^* can be computed by solving the following optimization problem:

$$T^* = \underset{T}{\operatorname{argmin}} \sum_i \sum_j w_i ||T\, p_{i,j} - T_i\, p_{i,j}||_2^2, \tag{6}$$

where $p_{i,j}$ is the j-th pixel of set S_i. The objective states that the optimal transformation T^* minimizes the weighted alignment distance for all the sets. We use the Gauss-Newton method to minimize this objective, where we iteratively linearize the objective function and solve a system of equations [Kerl et al. 2013]. We constrain T^* and each $\{T_i\}$ to be rigid transformations, composed only of translations and rotations. We then align $\mathcal{F}_t$ to $\mathcal{M}_{t-1}$ with T^*, and integrate the frame with the model, yielding a new model $\mathcal{M}_t$.

Figure 3 shows the effect that different terms of the reconstruction objective have in the results. We compare the results obtained with our full reconstruction objective to four different alternatives: (i) Traditional ICP without semantic information; (ii) Semantic ICP without the confidence term; (iii) Semantic ICP without the segment size term; and (iv) Semantic ICP without the normal variation term. We observe that the lack of each term causes misalignments in the reconstruction for different examples, while our method that includes the four terms, especially the use of semantic labels, consistently provides good reconstruction results.

After the alignment, we update the label probability distribution for each voxel in $\mathcal{M}_t$ by accumulating the label distribution of the corresponding pixel in $\mathcal{F}_t$:

$$P_t^{\mathcal{M}} = \frac{(t-1) \times P_{t-1}^{\mathcal{M}} + P_t^{\mathcal{F}}}{t}, \tag{7}$$

where $P_t^{\mathcal{M}}$ is the label distribution of a voxel in the model at iteration t, while $P_t^{\mathcal{F}}$ is the label distribution for the corresponding pixel in the frame t. The updated label probability distribution of each voxel is then normalized so that the sum of all entries is 1.

5.2 Background removal and label refinement

Although we only consider semantic parts during the registration, the background is also labeled and stored in the volumetric representation for the reconstructed 3D model. The background consists of any data not related to the object being acquired, e.g., walls behind the scanned object. To obtain the final 3D object, we remove the background and keep only the object voxels. However, simply removing voxels where the background label has maximal confidence may introduce holes in the final 3D model due to the accumulated uncertainty from the label prediction. Thus, to remove the background in a robust manner, we employ an adaptive version of the method of Papon et al. [2013] to group the voxels into super-voxels with high labeling confidence. This method incorporates 3D relationships between voxels into a clustering algorithm to prevent super-voxels from overflowing semantic object boundaries. After obtaining the super-voxels, we remove background super-voxels not sufficiently surrounded by part super-voxels.

More specifically, the method of Papon et al. [2013] requires a resolution parameter, which controls the size of the super-voxels generated. In our label refinement, we start by specifying a large super-voxel size, and generate a first set of super-voxels. Within each super-voxel, only the voxels with high-confidence labeling are taken into consideration when determining which super-voxels to remove. Since our goal is to remove the background, we do not need to distinguish different semantic labels. Thus, we divide voxels into two types, background voxels and part voxels, by checking the label with maximal probability. A super-voxel is considered to be a part/background super-voxel if 85% of its high-confidence voxels are part/background voxels. Supervoxels that do not satisfy these criteria are set as "undefined" and are not considered in further computations. A background super-voxel is then removed if less than half of its neighboring super-voxels are part super-voxels. After removing these background super-voxels, we keep the connected component that has the maximal number of part voxels. We repeat the process iteratively. At each iteration, we decrease the super-voxel size parameter to group the voxels from the connected component of the previous iteration into finer super-voxels. We end the iterations when there are only part super-voxels, or when the super-voxel size reached a given threshold. In our experiments, we set the initial super-voxel size as $n = 200$ and stop the process when $n < 10$, dividing n by 2 at each iteration. Figure 4(a) shows an example of the iterative background removal process.

During this process, we partition the model into super-voxels, where each super-voxel can be associated with an average part label distribution. Based on these distributions, we use the graph-cuts method [Boykov and Kolmogorov 2004] to smooth the labeling of the super-voxels. Specifically, we define a graph where each super-voxel is a node that is connected to its neighboring super-voxels. The data term for labeling a node is based on the label distribution of the corresponding super-voxel, where we translate the probability values into energy costs. The smoothness term simply follows the Potts model where the cost for different labels is 1. Only part labels are used during the refinement. Figure 4(b) shows an example of the refined labeling of an object.

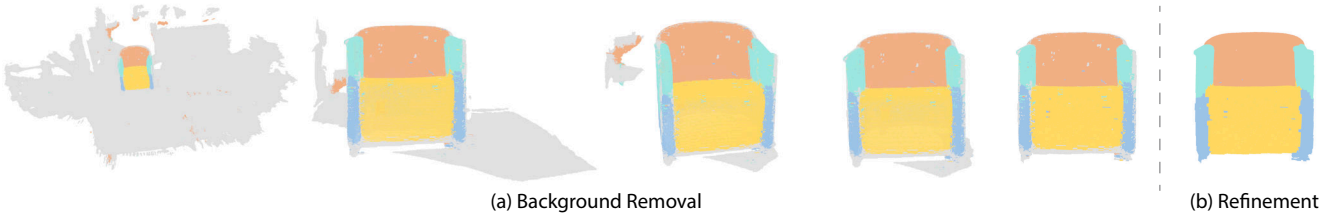

Fig. 4. Background removal and label refinement: (a) We remove the background of a reconstruction with an iterative process based on super-voxel grouping (5 iterations shown). (b) We then refine the labeling of voxels with a graph cuts method that smoothes the labels and fills gaps.

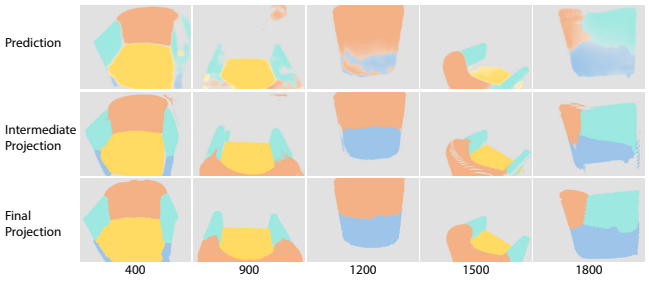

Fig. 5. Improvement of the frame labeling during the active learning: prediction given by the neural network for each frame, labels fused on the 3D model projected back onto the frames (intermediate projection), and labels projected back after refinement. Note how the quality of the labeling is improved after each step. The frame index is shown in the bottom row.

6 ACTIVE SELF-LEARNING COMPONENTS

After background removal and label refinement, we map the fused voxel labels back to the input frames. Figure 5 shows how the initial prediction of frames is improved after fusion, especially with label refinement, since the alignment and fusion provide a form of consensus of the labelings. Thus, the difference between the initial predictions and fused labels guides our active self-learning.

6.1 Confidence estimation and frame selection

To guide the active self-learning, we define a measure of label *consistency* of a frame. In an accurate reconstruction, the label prediction from the network and the labels projected from the reconstructed 3D model back onto the frame would have high *consistency*. For inaccurate reconstructions, there would be a large *inconsistency* between the prediction and projection, especially after the label refinement with graph cuts when poorly reconstructed regions would be deleted from the model.

To compute the label consistency of a frame, since each pixel is associated with a label probability distribution, we first assign the label with maximal probability to each pixel. Then, the consistency between the two labelings is defined as the percentage of pixels with the same labels on both labelings. In our work, a frame is considered to be labeled with high-consistency if the agreement of labels between its predicted and projected labelings is above 0.8. A sequence is considered to be of high-confidence if it does not contain any subsequence longer than 30 frames with all the frames being of low-consistency.

For the self-learning, we select frames from the high-confidence sequences to feed into the neural network for fine-tuning. Note that frames in high-confidence sequences have consistently high-confidence and could all be used to extend the training data. However, since the sequence is captured continuously, the difference between adjacent frames is often very subtle, and so these frames would not provide much new information to the network. Thus, we sample one out of every 25 frames from each high-confidence sequence for network fine-tuning. If after one round of fine-tuning we obtain new high-confidence sequences, we iterate the process to extend the training data with the new sequences. We continue this iterative process of self-learning until no new high-confidence sequences are generated. Then, we switch to the active learning to ask for additional user annotations.

For the active learning, we ask users to annotate frames selected from the low-confidence sequences. The total number of frames to be annotated, m, is divided by the number of low-confidence sequences to determine the number of frames that we select from each sequence. For each sequence, we randomly sample the frames by assigning a high sampling probability to low-consistency frames, since these frames are more likely to be labeled incorrectly.

6.2 User input

During the active learning, the user is asked to annotate frames from low-confidence sequences. Since we do not have an estimate of sequence confidence at the beginning of the active learning, we select the frames to be annotated by the user based on the registration error obtained using the method of Newcombe et al. [2011]. Specifically, when registering the frames using depth maps without semantic information, we record the values of the registration energy in the iterative alignment of the sequence. Peaks in the energy curve formed by a sequence of frames correspond to larger registration errors, which are more likely to be improved by adding semantic information to the frames involved. Thus, we sample frames by assigning high probability to frames corresponding to curve peaks.

For user annotation, we use a scribbling interface similar to the one proposed by Wong et al. [2015]. The input frame is first segmented into a set of super-pixels. The user selects a label from a menu and draws one or more scribbles over the image. The system then groups all the super-pixels covered by the scribbles into a segment, and assigns the selected label to the segment.

Fig. 6. Examples of labeling instructions provided to the users.

7 RESULTS AND EVALUATION

We present results of our semantic reconstruction method, and evaluate the active learning and frame segmentation method.

Dataset. We use RGBD sequences from a large dataset of object scans provided by Choi et al. [2016]. We select three representative categories that capture a variety of object characteristics to test our method: chairs, tables, and motorcycles. For example, chairs have various topologies and part connections, tables have large, flat regions which are challenging to register, and motorcycles possess many different small parts and complex structures. For each category, we select sequences where the object to be reconstructed appears in all the frames. Specifically, we selected 78 sequences with motorcycles, 100 sequences with tables, and 160 sequences with chairs. For each category, we divide the sequences roughly into an 8:2 ratio for training:testing. On average, each sequence has around 2,000 frames, which results in a dataset with a total of around 676,000 frames for all sequences and categories.

Frame annotation. To obtain annotations of the frames, we used the services of a company that labeled the frames with quality guarantees. Thus, this data is more reliable than data obtained through crowdsourcing, which would require multiple users to label each frame to ensure the consistency of the data. We provided example annotations to the employees, to ensure the consistency of the label names and segment sizes, as shown in Figure 6.

7.1 Evaluation of semantic segmentation

We first evaluate our deep network to demonstrate that it provides accurate labeled frames if trained with adequate data. Since the entire dataset is comprised of around half a million frames, it is infeasible for the users to annotate all the frames. Thus, we perform the training and evaluation on a subsample of the frames. Specifically, for each category, we sample m frames from the testing sequences to test the segmentation accuracy. For the active learning, we start with $2m$ training frames annotated by the users. In each iteration of the active learning, we ask the users to annotate m additional frames. We end the active learning after 8 iterations, once the number of annotated training frames reaches $9m$. We use a batch of m new frames at each iteration since it is expensive to retrain the network, and thus it would be impractical to retrain after

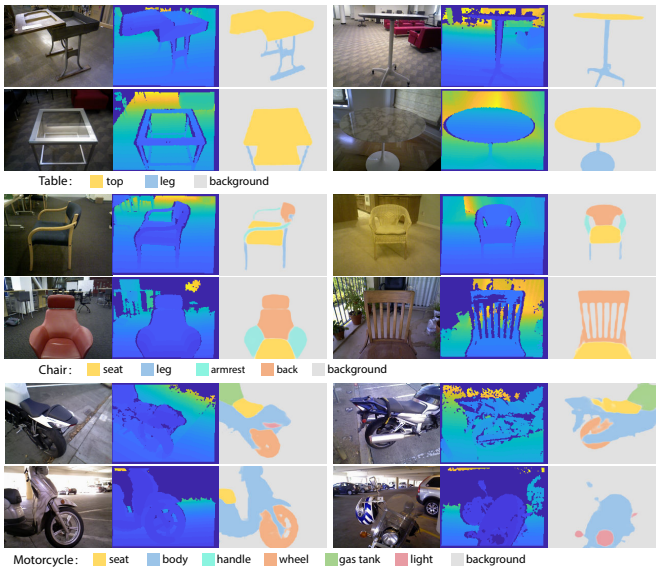

Fig. 7. Selected segmentations and labelings of frames obtained with our deep network. Each example shows the RGB and depth inputs, and the prediction. Note the semantic correctness and low noise level of the results.

adding single images. The frames labeled during the active learning are selected by the algorithm, while the initial set of training frames are uniformly selected from the training frames based on the registration error, as described in Section 6. In our experiments, $m = 200/100/120$ for chairs/tables/motorcycles.

The overall accuracies of labeling object parts on the test sets of chairs, tables, and motorcycles are 94.2%, 97.6%, and 91.7%, respectively. We obtain an average labeling accuracy of 94.3% for all classes. Note that the average *object classification* accuracy of the pre-trained network is 98.6%. Figure 7 shows example labeling results on selected test frames from the three classes used in our work. We observe in this qualitative evaluation that the segmentations are semantically correct, as implied by the average accuracy. The results also have a low amount of noise, and the size of the components is approximately correct, despite noise in the input frames, such as in the back of the bottom-right chair, and the region near the wheel of the top-right motorcycle. On the other hand, we note that improving 2D segmentation is not a contribution of our work, as we use a standard deep architecture for this task. Thus, it is possible that other network architectures may provide higher accuracies.

Active self-learning. We also show that we obtain a better performance by incorporating the self-learning into our approach, in contrast to using active learning only. Figure 8 shows the rate of improvement in the labeling accuracy of the test data for the two methods. Each circle in the graph represents a model trained with the data provided by the iterations up to that point. We clearly see that with the same number of frames provided by the users, the self-learning provides a more accurate model and improves the labeling accuracy, due to the additional training frames provided by the self-learning. Note how the accuracy improves not only when

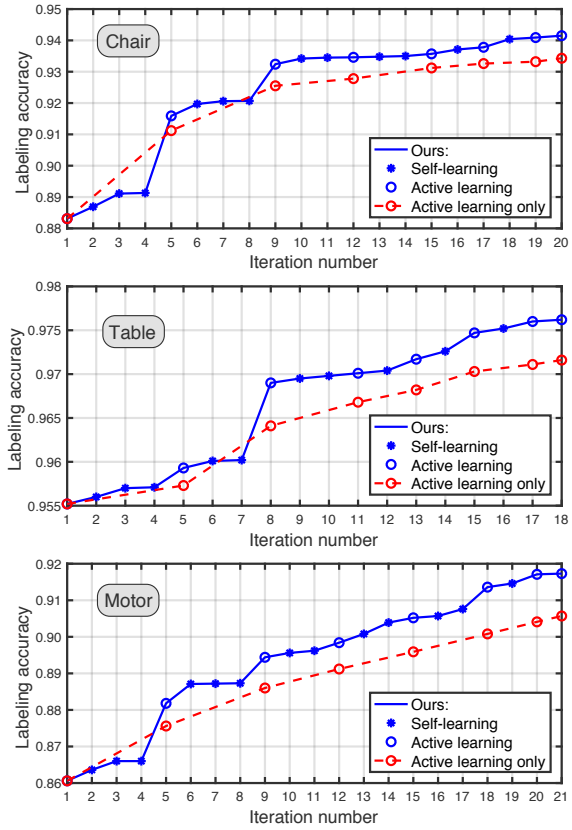

Fig. 8. Comparison of the segmentation accuracy of our active self-learning to a method based on active learning only, on three classes of objects.

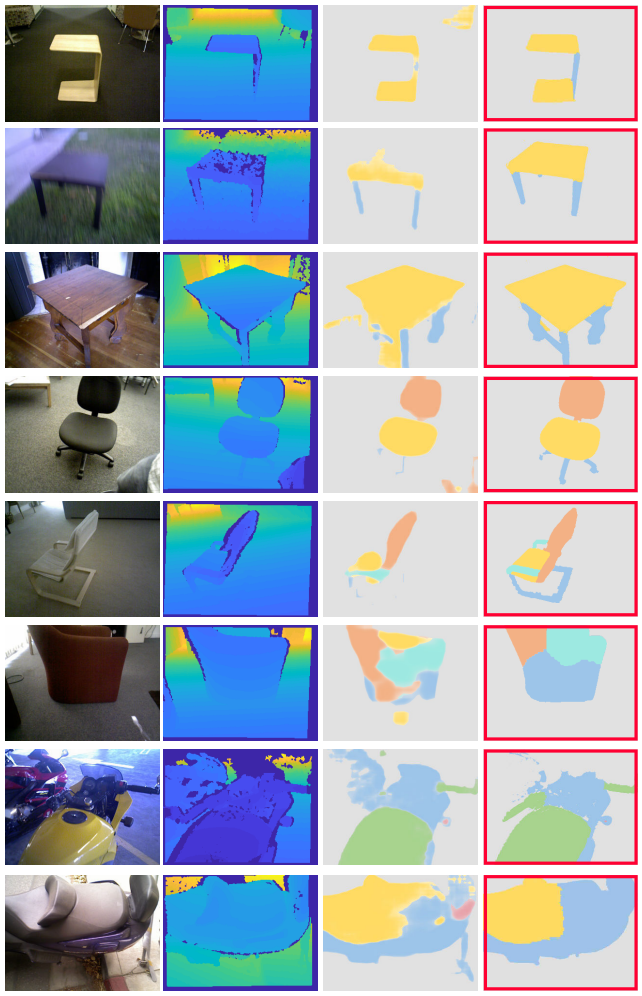

Fig. 9. Comparison between the label prediction provided by the neural network and the labels obtained after fusion and back-projection (in the red boxes). Note the improvement in the quality of segments after fusion.

frames annotated by users are added (points labeled in blue as "Active learning"), but also when frames automatically selected by the self-learning are added to the training data (points labeled in blue as "Self-learning"). In contrast, the method based purely on active learning (red points) never reaches the same accuracy as the active self-learning, after two or more batches of user input were provided.

To further demonstrate that the frames fed to the network during the self-learning provide useful information to the network, Figure 9 shows the labeling of frames provided by the neural network, compared to the labeling of the same frames obtained after the labels are fused, refined, and projected back onto the frame during the training. We notice a clear improvement in the accuracy of the segmentation, with less noise in the detected segments, as in the one-seat sofa. Some parts that were only faintly detected, such as the chair legs, are labeled with more accuracy after fusion. Moreover, for all the test frames, we compute the accuracy of re-projected labels and compare that to the predicted labeling accuracy. We obtain an accuracy of 96.9% (vs. 94.2%) for chairs, 97.7% (vs. 97.6%) for tables, and 94.4% (vs. 91.7%) for motorcycles. We see that for chairs and motorcycles, the accuracies increase by around 2.7% due to the re-projection. The accuracy of tables does not increase much since the prediction accuracy is already quite high at 97.6%.

Minimal number of images required to bootstrap the pipeline. Although we start executing our pipeline with m frames annotated for each category, where m is derived from the initial registration error given by the method of Newcombe et al. [2011], the active learning can be started without any labeled images. Here, we investigate the minimal number of images that need to be annotated so that the self-learning component is activated and the whole pipeline is bootstrapped. Since we need high-confidence sequences to enable the self-learning, we calculate the minimal number of annotated images required for high-confidence sequences to appear. To obtain the minimal number, we perform an iterative search where we start the pipeline with a fixed number of annotated images and verify if high-confidence sequences are produced. We start with 400/200/240 images for chairs/tables/motorcycles, and iteratively decrease these numbers of initial images by 50/20/20, until no high-confidence sequences are produced. We find that when the initial number of

Fig. 10. Gallery of reconstruction results obtained with our method.

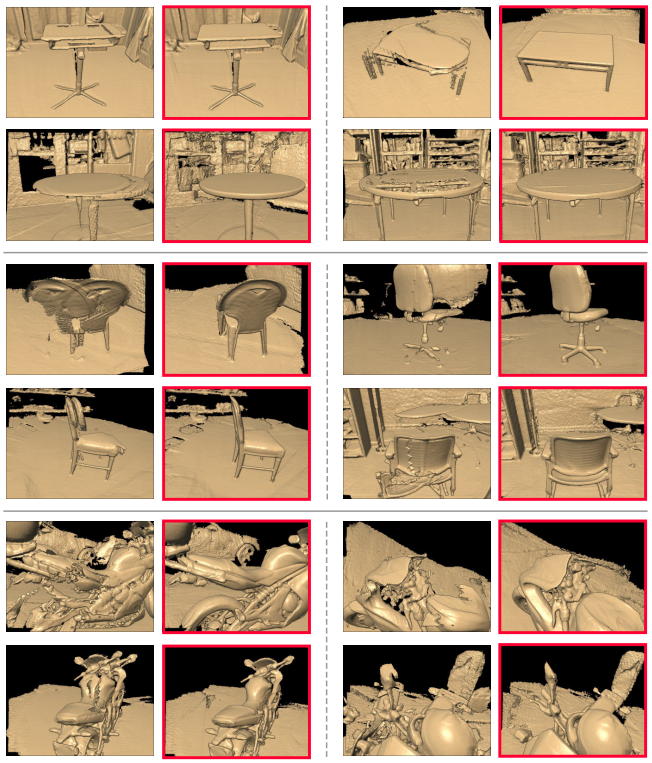

Fig. 11. Reconstruction results obtained with the method of Nießner et al. [2013], which does not consider semantic information (left of each example), compared to the results of our method that incorporates semantic information (right, in the red boxes). Note how our reconstructions are smoother, and have less missing data and less misalignments.

images is less than 200/140/180, no high-confidence sequences are produced for chairs/tables/motorcycles.

7.2 Evaluation of semantic reconstruction

Figure 10 shows a gallery of reconstruction results obtained with our method on test data, for four sequences from each class in the dataset. Note that both the reconstructions and their segmentations are of high-quality, despite the complexity of the acquired shapes. The tables and chairs possess different types of topologies, reflected by the connections between object parts, while the motorcycles have many small parts around their engines and handle bars. Although each shape is labeled mainly into four parts, the boundaries between parts are accurate, except for some of the motorcycle wheels (second row), since there are many training examples that have covered wheels (top motorcycle).

It is difficult to perform comparisons to related works such as Xiao et al. [2013] and Häne et al. [2016], given that the input assumptions of these works are quite different from our work, as discussed in Section 2. Thus, we compare our results to reconstructions obtained with the method of Nießner et al. [2013]. The method of Nießner et al. and our method are similar in that they align frames to an evolving method. However, their method uses mainly depth information for the alignment without involving any learning, while our method also considers semantics. We also perform this evaluation on separate test data not used during the active learning. Figure 11 shows a qualitative comparison of the reconstructions of both methods on selected test sequences. It is noticeable that the results obtained with our method are less noisy and have less missing regions in the reconstructed models, e.g., chairs and tables. In addition, the results on the motorcycles and one of the chairs do not have the misalignments of frames which are visible in the results obtained without semantics.

To evaluate the results in a quantitative manner, we randomly selected 20 sequences from each class and asked 5 users to visually compare the reconstruction results with and without semantics. Since the method of Nießner et al. [2013] does not automatically remove the background, to avoid a biased comparison, we only showed to the users our reconstruction results in the same manner as Figure 11, without coloring the semantic parts or removing the background. We put the reconstruction videos of those two methods side by side, and ask the users which reconstruction result is better. Users can choose from three options: A) Left is better, B) Right is better, and C) Same quality. For 95% of the sequences, the users considered our reconstruction results to have at least the same quality as the ones obtained with the method of Nießner et al. [2013], among which 32% of our results were considered to have better quality. The chair category received the most positive feedback from the users, with 50% of the reconstruction results being considered better. The main reason for this outcome is that most of the chairs have complex and thin structures which are hard to capture using

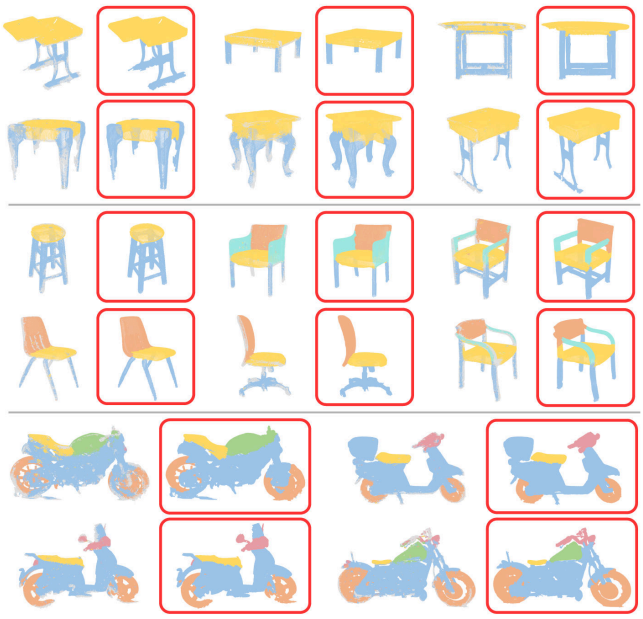

Fig. 12. Comparison of part labels before and after applying label refinement (in the red boxes).

depth alone, while our part label prediction can usually locate such regions well. The 5% of our results that were considered to have worse reconstruction quality have incorrect label predictions. The incorrect labels lead to an incorrect alignment between the frames. In the supplementary material, we also provide a comparison of our semantic registration to a global registration method, the 4PCS method of Aiger et al. [2008].

In Figure 12, we compare the reconstruction results obtained with and without applying the label refinement with graph cuts after the label fusion and background removal. In this manner, we evaluate the effect of this post-processing step in the final result. We see in these results how the refinement has the effect of smoothing out the labels and creating larger connected components with the same label, while the results without refinement have more gaps in the segments and small connected components.

7.3 Timing statistics

We test our method on a computer with an Intel Xeon 2.10GHz CPU with 32GB of memory and an NVIDIA Quadro M4000 GPU. The Average time for reconstructing one testing sequence, including the semantic segmentation of the frames, is 5.41fps for chairs, 7.41fps for tables, and 4.55fps for motorcycles. Comparing to the original implementation of Nießner et al. [2013], whose average timing is 13.33 fps on our computer, our method is 2-3 times slower. This is mainly due to the extra amount of data transmission between CPU and GPU for the labeling information. The processing time increases linearly with the number of labels.

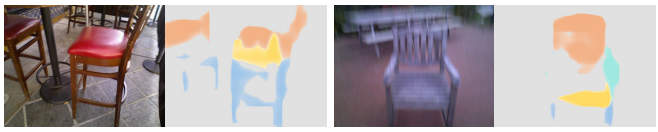

Fig. 13. Failure cases in the frame labeling. Left: object of the same class in the background. Right: motion blur.

8 CONCLUSION, LIMITATIONS, AND FUTURE WORK

We introduced a learning-based reconstruction method composed of two main components. First, a deep network that provides a semantic labeling of RGBD frames. Second, an alignment and reconstruction procedure guided by the semantic labels of the frames. To alleviate the creation of the training data for the deep network, we introduced an active self-learning framework that enables the creation of training data while requiring minimal user input. We showed in our experiments that the use of a semantic labeling improves the quality of reconstructions, especially when compared to a state-of-the-art method that aligns frames using depth information only. Moreover, we showed that the active learning performs well in terms of requiring a small amount of user-annotated frames, while enabling us to train a network for labeling RGBD frames accurately.

Limitations and future work. As a first solution for semantic reconstruction, a few of the components in our method currently have limitations and could thus be further developed in future work. For example, we use the semantic labeling mainly to partition frames and voxels into regions with contiguous labels; then, these regions are aligned with ICP based only on the depth of the frames. In future work, we would like to investigate possible approaches for taking advantage of the semantic labeling also during the alignment, e.g., by possibly finding matching features in the semantic regions.

Moreover, as shown by the examples in Figure 13, we can obtain an incorrect labeling of frames when an object of the same class is present in the background of the scans, or when significant blur exists in the input frames due to motion. In addition, our method is unable to handle datasets in which large occlusions are present. Occlusions can cause abrupt changes in the continuity of labels across the frames, which conflict with the smoothness expected by the registration and label refinement methods.

In general, we have shown that a semantic labeling leads to better reconstruction results. Nevertheless, reconstructing 3D objects from scans acquired in a casual, handheld setting remains a difficult problem. There are still many factors that can pose difficulties during semantic prediction and alignment of frames, such as background clutter, fast camera movements, occlusions, and suboptimal viewpoints. Thus, there is still room for further work in this area.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported in parts by NSFC (61602311, 61522213, 61761146002, 61702338), 973 Program (2015CB352501), GD Science and Technology Program (2015A030312015), Shenzhen Innovation Program (JCYJ20170302153208613, KQJSCX20170727101233642), ISF-NSFC Joint Research (2472/17), and NSERC Canada (2015-05407).

REFERENCES

Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 2008. 4-points congruent sets for robust pairwise surface registration. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 85.

Yuri Boykov and Vladimir Kolmogorov. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Analysis & Machine Intelligence* 26, 9 (2004), 1124–1137.

Kang Chen, Yu-Kun Lai, and Shi-Min Hu. 2015. 3D indoor scene modeling from RGB-D data: a survey. *Computational Visual Media* 1, 4 (2015), 267–278.

Sungjoon Choi, Q. Y. Zhou, and V. Koltun. 2015. Robust reconstruction of indoor scenes. In *Proc. CVPR*. IEEE, 5556–5565.

Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. 2016. *A Large Dataset of Object Scans.* Technical Report. arXiv:1602.02481.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. CVPR*. IEEE.

M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. 2015. 3D scanning deformable objects with a single RGBD sensor. In *Proc. CVPR*. IEEE, 493–501.

Noa Fish, Oliver van Kaick, Amit Bermano, and Daniel Cohen-Or. 2016. Structure-oriented Networks of Shape Collections. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 35, 6 (2016), 171:1–14.

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. 2015. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review* 43, 1 (2015), 55–81.

Kan Guo, Dongqing Zou, and Xiaowu Chen. 2015. 3D Mesh Labeling via Deep Convolutional Neural Networks. *ACM Trans. on Graphics* 35, 1 (2015), 3:1–12.

C. Häne, C. Zach, A. Cohen, and M. Pollefeys. 2016. Dense Semantic 3D Reconstruction. *IEEE Trans. Pattern Analysis & Machine Intelligence* PP, 99 (2016), 1–14.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. (2016).

Qixing Huang, Hai Wang, and Vladlen Koltun. 2015. Single-view Reconstruction via Joint Analysis of Image and Shape Collections. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 34, 4 (2015), 87:1–10.

O. Kähler, V. Adrian Prisacariu, C. Yuheng Ren, X. Sun, P. Torr, and D. Murray. 2015. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE TVCG* 21, 11 (2015), 1241–1250.

E. Kalogerakis, A. Hertzmann, and K. Singh. 2010. Learning 3D Mesh Segmentation and Labeling. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 3 (2010), 102:1–12.

C. Kerl, J. Sturm, and D. Cremers. 2013. Robust odometry estimation for RGB-D cameras. In *Proc. Int. Conf. on Robotics & Automation*. IEEE, 3748–3754.

Young Min Kim, Niloy J. Mitra, Dong-Ming Yan, and Leonidas Guibas. 2012. Acquiring 3D Indoor Environments with Variability and Repetition. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 138:1–11.

A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. 2014. Joint Semantic Segmentation and 3D Reconstruction from Monocular Video. *LNCS (Proc. ECCV)* 8694 (2014), 703–718.

Minmin Lin, Tianjia Shao, Youyi Zheng, Niloy Jyoti Mitra, and Kun Zhou. 2018. Recovering Functional Mechanical Assemblies from Raw Scans. *IEEE transactions on visualization and computer graphics* 24, 3 (2018), 1354–1367.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*. IEEE, 3431–3440.

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. 2017. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *Proc. Int. Conf. on Robotics & Automation*. IEEE.

V. Morell-Gimenez, M. Saval-Calvo, J. Azorin-Lopez, J. Garcia-Rodriguez, M. Cazorla, S. Orts-Escolano, and A. Fuster-Guillo. 2014. A comparative study of registration methods for RGB-D video of static scenes. *Sensors* 14, 5 (2014), 8547–8576.

Liangliang Nan, Ke Xie, and Andrei Sharf. 2012. A Search-classify Approach for Cluttered Indoor Scene Understanding. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 137:1–10.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. Int. Symp. on mixed and augmented reality*. IEEE.

Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. 2013. Real-time 3D Reconstruction at Scale Using Voxel Hashing. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 32, 6 (2013), 169:1–11.

Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. 2013. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proc. CVPR*. 2027–2034.

M. Rünz and L. Agapito. 2017. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *Proc. Int. Conf. on Robotics & Automation*. 4471–4478.

R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. 2013. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proc. CVPR*. IEEE, 1352–1359.

Ariel Shamir. 2008. A survey on Mesh Segmentation Techniques. *Computer Graphics Forum* 27, 6 (2008), 1539–1556.

Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. 2012. An Interactive Approach to Semantic Modeling of Indoor Scenes with an RGBD Camera. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 136:1–11.

Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. 2012. Structure Recovery by Part Assembly. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 180:1–11.

Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. 2016. Deep Automatic Portrait Matting. In *Proc. Euro. Conf. on Computer Vision*. Springer, 92–107.

Zhenyu Shu, Chengwu Qi, Shiqing Xin, Chao Hu, Li Wang, Yu Zhang, and Ligang Liu. 2016. Unsupervised 3D shape segmentation and co-segmentation via deep learning. *Computer Aided Geometric Design (Proc. Geometric Modeling and Processing)* 43 (2016), 39–52.

Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 30, 6 (2011), 126:1–10.

Jörg Stückler, Benedikt Waldvogel, Hannes Schulz, and Sven Behnke. 2015. Dense Real-time Mapping of Object-class Semantics from RGB-D Video. *J. Real-Time Image Process.* 10, 4 (2015), 599–609.

S. Thrun. 2002. Robotic Mapping: A Survey. In *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel (Eds.). Morgan Kaufmann, 1–35.

Stavros Tsogkas, Iasonas Kokkinos, George Papandreou, and Andrea Vedaldi. 2015. Semantic part segmentation with deep learning. *arXiv preprint arXiv:1505.02438* (2015).

Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. 2011. A Survey on Shape Correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707.

Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2012. Active Co-Analysis of a Set of Shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 165:1–10.

Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. 2015. ElasticFusion: Dense SLAM Without A Pose Graph. In *Proc. of Robotics: Science and Systems*.

Yu-Shiang Wong, Hung-Kuo Chu, and Niloy J. Mitra. 2015. SmartAnnotator: An Interactive Tool for Annotating Indoor RGBD Images. *Computer Graphics Forum (Proc. Eurographics)* 34, 2 (2015), 447–457.

Fangting Xia, Peng Wang, Liang-Chieh Chen, and Alan L Yuille. 2015. Zoom Better to See Clearer: Human and Object Parsing with Hierarchical Auto-Zoom Net. In *Proc. Euro. Conf. on Computer Vision*, Vol. 9909.

J. Xiao, A. Owens, and A. Torralba. 2013. SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels. In *Proc. CVPR*. IEEE, 1625–1632.

Kai Xu, Hanlin Zheng, Hao Zhang, Daniel Cohen-Or, Ligang Liu, and Yueshan Xiong. 2011. Photo-Inspired Model-Driven 3D Object Modeling. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 30, 4 (2011), 80:1–10.

Mingliang Xu, Mingyuan Li, Weiwei Xu, Zhigang Deng, Yin Yang, and Kun Zhou. 2016. Interactive mechanism modeling from multi-view images. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 236.

Feilong Yan, Andrei Sharf, Wenzhen Lin, Hui Huang, and Baoquan Chen. 2014. Proactive 3D Scanning of Inaccessible Parts. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 33, 4 (2014), 157:1–8.

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. 2016. A Scalable Active Framework for Region Annotation in 3D Shape Collections. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 35, 6 (2016), 210:1–12.