Semantic Mapping: A Semantics-based Approach to Virtual Content Placement for Immersive Environments

Jingyang Liu
Computational Design Laboratory
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
jingyanl@andrew.cmu.edu

2021 17th International Conference on Intelligent Environments (IE) | 978-1-6654-0346-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/IE51775.2021.9486580

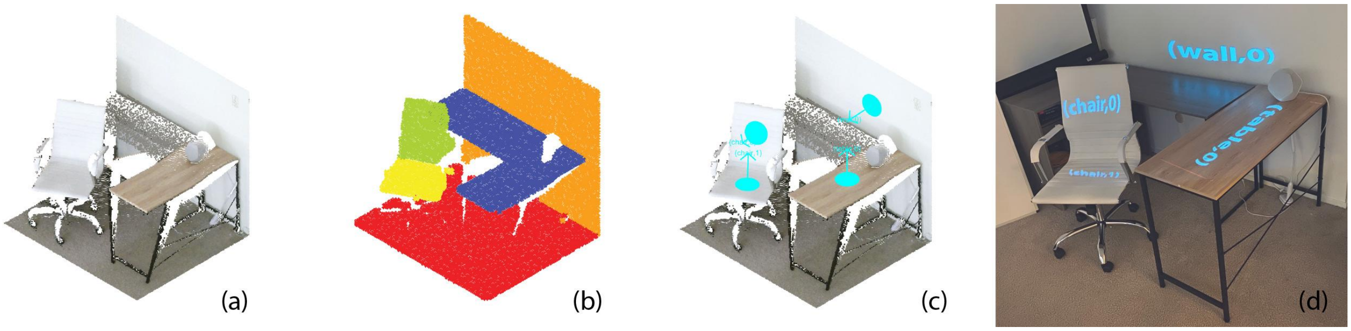


Fig.1.Semantic mapping reconstructs and segments a scene into surfaces with semantic labels. The system allows users to place virtual content to a target surface by referring to its semantic label. (a) the system reconstructs the scene and represents it in point clouds (b) the system segments the point clouds into semantic groups (c) the system creates an anchor point for each surface (d) the system project the semantic label to the associated surface

***Abstract*—Semantic Mapping is a semantics-based interactive system that enables intuitive virtual content placement for projection mapping in intelligent environments. Our semantic mapping system embeds semantic information of the environment to provide a user with an easy way to control and place projected virtual items in the physical world. In contrast to traditional projection mapping that involves manual adjustments, this semantic mapping system enables efficient manipulation of virtual content through inputs from users via speech or text. To build the system, we first use a commercial depth camera for scene reconstruction and an end-to-end deep learning framework for semantic segmentation at the instance level. We illustrate the system by developing a prototype for a set of proof-of-concept, room-scale applications. The accuracy study and user study results show that the system can provide users with accurate semantic information for effective virtual content placement.**

I. Introduction

Placement of virtual content can be a tedious task in immersive environments. Matching the virtual information with its correlated physical objects requires careful design and manual corrections, which poses challenges for end-users when the physical environment is uncontrolled or dynamic. For example, within the built environment like the workplace and home, it is not practical to manually adjust projection mapping in adaptation to changes in real-time. Many prior works have investigated approaches for automatic virtual content placement. For example, researchers have built a user-centric toolkit that recognizes user gestures for manually defining the placement of the interface [1]. Many studies have been done on ad hoc virtual content placement based on the user's perspective [2], the geometry of the physical surfaces [3], and the state of space [4]. However, using nonverbal communication to convey information in human computer interaction can be ambiguous, since an action can be interpreted many different ways due to different context such as culture and gender [5].

Beyond low-level features such as the geometry of surfaces and users' position, semantics, as a high-level understanding of the physical environment, can play a crucial role in virtual content placement. For example, embedding semantic information allows users to place virtual content naturally and intuitively. Instead of translating cognitive goals into gestures or body movements, users can directly define the location of virtual content by referring to the semantic information of a physical object. With multiple modalities of interaction techniques such as speech recognition, the ambiguity and information loss problems among the user commands and intentions of where to place virtual content can be solved. Additionally, for augmented reality applications, semantics can provide situated information for building context-aware interactions with virtual content and environments.

To facilitate the semantic-based virtual content placement at room-scale, we presented a semantic-based system that is

978-1-6654-0346-7/21/$31.00 ©2021 IEEE

composed of three key components: 1) 3D reconstruction 2) 3D segmentation, and 3) Labeling. First, we used a commercial depth camera (Microsoft Kinect V2) for dense scene reconstruction. Based on the KinectFusion framework [6], we obtain a 3D model of the scene represented in a dense point cloud. Then, we use an end-to-end semantic segmentation model to segment the captured point cloud into clusters at the instance-level. For each cluster, we assign a geometry ID for coplanar point clouds and use the geometry ID to unify the semantic label of the point cloud via majority voting. Thus, the scene is parsed into a group of surfaces with both semantic and geometric labels. Finally, the anchor point for virtual content placement is placed at the center of the projectable region. Each anchor point contains the information of its semantic label, position ,and normal. Thus, users may place virtual content directly onto a physical surface by referring to its semantic label, which is automatically generated by our proposed system (Figure 1).

In summary, this paper contributes to the topic of virtual content placement for immersive environments by:

- We built an end-to-end 3D point cloud processing pipeline including 3D scene reconstruction, 3D semantic segmentation, 3D geometric segmentation, and labeling.
- Built upon the pipeline, we created *Semantic Mapping*, a system that achieves automatic instance-level projection mapping and semantic-based interaction for virtual content placement.
- We presented a generic prototype with a proof-of-concept application to facilitate similar application deployment for using natural language to place digital content in the physical world.

We believe that by embedding semantic information of the scene, our proposed system can provide both content creators and end-users with a high-level and intuitive tool for arranging virtual content in the physical environment. The system can be applicable to a wide range of room-scale applications in projection mapping, augmented reality and mixed-reality,

II. Related Work

A. Spatially Augmented Reality

Spatial augmented reality (SAR) uses projection mapping to augment physical objects with virtual information [7]. The concept was initially demonstrated by Raskar et al. with applications [8] [9]. Previous works have explored SAR from tabletop [10] to room-scale augmentations [11].

With the increasing accessibility of commercial depth sensors such as Kinect, intensive studies have been done to integrate context-awareness into interactive projection mapping. At room-scale, the real-time information captured by depth sensors enables the rectification of the projector's output to accommodate users' perspective [3] or the physical layout of a room [4]. At human-scale, prior works presented elegant approaches for gesture-based input on everyday projected surfaces. For instance, *WordKit* provides a system for users to "paint" a user interface where and when it is needed [1]. *OmniTouch* provides a depth camera and projection system that enables multi-touch finger interaction on arbitrary, every day surfaces [12].

B. Virtual Content Placement

The placement of virtual content plays a crucial role in augmented reality (AR) and projection mapping. The topic is closely related to the problem of view management [13]. Factors such as visibility [14] and legibility [15] [16] have largely been investigated in previous study. Context-aware systems automatically decide when, where and how much information to be displayed based on users' current cognitive load and knowledge about their task and environment [17]. Multiple works utilize features from the real world such as point lights [18] and visual saliency [19] for adjusting the placement of virtual content.

Geometry-based system addresses automated content placement based on the geometry of physical surfaces [20]. By detecting planes in the real world, AR system can adapt virtual content to the target physical surfaces and integrate physical constraints to virtual systems. For instance, *SnapToReality* extracts 3D geometric constraints from real world for snapping virtual content to real 3D edges and planar surfaces in augmented reality [21]. DepthLab uses real time depth data for building a variety of depth-based UI/UX paradigms for augmented reality [22]. Mobile AR systems such as ARKit and ARCore have encapsulated plane detection for building geometry-aware augmented reality applications.

Semantics, as a high-level understanding of the physical environment, can provide context awareness for virtual content placement. However, to the best of our knowledge, semantic information of the real world is limited or undetected in prior works. A high-level semantic-based system for content placement remains absent. In this work, we present a system embedded with semantic information of the real world. The system allows end-users to interact with projected virtual content in the physical world intuitively and naturally. For instance, semantic-based system can support natural language interfaces that allow users to place virtual content using a human language.

C. Scene Understanding

Scene understanding aims to analyze objects in context with respect to the 3D structure of the scene. Most existing research on scene understanding is based on 2D images enabled by the success of deep convolutional neural networks [23] [24] [25]. Multiple prior works leverage 2D scene understanding for building context-aware applications in AR applications [26] [27].

With recent advances in volumetric scan fusion techniques, it is possible to reconstruct fine-grained 3D scenes from scans captured by a commodity depth camera [6]. In this work, we use a depth camera to capture 3D data of the environment and build a framework for 3D reconstruction and semantic segmentation. 3D segmentation is the process of decomposing 3D model into functionally meaningful regions.

Several traditional methods, such as edge-based [28],region-based [29], and model-fitting [30] have been proposed to group point clouds into homogeneous groups with similar local features. With the ever-growing amount of 3D shape databases [31] [32] and annotated RGB-D datasets [33] [34] becoming available, the data-driven approach starts to play an important role in 3D object recognition and has achieved impressive progress [35] [36]. Built upon prior works, we present a learning-based pipeline for acquiring semantic information from the scene represented in point clouds.

III. Method

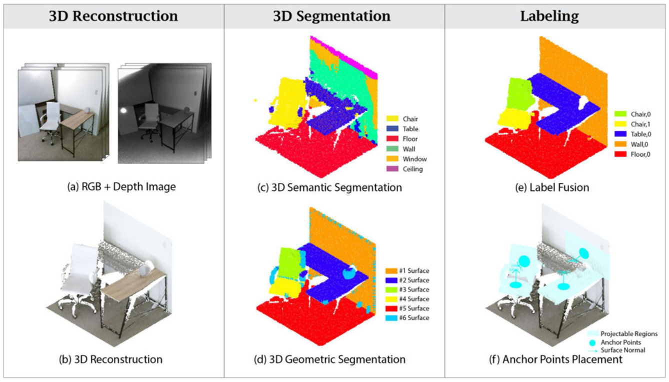


Fig. 2. The framework of the semantic-based system consisting of three components: 3D reconstruction, 3D segmentation and labeling. The 3D reconstruction component translates the acquired RGB and depth data from a Kinect sensor into point clouds. The 3D segmentation component segments point clouds into discrete surfaces with semantic label. The labeling component fuses geometric and semantic labels and creates anchor point for each segmented surface.

We use a commercial RGB-D camera Kinect V2 to acquire the depth and color information of the physical environment. To display virtual content, we use two projectors. Projectors and the Kinect V2 sensor are calibrated using the RoomAlive Toolkit [37].

As depicted in Figure 2, we first obtain RGB and depth data from a Kinect sensor. The system then automatically translates the low-level point cloud information into high-level semantic information through steps such as 3D reconstruction, 3D semantic segmentation, 3D geometric segmentation and labeling. Finally, the system places an anchor point at each segmented surface. Users can map virtual content onto a physical surface by referring to its semantic label. We use the Point Cloud Library in C++ for point cloud processing and TensorFlow Library in Python for building the deep learning architecture.

A. 3D Reconstruction

In this work, we obtain 3D reconstruction of the physical environment through dense simultaneous localization and mapping (SLAM). Following the KinectFusion framework [6], we use a Kinect V2 sensor to reconstruct the scene in four steps:

1) We obtain raw depth information at each image pixel in the image domain. To reduce noise, we applied a bilateral filter to the raw depth map.
2) Each frame of depth images is transformed into 3D points and integrated into a 3D volumetric data structure.
3) Like live camera localization that involves estimating the current camera pose for each frame, we obtain the Kinect sensor pose by the full frame model ICP method [38]. We assume that only a small camera motion occurs from one frame to the next, thus we can use a fast projective data association algorithm to obtain correspondence points and the point-plane metric for Kinect sensor pose estimation.
4) The point cloud reconstructed contains noise and outliers inherent due to the errors of the depth camera, we use statistical outlier removal algorithms to remove outliers and prepare an effective 3D model for further processing.

B. 3D Segmentation

The 3D segmentation component consists of two steps: 3D semantic segmentation and 3D geometric segmentation. The first step, 3D semantic segmentation, segments point clouds into clusters. Points in each cluster share the same semantic label. The second step, 3D geometric segmentation, is to further segment the point clouds into clusters based on their geometric properties. Each cluster contains the point clouds that are co-planar.

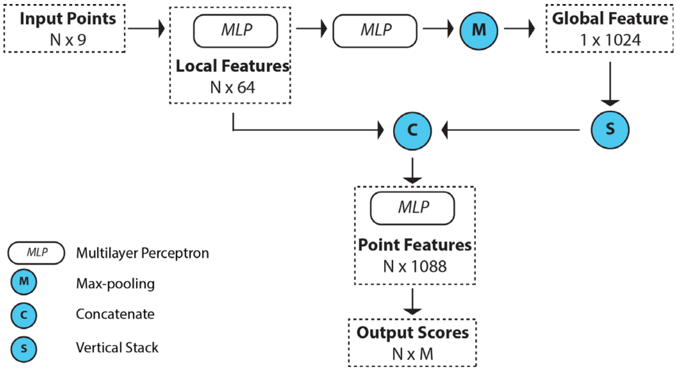


Fig. 3. Simplified architecture of PointNet [35]. (C): Concatenate (M): Max-pooling (S): Vertical stack The network samples N points within a region (in this case, we use 0.5 m by 0.5 m) as input, through a series of multi-layer-perceptrons the input N points are mapped into a 64 dimensional space, these are called local point features. Mas-pooling is applied to aggregate information from all the points resulting in a common global features, then the global feature is concatenated with all local features, after muti-layer perceptrons, these combined features are used to predict M output class scores.

In this work, we have trained a deep neural network for identifying the semantic meaning of 3D point cloud reconstructed from the depth camera. We used the PointNet architecture [35] for training a dataset of Stanford Large Scale 3D Indoor Scenes [39] [40]. The dataset contains 6020 square meters of indoor areas from diversified building typologies such as offices, conference rooms and open spaces. 12 semantic elements cover most commonly seen objects indoor, such as structural elements (*ceiling, floor, wall, beam, column, window and door*) and furniture (*table,chair, sofa, bookcase and board*).

PointNet is a deep neural network that directly consumes point clouds and outputs the per point semantic class labels. To prepare the training data, we first split the captured point cloud with area 0.5 m by 0.5 m and randomly sample 2,048 points from each block. Each selected point is represented by its Cartesian coordinates, color information, and its normalized coordinates to the captured scene. The 9-dimensional vectors are mapped into high-dimensional space via Multi-Layer-Perceptrons (MLPs). The high-dimensional local features are then aggregated into the global feature via Max-pooling. The global feature and the local feature are then concatenated as the point feature. Finally, the point feature is mapped to the output class scores via MLPs (Figure 3) .

After semantic segmentation, each point of the captured scene is classified into a given semantic class. We then segment the point cloud into parallel or shared planes based on the geometric properties of the point cloud (Figure 4).

Each point cloud is characterized by its estimated normal and normal distance from the origin [41] [42]. The normal vector of each point cloud can be estimated based on its adjacent point clouds. For each point p_i, we pick k nearest neighbors $N_p$and compute the corresponding covariance matrix C, which is defined as:

$$C = \begin{bmatrix} p_{i_1} - \overline{p} \\ \cdots \\ p_{i_k} - \overline{p} \end{bmatrix}^T \cdot \begin{bmatrix} p_{i_1} - \overline{p} \\ \cdots \\ p_{i_k} - \overline{p} \end{bmatrix}, i_j \in N_P$$

$\overline{p}$ is the centroid of N_p . Then we estimate surface normal by finding the smallest eigenvalue λ_0 and its corresponding eigenvector v_0 of the covariance matrix C . Assuming $\lambda_0 < \lambda_1 < \lambda_2$, we can estimate surface variation σ_i by:

$$\sigma_i = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$$

σ_i is a feature for detecting edge points. When the point clouds are distributed in a plane, σ_i is small. If σ_i of a point is larger than a threshold σ_τ, the point can be categorized as a point on edges or borders.

After normal estimation, the point clouds are grouped into surfels based on the angle between normal vectors. The angle θbetween vectors can be estimated by:

$$\theta = \cos^{-1}(u, v)$$

u and v are the normal of two points. If θ is within the defined angle threshold, two point clouds are grouped as parallel surfels. Finally, for each surfel, we use the normal distances of points from the origin to determine if the surfel shares a plane with other parallel surfels. Similarly, by setting a threshold d_t we find clusters of co-planar surfels from parallel surfels.

In order to obtain a robust and accurate segmentation, we used random sample consensus algorithm (RANSAC) to find inlier surfels and remove outliers. RANSAC algorithm first estimates a hypothesis plane based on the randomly selected three points from coplanar surfels. Point clouds are categorized as inliers if the distance between points and the hypothesis plane is below a threshold. After iterative processing, we find the plane that categorizes the maximum fraction of points as inliers. The outlier points are removed. The inlier points are labeled with the plane normal and each surfel is assigned a unique geometry ID.

C. Label Fusion

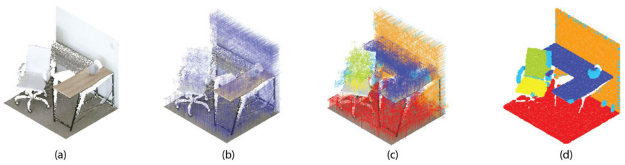


Fig. 4. Steps of 3D geometric segmentation : (a) Input point clouds (b) Estimate normal of each point (c) Cluster point clouds by normal and normal distance from the origin (d) Result of 3D geometric segmentation

After semantic segmentation and geometric segmentation, each point cloud is annotated with two labels — its semantic class and geometry ID. However, due to noise in the 3D reconstruction, the result of semantic segmentation is a combination of major correct point clouds and a small fraction of mislabeled point clouds.

We use a majority voting scheme to unify semantic labels of point clouds that shares the same plane. First, the result of 3D geometric segmentation is used to enclose a set of point clouds P as voters. Then we assign a representative semantic label L_P. The representative semantic label assigned for all of the points in $P$is determined by choosing the semantic label with the highest probability. The final semantic label L_P is obtained by

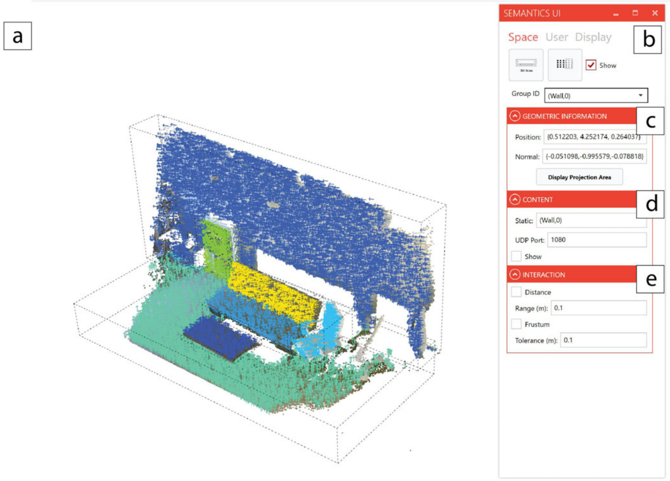
$$L_p = \underset{l \in \{1,\ldots,\mathcal{L}\}}{\arg\max} \frac{N_l}{N}$$

where l indexes through semantic labels and L is the number of semantic labels. N is the number of points in P, N_l is the number of points with the semantic label l .

We then determine the points that are visible from given locations of projectors based on their field of view using frustum culling algorithm [43]. By culling points visible from projector locations, we can determine regions in the scene that are projectable. For each region, we set an anchor point for virtual content at the centroid of all points within the region. An anchor point $Pt_i$is annotated with its Cartesian coordinates (x_i, y_i, z_i), normal N_i, semantic label l_{S_i} and geometric label l_{G_i}, formatted as $((x_i, y_i, z_i), N_i, l_{S_i}, l_{G_i})$.

IV. Result

Based on our proposed approach, we develop a prototype *Semantics UI* for end users to set up semantic-based content placement system. Then, we demonstrate the capability of the system using a proof-of-concept application.



SEMANTICS UI

Space User Display

3D Scan

☑ Show

Group ID (Wall,0)

GEOMETRIC INFORMATION

Position: {0.512203, 4.252174, 0.264037}

Normal: {-0.051098,-0.995579,-0.078818}

Display Projection Area

CONTENT

Static: (Wall,0)

UDP Port: 1080

□ Show

INTERACTION

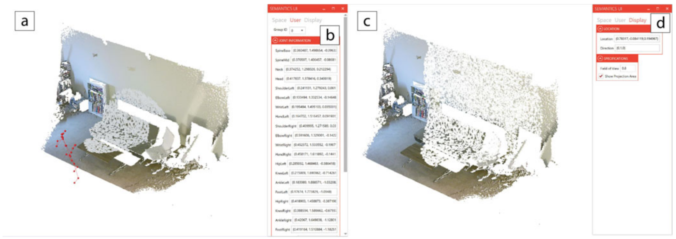
□ Distance

Range (m): 0.1

□ Frustum

Tolerance (m): 0.1

Fig. 5. The space component of Semantics UI prototype (a) the visual representation of 3D reconstruction point cloud of the physical world and its semantic segmentation result (b) the control panel for scanning, segmenting and labeling surfaces (c-e) the information panel for each segmented surface: (c) shows the default position for digital content placement of the surface (d) shows the server port which streams digital content to be displayed (e) provides two options for triggering the display: distance: the content will be projected if the distance between the user and the surface is below the set threshold (2) frustum: the content will be projected if the user's frustum intersects with the surface



SEMANTICS UI

Space User Display

Group ID 0

JOINT INFORMATION

SpineBase {0.360487, 1.498654, -0.2963

SpineMid {0.370597, 1.400457, -0.08681

Neck {0.374252, 1.298026, 0.212294}

Head {0.417837, 1.378416, 0.340019}

ShoulderLeft {0.241131, 1.279243, 0.061

ElbowLeft {0.133494, 1.332534, -0.14648

WristLeft {0.195484, 1.495103, 0.095031}

HandLeft {0.164702, 1.516457, 0.091901

ShoulderRight {0.409995, 1.271585, 0.03

ElbowRight {0.591606, 1.329001, -0.1422

WristRight {0.452372, 1.533552, -0.1907

HandRight {0.458171, 1.611892, -0.1441

HipLeft {0.285032, 1.468463, -0.380418}

KneeLeft {0.215009, 1.690962, -0.714261

AnkleLeft {0.183989, 1.888571, -1.03206

FootLeft {0.17674, 1.773829, -1.0948}

HipRight {0.418903, 1.458873, -0.387190

KneeRight {0.388594, 1.580662, -0.6793

AnkleRight {0.42067, 1.649838, -1.12801

FootRight {0.419164, 1.510884, -1.18251

SEMANTICS UI

Space User Display

LOCATION

Location {0.78317,-0.084119,3.194967}

Direction {0,1,0}

SPECIFICATIONS

Field of View 0.8

☑ Show Projection Area

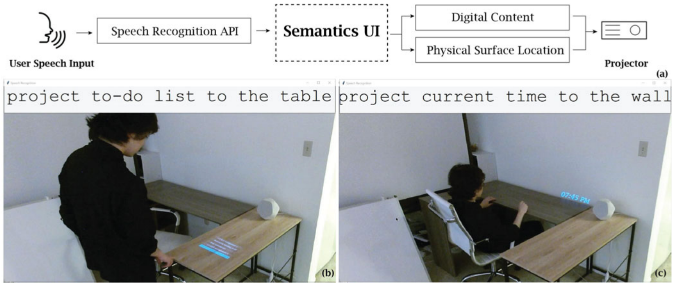
Fig. 6. The user component (a-b) and display component (c-d) of Semantics UI prototype, (a) the visual representation of the tracked user, regions that are visible to the user are highlighted (b) the panel showing the skeletal information of the tracked user (c) the visual representation of projectable area, regions that can be projectable are highlighted (d) the control panel for inputting the projector's specifications and location

A. Semantics Mapping UI

We developed a prototype *Semantics Mapping UI* (Figure 5) for end-users and content creators to set up a semantic-based system for projection mapping. The prototype is composed of three components: space, user and display.

- The space component allows users to access the point cloud reconstructed from the Kinect sensor. Based on our proposed method, users can click the segmentation button to segment the point cloud into instances and visualize the result. Each instance contains the information of its anchor point for virtual content placement, semantic label and UDP port address. Users can override the automatically generated anchor point by inputting arbitrary coordinate information. For each anchor point, users can define a UDP port for wireless communicating with an associated media server. The media server stores and streams virtual content to be displayed at the location of its bonding anchor point.
- The user component (Figure 6 (a-b)) can be used for tracking the information of users. The panel shows the user's skeletal and proxemic information. Based on the user's proxemic information such as position and orientation, we build a 3D isovists model to imitate human visual perception. 3D isovists model is a computational model ensuring that the salient visual characters can be directly visible rather than inferred indirectly [44]. We use the head joint position, head orientation, and the field of view of the user to cull point clouds that are visible to the user. The visibility of point clouds is a critical information for choosing the placement of virtual content.
- The display component stores the information of display devices such as projectors. Users can define the location and field of view of the projector by text input. Semantics Mapping UI can calculate and display point clouds that can be projectable from the projector location (Figure 6 (c-d)).

B. Example Application: Semantic-based Placement



project to-do list to the table

project current time to the wall

Fig. 7. The semantic-based placement feature parses the user's voice command and projects the virtual content onto the surface with the targeted semantic label (a) the pipeline for the semantic-based placement feature (b) the user uses speech input to places the digital content "to-do list" at the surface with semantic label "table"(c) the user uses speech input to places the digital content "time" at the surface with semantic label "wall"

Semantic-based placement allows users to place virtual content onto a target surface using human language. User can define the target surface by calling its semantic label (Figure 7). We build the natural language interface using *Google Speech-to-Text API*. The interface decomposes the speech input from the user into a pair of action and location for virtual content placement. For example, as depicted in Figure 7, the user says "*project to-do list to the table*", the system interprets speech input as a command *"project"*, a content *"to-do list"* a referent *"wall"*. Then the system retrieves the content "to-do list" from the media server and maps the media server UDP sender port to the UDP receiver port associated with the semantic label *"table"*. Finally, the virtual content *"to-do list"* is projected to the table at its anchor point. In contrast to traditional projection mapping that may require manual adjustment, the proposed semantic-based interaction allows end-users to place virtual content using high-level instructions directly. Moreover, since the system segments an object into multiple projectable planar surfaces, the user may place virtual content at a sub-surfaces of an object. For example, the label *"(chair, 0)"* represents the vertical surface of the chair and the label *"(chair, 1)"* represents the horizontal surface of the chair. Users can place virtual content at the horizontal surface of the chair by referring to its label *"(chair, 1)"*. To help user better understand our segmented label in real environment, we support a wake-up phrase – "*Show Labels*". After detecting such wake-up phrase, our system will project semantic labels to all recognized objects for 10 seconds.

V. Evaluation

To evaluate the usability of the system, we conducted both an accuracy study and user study. The accuracy study measures the accuracy of the 3D segmentation in both quantitative and qualitative manner. The user study evaluates the effectiveness of semantic-based interaction for virtual content placement.

A. Model Performance

TABLE I
Model evaluation on self-scanned point cloud

Scene	Method	Mean IoU	Overall Accuracy
Scene 1	Semantic-Only (PointNet)	66.3%	96.3%
	Label Fusion	71.1%	94.4%
Scene 2	Semantic-Only (PointNet)	71.3%	93.7%
	Label Fusion	86.4%	96.5%
Scene 3	Semantic-Only (PointNet)	46.2%	77.8%
	Label Fusion	65.7%	78.3%

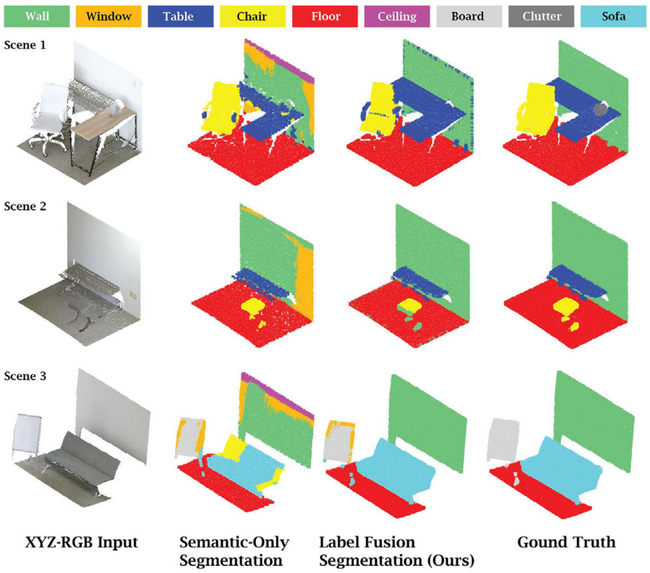


Fig. 8. Qualitative model evaluation of the semantic-only segmentation approach and the label fusion segmentation approach the point cloud of scanned scenes.

To evaluate the performance of our system, we evaluate and compare the accuracy of semantic-only approach and our proposed label fusion approach using two criteria including intersection over union (IoU) and overall accuracy. The result can be seen in the Table I .The mean accuracy over all classes was calculated using the following formula:

$$OverallAccuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN are the true positive case, true negative case, false-positive case, and false-negative case, respectively. Mean IoU over all classes was calculated using the following formula:

$$MeanIoU = \frac{\sum_{i=1}^{k} \frac{TP_i}{FP_i+FN_i+TP_i}}{k}$$

where TP_i is the true positive and FP_i is the false positive and FN_i is the false negative for class i. k is the number of class.

We then evaluate our trained model on point clouds obtained from workplaces and homes using a Kinect Sensor. The ground truth for each scene is manually labeled. We compare semantic-only approach with the label fusion approach we adopted for the system, which uses the majority voting to unify the semantic label of points sharing a plane. As depicted in Figure 8, point clouds that were misclassified by the semantic-only approach are corrected by adding plane constraints, which means our system can produce more uniform and accurate result on our own environment point cloud. However, point clouds we captured in this study are majorly composed of objects with simple geometry. We unified the semantic label of point clouds belonging to the same plane based on the assumption that co-planar point clouds share the same semantic class. Since our goal is to identify semantic labels of projectable surfaces, instead of recognizing semantic labels of every items, the geometry-priortized approach we used can well serve for projection mapping applications. This approach might not work for scenes that contain geometrically complex objects or various semantically different objects sharing similar geometry.

B. User Study

In order to evaluate the effectiveness of the semantic mapping system, we invited 11 undergraduate and graduate students to perform tasks with and without the system. The user study was done remotely, we used a commercial video conferencing software for communication.

The participants were invited to locate the virtual content by speech input remotely. We start a video conference, and a user on-site walks around the workplace. The remote participant was asked to observe the user on-site and relocate projected virtual content to the location that is visible to the user. The task was completed in two set-ups. In the first set up, we used a traditional projection mapping system. The participant can only give low-level instructions such as "move up", "move down", "move left" and "move right". An operator on-site adjusts the location of the projected content according to the participant's instructions. We choose to set the distance for each movement at 50 mm to strike a balance between speed and granularity. In the second set up, the participant tested the semantic mapping system. With the semantic-based system, the participant can define the location by referring to the surface semantic label. The semantic mapping system then parses instructions and automatically projects the content to the target surface. The task is considered as complete if the participant confirms the virtual content is placed at the desired location. We conducted ten iterations of tests and recorded completion time for each iteration. Each iteration consists of three tasks, according to observation, the tasks performed by the participants included remapping the virtual content from the chair to the wall, remapping the virtual content from the wall to the table and remapping the virtual content from the table to the chair. We found that semantic based mapping system (Mean = 6.98, Std = 1.22) was significantly faster ($p<.001$) overall compared to the traditional projection mapping system (Mean = 25.71, Std = 4.58).

We found significant differences in the task completion time between the traditional projection mapping and the semantic mapping for virtual content placement. The semantic mapping system allows users to remap the virtual content by referring to the target surface's the semantic label directly. The distance between the current and the target surface does not affect the completion time. However, users need to move the content incrementally without using the semantic-based system. The relocation of virtual content is time-consuming and sensitive to the distance of movement.

After the test, the participants were asked to take a survey to evaluate the semantic mapping system. In the survey, the participants were asked about four likert-scale questions: (1) The system can improve task performance (2) The system is useful in my daily life (3) The system is intuitive and easy to use (4) The system meets your expectation. Each question can be rated from 1 (strongly disagree) to 7 (strongly agree) with a step size of 1.

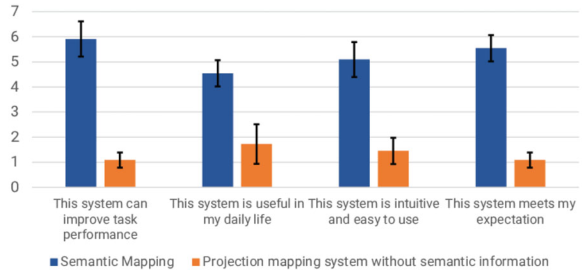


Fig. 9. Results of likert-scale survey questions

As depicted in Figure 9 , all four questions got positive results from our participants. More specifically, all participants agreed that our system can improve their task performance (Mean: 5.91, Std: 0.71). More than half of the participants believe our system is useful in their daily life (Mean: 4.55, Std: 0.52). 9 out of 11 participants agreed that our system is intuitive and easy to use (Mean: 5.10, Std: 0.70). And all participants agreed that the performance of our system met their expectation (Mean: 5.55, Std: 0.52). By comparing with speech-input projection system without semantic information, our system is significantly better from all four perspectives ($p < 0.001$ for all four questions). Also, according to the user's feedback, the semantic labels created by the system are similar to the "keyboard shortcuts" that efficiently map an instruction to an action.

VI. Conclusion and Future Work

We have presented a semantic-based approach to virtual content placement for immersive environments. We developed a system Semantics UI that automatically reconstructs a scene and segments it into surfaces with semantic labels. Enabled by the system, users can directly place virtual content onto a physical surface by referring to its semantic label. We implement a prototype to demonstrate a framework consisting of three components, 3D reconstruction, 3D segmentation ,and labeling. The prototype supports applications that allow users to interact with virtual content using natural language. To test the usability, we evaluated the system's accuracy and conducted a user study. According to the test results, the semantic-based system can provide users with efficient approaches to interact with virtual content in the real world. We believe that our proposed system can be applied to a wide range of applications in immersive environments, augmented reality and mixed reality. The system that segments scenes into surfaces with semantic labels can provide opportunities for natural and intuitive interaction and novel interaction techniques customized to the properties of objects.

For the next step, we aim at designing a more computationally efficient semantic segmentation algorithm and migrating the system to mobile devices. Alternative to direct point cloud segmentation, 2D scene segmentation model such as Faster R-CNN [45] and Mask R-CNN [46] can solve instance segmentation in 2D images. Semantic label obtained from 2D image can be associated with the point cloud reconstructed from correlated RGB images and depth images. By integrating 3D scene understanding and segmentation in mobile device applications, we can potentially use the physical world as a shared canvas for cross-device collaborations and enable users to interact with digital contents in immersive environments using natural language.

References

[1] R. Xiao, C. Harrison, and S. E. Hudson, “WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 879–888.

[2] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi, and L. Shapira, “RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units,” in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 2014, pp. 637–644.

[3] A. Fender, P. Herholz, M. Alexa, and J. Müller, “OptiSpace: Automated Placement of Interactive 3D Projection Mapping Content,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–11.

[4] A. Fender and J. Müller, “SpaceState: Ad-Hoc Definition and Recognition of Hierarchical Room States for Smart Environments,” in *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, 2019, pp. 303–314.

[5] R. B. Adler, L. B. Rosenfeld, N. Towne, and M. Scott, *Interplay: The process of interpersonal communication*. Holt, Rinehart, and Winston, 1986.

[6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, and A. Davison, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.

[7] R. Raskar, G. Welch, and H. Fuchs, "Spatially augmented reality," in *Proceedings of the international workshop on Augmented reality: placing artificial objects in real scenes: placing artificial objects in real scenes*, 1999, pp. 63–72.

[8] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay, "Shader lamps: Animating real objects with image-based illumination," in *Eurographics Workshop on Rendering Techniques*, 2001, pp. 89–102.

[9] R. Raskar, J. V. Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines, "iLamps: geometrically aware and self-configuring projectors," pp. 7–es, 2006.

[10] A. D. Wilson, "Depth-sensing video cameras for 3d tangible tabletop interaction," in *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*, 2007, pp. 201–204.

[11] J. Rekimoto and M. Saitoh, "Augmented surfaces: a spatially continuous work space for hybrid computing environments," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1999, pp. 378–385.

[12] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: wearable multitouch interaction everywhere," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 441–450.

[13] B. Bell, S. Feiner, and T. Höllerer, "View management for virtual and augmented reality," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 2001, pp. 101–110.

[14] A. Fender, D. Lindlbauer, P. Herholz, M. Alexa, and J. Müller, "Heatspace: Automatic placement of displays by empirical analysis of user behavior," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 611–621.

[15] B. Bell, S. Feiner, and T. Höllerer, "View management for virtual and augmented reality," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 2001, pp. 101–110.

[16] M. Tatzgern, V. Orso, D. Kalkofen, G. Jacucci, L. Gamberini, and D. Schmalstieg, "Adaptive information density for augmented reality displays," in *2016 IEEE Virtual Reality (VR)*, 2016, pp. 83–92.

[17] D. Lindlbauer, A. M. Feit, and O. Hilliges, "Context-aware online adaptation of mixed reality interfaces," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 147–160.

[18] K. Ahuja, S. Pareddy, R. Xiao, M. Goel, and C. Harrison, "Lightanchors: Appropriating point lights for spatially-anchored augmented reality interfaces," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 189–196.

[19] R. Grasset, T. Langlotz, D. Kalkofen, M. Tatzgern, and D. Schmalstieg, "Image-driven view management for augmented reality browsers," in *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2012, pp. 177–186.

[20] R. Gal, L. Shapira, E. Ofek, and P. Kohli, "FLARE: Fast layout for augmented reality applications," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2014, pp. 207–212.

[21] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson, "Snaptoreality: Aligning augmented reality to the real world," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1233–1244.

[22] R. Du, E. L. Turner, M. Dzitsiuk, L. Prasso, I. Duarte, J. Dourgarian, J. Afonso, J. Pascoal, J. Gladstone, and S. Izadi, "DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality," 2020.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[24] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[25] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.

[26] J. Barreira, M. Bessa, L. Barbosa, and L. Magalhães, "A context-aware method for authentically simulating outdoors shadows for mobile augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 3, pp. 1223–1231, 2017.

[27] L. Chen, W. Tang, N. W. John, T. R. Wan, and J. J. Zhang, "Context-Aware Mixed Reality: A Learning-Based Framework for Semantic-Level Interaction," in *Computer Graphics Forum*, vol. 39, 2020, pp. 484–496.

[28] A. D. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 292–299.

[29] A. Jagannathan and E. L. Miller, "Three-dimensional surface mesh segmentation using curvedness-based region growing approach," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 29, no. 12, pp. 2195–2204, 2007.

[30] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," in *Computer graphics forum*, vol. 26, 2007, pp. 214–226.

[31] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, and H. Su, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[32] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746–1754.

[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European conference on computer vision*, 2012, pp. 746–760.

[34] 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," *arXiv preprint arXiv:1709.06158*, 2017.

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

[36] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[37] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi, and L. Shapira, "RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units," in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 2014, pp. 637–644.

[38] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, 1992, pp. 586–606.

[39] 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.

[40] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[41] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 11, pp. 1114–1116, 1994.

[42] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *IEEE Visualization, 2002. VIS 2002.*, 2002, pp. 163–170.

[43] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," pp. 24–es, 2007.

[44] M. L. Benedikt, "To take hold of space: isovists and isovist fields," *Environment and Planning B: Planning and design*, vol. 6, no. 1, pp. 47–65, 1979.

[45] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.