The Visual Computer
https://doi.org/10.1007/s00371-022-02707-8

ORIGINAL ARTICLE

A semantic-based approach to digital content placement for immersive environments

Jingyang Liu[1] · **Yunzhi Li**[1] · **Mayank Goel**[1]

Accepted: 12 October 2022

Abstract
This paper presents a semantic-based interactive system that enables virtual content placement using natural language. We propose a novel computational framework composed of three components including 3D reconstruction, 3D segmentation, and 3D annotation. Based on the framework, the system can automatically construct a semantic representation of the environment from raw point cloud data. Users can then assign virtual content to a specific physical location by referring to its semantic label. Compared with traditional projection mapping which may involve tedious manual adjustments, the proposed system can facilitate intuitive and efficient manipulation of virtual content in immersive environments through speech inputs. The technical evaluation and user study results show that the system can provide users with accurate semantic information for effective virtual content placement at room scale.

Keywords Digital content placement · Semantic-based interaction · Scene understanding

1 Introduction

Embedding virtual content in physical space can provide users with instant access to digital information when and where desired, which has been a vision of research areas such as ubiquitous computing and augmented reality. However, the placement of virtual content can be a tedious task in physical 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 instance, within built environments like workplaces, it can be difficult to manually adjust projection mapping in adaptation to changes in real time. To facilitate intuitive interaction with digital content in immersive environments, this study proposes a computational framework for a semantic-based approach to digital content placement. Users can assign digital content to a location in the physical space through natural language. Compared to previous user-centric digital content placement toolkits [1–4], the integration of semantic-based input can provide benefits including:

- Interaction efficiency: Speech can potentially outperform gesture recognition since natural language has a more standard interpretation than hand gestures [5].
- Intuitiveness: Users can interact with projected digital content through natural language instead of complex sets of arbitrary commands.
- Adaptability: Semantic-based interaction systems can enable domain adaptability through training language models on domain-specific data.
- Connectedness: Users can leverage the higher level causality of semantics to compose a set of meaningful manipulation instead of creating a set of discrete and sequential commands [6].

The semantic-based input for digital content placement relies on a shared understanding of the surrounding environment. For instance, to execute the command *"project to-do list to the table,"* a projection mapping system needs to connect the symbol *"table"* to the actual location of the table in the physical world. The lack of context awareness may lead to failures in identifying spatial references from human utterances.

✉ Jingyang Liu
jingyanl@andrew.cmu.edu

Yunzhi Li
yunzhil@andrew.cmu.edu

Mayank Goel
mayankgoel@cmu.edu

[1] Carnegie Mellon University, Pittsburgh, USA

Published online: 09 November 2022

This study leverages recent developments in three-dimensional (3D) reconstruction and scene understanding for building a semantic-based digital content placement system. The system automatically constructs a linguistic representation of physical environments through three components:

- 3D Reconstruction: The component aims to capture the indoor environment and represent the environment in point clouds.
- 3D Segmentation: The captured point clouds are classified into clusters. In each cluster, point clouds share a co-planar flat surface. The surface can potentially serve as a projectable area for virtual content placement.
- 3D Annotation: We use a data-driven approach to automatically annotate each surface with semantic labels such as table, wall, floor, etc.; thus, users can refer to the target location for projection mapping using natural language.

This paper contributes to the topic of virtual content placement for immersive environments by:

- Designing a computational framework for context-aware projection mapping systems that allow users to interact with virtual content in physical space through natural language: the framework incorporates recent developments in 3D reconstruction, scene understanding for constructing semantic representations of physical environments, extended from previous works [7], we integrate a natural language understanding component for parsing and grounding user instructions.
- Developing a novel pipeline for point cloud semantic annotation: the pipeline partitions point clouds and produces semantic labels for each surface based on a deep learning architecture and clustering method. The deep learning architecture groups point clouds with similar semantics, and the clustering method captures contextual information of point clouds. Based on the assumption that co-planar point clouds can share the same semantic label, a majority voting approach was then used to unify the semantic label of point clouds within a geometrically homogeneous partition. Experiment results show that the integration of the geometric features can improve mean per-class intersection over union and accuracy by reducing over-segmentation.
- Presenting three room-scale user cases to demonstrate how the semantic-based interaction can facilitate intuitive virtual content placement and validating the usability of the system through both technical evaluation and user study.

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

2 Related work

2.1 Spatially Augmented Reality

Spatial augmented reality (SAR) uses projection mapping to augment physical objects with virtual information [8]. The concept was initially demonstrated by Raskar et al. with applications [9,10]. Previous works have explored SAR from tabletop [11] to room-scale augmentations [12]. 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, everyday surfaces [13]. Beyond unimodal interaction techniques, previous studies such as [14,15] leverage voice and gesture to create a multimodal interface that uses the strength of both input modalities, for instance, natural language is suited for descriptive techniques, while gesture can play a key role in the direct manipulation of objects [16,17].

2.2 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 [18]. Factors such as visibility [19] and legibility [20,21] investigated extensively in previous work. Context-aware systems automatically decide when, where and how much information to display based on users' current cognitive load and knowledge about their task and environment [22]. Multiple works utilize features from the real world such as point lights [23] and visual saliency [24] for adjusting the placement of virtual content.

Geometry-based systems address automated content placement based on the geometry of physical surfaces [25]. By detecting planes in the real world, AR systems can adapt virtual content to the target physical surfaces and integrate physical constraints into virtual systems. For instance, *Snap-*

ToReality extracts 3D geometric constraints from the real world for snapping virtual content to real 3D edges and planar surfaces in augmented reality [26]. DepthLab uses real-time depth data for building a variety of depth-based user interface (UI) paradigms for augmented reality [27]. Mobile AR systems such as ARKit and ARCore have encapsulated plane detection for building geometry-aware augmented reality applications.

However, to the best of our knowledge, semantic information of the real world is limited or undetected in prior works. Semantics, as a high-level abstraction of the physical environment, can serve as a both machine-readable and human-readable format for intuitive interaction with information. Early attempts have been made to map physical objects to semantic representations for natural human-machine interaction. For instance, the SHRDLU program [28] uses rule-based approaches to facilitate virtual content manipulation through natural language. "Put-that-there" work of the Architecture Machine Group [29] combines gesture with speech input for spatially annotating and referencing digital content in a physical "media room." In this work, we incorporate recent developments in scene understanding and 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.

2.3 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 [30–32]. Multiple prior works leverage 2D scene understanding for building context-aware applications in AR applications [33,34].

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 [35]. 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 [36], region-based [37], and model-fitting [38] have been proposed to group point clouds into homogeneous groups with similar local features. With the ever-growing amount of 3D shape databases [39,40] and annotated RGB-D datasets [41,42] becoming available, the data-driven approach starts to play an important role in 3D object recognition and has achieved impressive progress [43,44] . While most of the works focus on individual sampling points, some works focus on geometric features such as primitives [45] and planar surface patches [46,47] as more efficient representations for scene segmentation. Considering the factors that (1) point clouds sharing the same planar surface patches in indoor environments likely belong to the same object, (2) planar surface patches can be used to filter out noises and rectify over-segment point clouds, and (3) planar surface patches are ideal locations for projection mapping, in this work, we used surface patches as a representation for scene understanding and presented a novel approach that combines traditional geometric segmentation with deep learning models to create semantic labels for each surface patch in indoor scenes.

3 Method

In this work, 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 synchronized projectors. Projectors and the Kinect V2 sensor are calibrated using the RoomAlive Toolkit [48].

This work proposed a computational framework for a semantic-based interactive system that can automatically transform the low-level point cloud information into high-level semantic information. The system is composed of three components including 3D reconstruction (Section 3.A), 3D segmentation (Section 3.B), and 3D annotation (Section 3.C) (Fig. 1). Users can then map virtual content onto a physical surface by referring to its semantic label. For example, a user can visualize a to-do list on the wall by saying *"project to-do list to the wall."* We build a natural language understanding component (Section 3.D) to parse the user instruction into intentions and key entities such as digital content and location. By comparing the semantic similarity between the extracted entity and the semantic label of the physical environment, the system can identify the user-defined referent for digital content placement.

3.1 3D reconstruction

We obtain a 3D reconstruction of the physical environment through dense simultaneous localization and mapping (SLAM). Following the KinectFusion framework [35], 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 iterative closest point

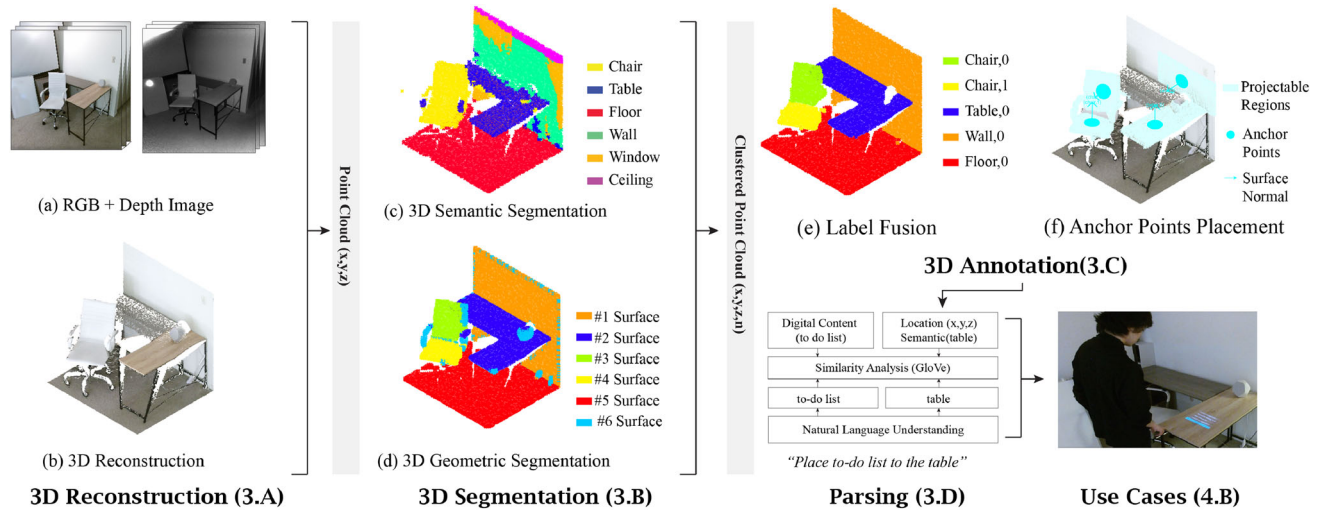


Fig. 1 The framework of the semantic-based system is composed of three components: 3D reconstruction, 3D segmentation, and 3D annotation. The 3D reconstruction component translates the acquired RGB and depth data from a Kinect sensor into point clouds. The 3D segmentation component segments point clouds into discrete surfaces with semantic label. The 3D annotation component fuses geometric and semantic labels and creates anchor point for each partitioned surface. To parse user instructions, we create a parser to extracts intent and entities from utterances for retrieving and mapping digital contents to a user-defined physical surface

(ICP) method [49]. 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.

3.2 3D segmentation

The 3D segmentation component partitions the input point clouds in two steps. The first step, 3D semantic segmentation, partitions point clouds into groups with homogeneous semantic characteristics. The second step, 3D geometric segmentation, partitions the input point clouds into groups with planes based on their geometric properties such as normal.

To perform the 3D semantic segmentation, we have trained a deep neural network [43] on the Stanford Large Scale 3D Indoor Scenes dataset [50,51]. 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 indoors, such as structural elements (*ceiling, floor, wall, beam, column, window, and door*) and furniture (*table, chair, sofa, bookcase, and board*).

The deep neural network based on PointNet [43] directly consumes point clouds and outputs the per point semantic class labels. To prepare the training data, we first split the captured point cloud into areas of 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 (Fig. 2).

Geometric features can play an important role in partitioning one shape into parts or connecting parts into a continuous shape. For instance, a new recursive formula for constructing the generalized blended trigonometric Bernstein (GPT Bernstein) can preserve G^2 and C^3 the continuity of composite curves [52,53]. A Bezier curve based on the generalized hybrid trigonometric basis function can be extended to construct symmetric rotation surfaces efficiently [54]. Bernstein-Bezier curves can be used as fitting curves for stroke segmentation and reconstruction [55]. Similarly, geometric properties of point clouds such as normal and curvature can be used to partition point clouds into regions of interest [56]. However, the existing learning-based architecture based on a regular voxel grid may fail to capture the inherent geometric properties of 3D point clouds. As a result, an entire object can be over-segmented into parts with different semantic labels without considering the local context. This work extends the learning-based semantic segmentation

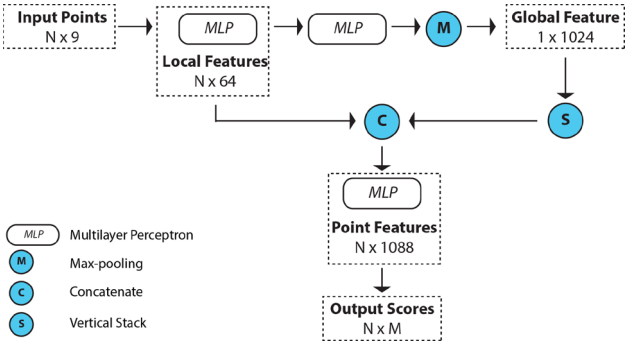


Fig. 2 Simplified architecture of PointNet [43]. (C): Concatenate (M): Max-pooling (S): Vertical stack The network samples N points within a region (in this case, we use 0.5 m by 0.5 m) as input, through a series of multi-layer-perceptrons the input N points are mapped into a 64-dimensional space, these are called local point features. Max-pooling is applied to aggregate information from all the points resulting in common global features, then the global feature is concatenated with all local features, after multi-layer-perceptrons, these combined features are used to predict M output class scores

architecture by adding a clustering process based on point cloud geometric properties.

To capture both the global structure and local contextual information, we chose the normal and the normal from origin to represent the geometric properties of 3D point clouds [57, 58]. The normal vector of each point cloud can be estimated based on its adjacent point clouds. For each point, we pick k nearest neighbors and compute the corresponding covariance matrix C, which is defined as:

$$C = \sum_{i=1}^{k} (p_i - \overline{p})^T \cdot (p_i - \overline{p}) \tag{1}$$

where $i = 1....k$, $\overline{p}$ is the centroid of k selected points (where $\overline{p} = \frac{1}{k} \cdot \sum_{i=1}^{k} p_i$). Then we estimate surface normal by finding the smallest eigenvalue λ_0 of the covariance matrix C . Assuming $\lambda_0 < \lambda_1 < \lambda_2$, we can estimate surface variation σ_i by:

$$\sigma_i = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{2}$$

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 θ between vectors can be estimated by:

$$\theta = \cos^{-1}(u, v) \tag{3}$$

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 we can find clusters of co-planar surfels from parallel surfels.

In order to obtain a robust and accurate segmentation, we used the 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 label (Fig. 3).

3.3 3D annotation

After semantic segmentation and geometric segmentation, each point is annotated with two labels—its semantic label and geometry label. However, due to the noise in the 3D reconstruction, the result of semantic segmentation is a com-

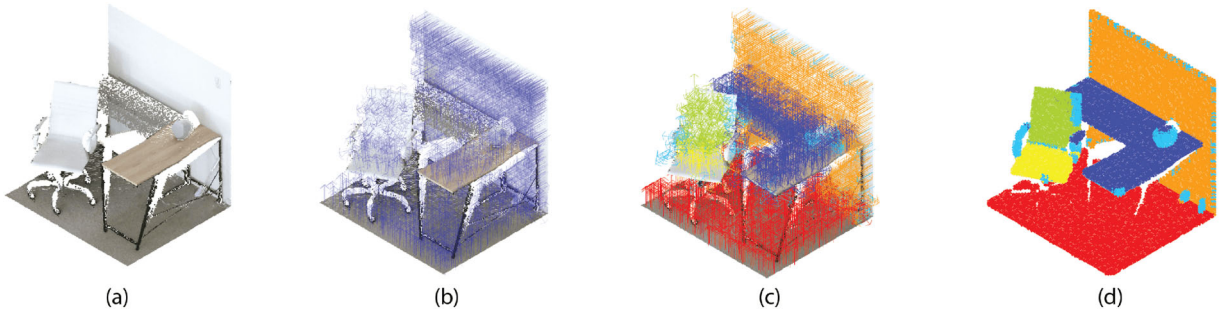


Fig. 3 Steps of 3D geometric segmentation: **a** Input point clouds (**b**) Estimate the normal of each point (**c**) Cluster point clouds by their normal and normal distance from the origin (**d**) The result of 3D geometric segmentation

bination 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

$$L_p = \underset{l \in \{1,\ldots,\mathcal{L}\}}{\arg\max} \frac{N_l}{N} \tag{4}$$

where l indexes through semantic labels and $\mathcal{L}$ 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 [59]. 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 geometry label l_{G_i}, formatted as $((x_i, y_i, z_i), N_i, l_{S_i}, l_{G_i})$.

3.4 Parsing

After the physical environment is scanned and annotated, users can interact with digital content at room scale through natural language. Semantic-based interaction techniques have been widely used in intelligent environments such as smart homes, as it provides an intuitive and flexible medium for users to interact with digital information. In this study, we chose to use a pipeline-based natural language understanding component, since we need to both interpret the user's intent and extract the entities of instructions. The component first converts audio input into a written representation. Then the user's unstructured instructions are transformed into an action language which is structured as *"Project A to B."* In this phrase, "*A*" represents the digital content to be projected, and "*B*" is the semantic label of the location for projection mapping.

We jointly trained a model for intent classification and slot filling based on the joint Bidirectional Encoder Representations from Transformers (BERT) model [60]. The model extends the BERT model by defining a joint loss function. We use the CMU Sphinx API for speech recognition which converts the user's voice input into a text representation in sentences. The input sentence is then tokenized into words using the Natural Language Toolkit (NLTK) library. A special classification word ([CLS]) is added as the first token. Each token is featurized with dense features based on a pre-trained BERT model [61] for word embedding. At a sentence level, the joint-Bert model uses the first special token denoted as h_0 to obtain the sentence intent classification probability

$$P_c = softmax(W^i h_0 + b^i) \tag{5}$$

where W^i and b_i are model parameters.At a token level, to classify labels over a sequence of $S = (t_0, t_1, t_2, .., t_n)$, the final hidden states of tokens except for the first special token are fed into a softmax layer to predict slot filling tags. The categorical probability for the token x_n can be represented as:

$$P_n^s = softmax(W^s h_j + b^s) \tag{6}$$

where W^s and b_s are model parameters, h_j is the final hidden stage of token t_j, for $j = 1, ..., n$.

We use a 3-layer bidirectional transformer [62] to encode the contextual information for each token through self-attention [63], and generate contextual embedding. The objective of learning is to jointly find the (W_i, b_i, W_s, b_s) by minimizing the total loss L_t, which is defined as:

$$L_t = L_p + L_s \tag{7}$$

where L_p and L_s are the cross-entropy loss for slot filling and intent detection, respectively. To improve the entity labeling, we add a conditional random field (CRF) layer for modeling slot label dependencies [64], The addition of a CRF layer can provide constraints to the final predicted labels to ensure that the label between slots is valid, the constraints are learned automatically from the training data. (Fig. 4).

The training configuration for building the component is set as 300 for epochs. The learning rate is set as 10^{-3}. The batch size is set to 32. The number of layers of the bidirectional Transformer is set to 3. We use Adam optimizer for the training process.

To build the Natural Language Understanding (NLU) component, we created a task-specific custom language dataset by collecting non-expert user data through surveys. The goal of the survey is to capture potential requests for digital content placement in indoor settings. 4 types of pre-designed intents such as *"ShowLabel," "ShowContent," "EditContent"* and *"HideContent"* were included. The intent *"ShowLabel"* displays a semantic label on each projectable surface. The intent *"ShowContent"* and *"HideContent"* show and hide digital content on a specified surface, respectively. For each intent, users provided utterance examples for completing a digital content placement task. For instance, a user can say *"project time to the wall"* to display the current time on the wall surface.

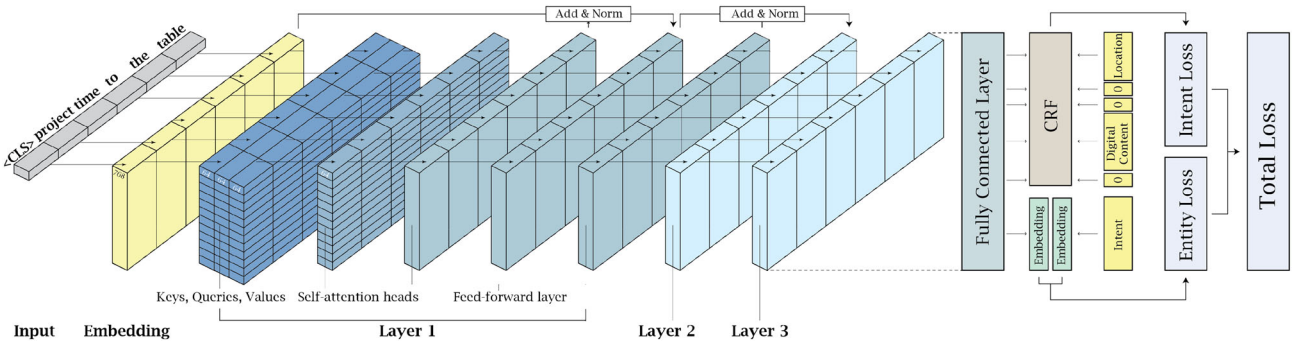


Fig. 4 The schematic representation of the natural language understanding architecture. The instruction *"project time to the table"* has the intent *"ShowContent"* and entity *"DigitalContent"* with value "time"and *"Location"* with value "table."

Table 1 Intents, entities and examples of the dataset

Intent	Utterance Example	Entity
ShowLabel	Project semantic labels	N/A
ShowContent	Project **time** to the **wall**	DigitalContent:time , Location:wall
ShowContent	Project **to-do list** to the **table**	DigitalContent:to-do list , Location:table
EditContent	Edit the **spatial note** on the **whiteboard**	DigitalContent:spatial note , Location:whiteboard
HideContent	Clear the **spatial note** on the **wall**	DigitalContent:spatial note , Location:wall

Both *"ShowContent"* and *"HideContent"* intents contain entities that can be classified into two types *"DigitalContent"* and *"Location."*The entity *"DigitalContent"* is specific to the type of digital content in the digital content repository, including *"time," "schedule," "weather,"* etc. The entity *"location"* represents the semantic label of the physical surface for projection mapping including *"table," "wall," "desk,"* etc. We collected 169 unique sentences and 25 entities. The example of intents and utterances can be seen in Table 1. In order to evaluate the NLU component, we performed a ten-fold cross-validation on the dataset. The precision, recall, and F1 score of intent recognition are 98.2%, 97.1%, 97.6%, respectively. The precision, recall, and F1 score of slot filling 88.0%, 84.0%, 86.0%, respectively.

After extracting entities from utterances, we build two matches: (1) a match between the digital content tag and the extracted digital content entity; (2) a match between the semantic label of physical objects and the extracted location entity. By constructing the matches, the system can retrieve the user-specified digital content from the digital content repository and project the content to the target location. To find the best match, we first transform words into 300-D vectors using GloVe [65]. The extracted location entity is transformed into a vector $v_{loc} \in \mathbb{R}^{1\times 300}$. For instance, the semantic label of the physical environment is transformed into vector $v_{sem,i} \in \mathbb{R}^{1\times 300}, i \in (1, N)$, where N denotes the number of the physical surfaces annotated with semantic labels. The semantic similarity is calculated by

$$Sim(v_{loc}, v_{sem,i}) = \frac{v_{loc} \cdot v_{sem,i}}{||v_{loc}||||v_{sem,i}||} \tag{8}$$

where $||v_{loc}||$ is the Euclidean norm of vector v_{loc} and $||v_{sem,i}||$ is the Euclidean norm of vector $v_{sem,i}$.

We set the target location for digital content placement by selecting the physical object with the largest semantic similarity value of the semantic-location pair. Similarly, the user-specified digital content such as *"time"* can be retrieved based on the similarity score between the extracted digital content entity and the tag of the digital content in the repository. The repository stores digital content such as temperature, weather, to-do list, time, etc. Each content contains a tag, anchor point, dimension, and graphics. Contents can be retrieved by its tag. The anchor point and dimension are used for calculating the transformation matrix between the model space and the physical space for projection mapping.

4 Result

4.1 User interface

We develop a proof-of-concept prototype for end users to set up a semantic-based virtual content placement system. The prototype allows end users to (1) stream the 3D scan of the environment for digital content placement (2) partition the captured scene into groups and annotate each group with a semantic label (3) intuitively modify the automatically

Fig. 5 Semantics Mapping UI (**a**) the visualization of 3D segmentation (1) the visual representation of the physical world and its semantic segmentation result (2) the control panel for scanning, segmenting, and annotating surfaces (3) the information panel for each segmented surface, the panel shows the default position for digital content placement of the surface | (**b**) The system highlights the anchor point of the wall surface for digital content placement, user can edit the position through manual input

created semantic labels with real-time visual feedback. The functions are supported by three key components:

- 3D Scanning: the 3D scanning component connects Kinect sensors through the Kinect Software Development Kit (SDK). In this study, we use Kinect V2 for 3D reconstruction, each Kinect sensor is connected to an Intel NUC computer. The depth and infrared data are then streamed to a server computer through an Ethernet connection. Then the component reconstructs the environment by fusing depth images based on an implementation of KinectFusion in C++ with the point cloud library (PCL). The corresponding RGB color images are used to reconstruct the surface texture. The 3D scanning component represents the reconstructed scene in a downsampled point cloud format and outliers are removed by statistical outlier removal algorithms. The component saves the point cloud file as a Point Cloud Data (PCD) format loaded by the user interface and visualized through the Open3D library for Python [66].
- 3D Segmentation: After loading the point cloud of the environment, users can partition the point cloud into clusters annotated with semantic labels. The trained point cloud segmentation model is loaded through PyTorch. After semantic segmentation, the geometric segmentation and 3D annotation process are implemented in Python based on the Open3D library. Each point is labeled with a unique semantic label. Users can explore the segmentation result intuitively in the visualization panel after setting the visibility of the result as true through a click-box. Each semantic cluster is colorized to improve visual readability (Fig. 5). The scene we tested is of $35m^2$ and 196809 points. The computation time for the pipeline is measured on a 4 HZ CPU and RTX 2080 Ti GPU. The bulk of the time was spent on semantic segmentation which takes $2.7s$ and geometric segmentation takes $0.6s$ and the 3D annotation process takes $0.1s$.
- Editing: The reconstructed environment is organized into clusters of point clouds sharing the same semantic label. Users can select a group of point clouds by their semantic label through a drop-down menu. The target of projection mapping for each cluster is initially set to the geometric center of all points in the cluster and the normal or the target is set to the weighted average of the normal of all points in the cluster. Users can visualize the projection mapping target for each cluster in the visualization panel, both the target location and normal can be modified through manual input (Fig. 5).

In addition, the prototype provides support for human motion tracking and projector-camera calibration. Users can use proxemic information including the position and orientation of people to invoke the projection mapping system. For example, users can define a threshold for the distance between a surface and the position of the detected people, while the distance is within the threshold, the digital content assigned to the surface will be projected onto the surface. The display panel is used for projector and camera pairing. Users can either manually input the position and orientation of the camera and projector or use auto-calibration. In the auto-calibration mode, the projector casts structured light patterns onto the scene. Two Kinect Sensors use structured light sequence to find correspondence and determine the projector-camera pair as a stereo camera setup [67]. Then

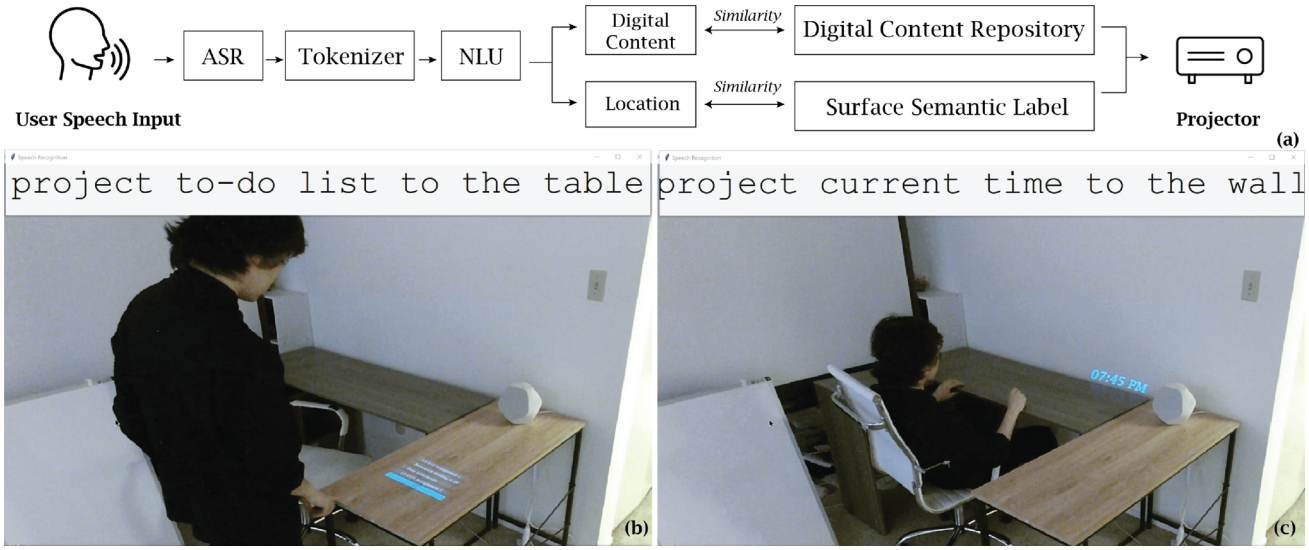


Fig. 6 The semantic-based placement system interprets the user's voice command and projects the specified virtual content onto the surface with the targeted semantic label (**a**) the pipeline for the semantic-based placement system (ASR - Automatic Speech Recognition, NLU - Natural Language Understanding) (**b**) the user uses speech input to place 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" on the surface with the semantic label *"wall"*

we use the Iterative Closest Point (ICP) algorithm [68] to find the geometric transformation between the Kinect sensors and the scanned scene. Finally, the sensor, the projector, and the scanned scene are registered in a shared coordinate system.

4.2 Example applications

To illustrate the capability of the semantic-based content placement system, three applications, including (1) personal assistant (2) sensor reading visualization, and (3) spatial notes, are presented. This section will also describe how semantic-based interaction can provide users with intuitive and efficient access to virtual content in physical environments.

4.2.1 Personal assistant

In this scenario, the system provides users with instant access to a wealth of information such as time, schedule, and to-do list by projecting the virtual content to a physical surface specified by the user through natural language. For instance, the user says utterances such as *"project current time to the wal."* The Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) pipeline interprets the user intent as *"ShowContent"* and extracts entities including"time" and "wall." The entity *"time"* is classified as *"DigitalContent"*, the entity *"wall"* is classified as *"Location."* Based on the cosine similarity, the system extracts the digital content of the highest matching score from the repository. The system can identify the target from all annotated surfaces by comparing the cosine similarity between the entity name *"wall"* and the semantic labels of the annotated physical surfaces. Finally, the system calculates the transformation matrix between the model space and the world space. The projector then projects the digital content *"time"* onto the physical surface with a semantic label of *"wall"* (Fig. 6). The system provides users with a natural interface for visualizing digital information at a specified location. The intuitive control of peripheral information display can enhance a user's ability to manage multiple digital information simultaneously.

4.2.2 Sensor reading visualization

It can be difficult for users to read the status of a sensor locally if the sensor is not equipped with a display. The simple implementation of the semantic-based content placement system allows users to visualize sensor information at a nearby location. The visualization of sensor readings can be used to support context-aware service. For instance, the reading of a plant sensor projected onto a nearby physical surface can inform the user whether the soil needs watering or not. A user can use the utterance *"project moisture reading to the wall"* while watering the plant in this scenario. The entity *"moisture reading"* is classified as *"DigitalContent."* The entity *"wall"* is classified as *"Location."* We use semantic-based data management and processing middleware for modeling and describing connected devices and sensors. Based on the query language SPARQL, the reading of the moisture sen-

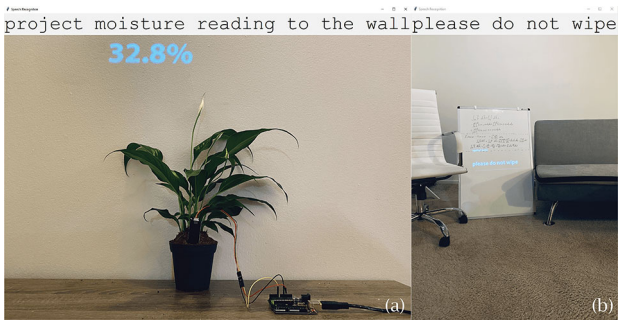


Fig. 7 **a** A user places the moisture sensor reading on the wall near the plant (**b**) A user leaves a spatial note on the whiteboard to remind other users not to wipe the written content

sor can be retrieved from a relational database and projected to the physical surface annotated as *"wall."* The display of moisture sensor reading at a nearby wall surface supports the user's in-situ decision making on tasks such as plant watering (Fig. 7).

4.2.3 Spatial notes

Users can use the semantic-based tool to annotate physical objects with additional virtual information using natural language. The customized labels attached to physical objects can serve as a communication medium for co-located users. For instance, a user can leave spatial notes such as *"please do not wipe"* on the whiteboard to remind other users who share the room. The user can give voice commands like *"project a spatial note to the whiteboard"* to post such notes. The *"spatial note"* is recognized as a *"DigitalContent"* entity associated with a *"Location"* entity - *"whiteboard."* Then the user can edit the content of the spatial note on the door by saying the utterances *"edit the spatial note on the whiteboard."* The spatial note associated with the whiteboard becomes editable, and the user can input the note content *"please do not swipe"* through natural language. Finally, the spatial note is presented in text and projected onto the whiteboard (Fig. 7). The system provides users with an efficient tool for annotating information on physical objects. The projected public viewable information can serve as a virtual post-it adhered to physical environments to support collaboration between users in workplace environments.

5 Evaluation

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

Table 2 Model evaluation on self-scanned point cloud

Scene	Method	Mean IoU	Accuracy
Scene 1	PointNet	66.3%	96.3%
	Ours	71.1%	94.4%
Scene 2	PointNet	71.3%	93.7%
	Ours	86.4%	96.5%
Scene 3	PointNet	46.2%	77.8%
	Ours	65.7%	78.3%

5.1 Model performance

To evaluate the performance of our system, we compare the accuracy of the PointNet approach and our proposed approach using two criteria including mean intersection over union (IoU) and accuracy. The result can be seen in Table 2 . The mean accuracy over all classes was calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

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:

$$MeanIoU = \frac{\sum_{i=1}^{k} \frac{TP_i}{FP_i + FN_i + TP_i}}{k} \tag{10}$$

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 the PointNet approach with our approach which uses the majority voting to unify the semantic label of points sharing a plane.

The results show that parts of the sofa point clouds are misclassified into the chair group, and parts of the wall are misclassified into the window group in Scene III. The reason may be attributed to (1) environmental factors and (2) semantic factors.

– Environmental factors: In the segmentation process, color features are encoded as the attribute of a point;

thus, the semantic segmentation is sensitive to the environmental factors that may affect the object's color, such as lighting and shading. For instance, the uneven lighting on the wall causes local highlights and color changes. The local color changes can result in over-segmentation.

– Semantic factors: A single object or objects of the same semantic class can be made of different materials which can result in misclassifications. For instance, in Scene III, most of the point clouds on the whiteboard frame are misclassified as windows. Unlike the board surface made of matte white plastics, the frame of the whiteboard is made of reflective metal. The reflectance of metal frames can cause color variation and noises in 3D reconstruction. Since the color features and point cloud normal are encoded as features of points in the semantic segmentation, the captured whiteboard frame point clouds with distinguished color and normal are over-segmented. Due to the similarity between the whiteboard frame and window frames, in this case, the whiteboard frame point clouds are misclassified into window class. The error was not rectified by the geometric segmentation and majority voting. The reason can be that the normal of the frame point clouds are different from the surface area due to the noise caused by reflections in 3D reconstruction.

As depicted in Fig. 8, most of the point clouds that were misclassified by the PointNet approach are corrected by the post-processing process which unifies the label of point clouds on the same plane. The result shows our system can produce a more uniform and accurate result on our own environment point cloud. However, the 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 item, the plane constraints added for segmentation 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.

5.2 User study

In order to evaluate the effectiveness of the semantic-based input, we invited nine undergraduate and graduate students to perform the digital content placement task with and without the system. Each participant was asked to relocate a projected virtual content to a specified location. The task was completed in two set-ups. In the first set-up, we used a traditional projection mapping system. The participants move the

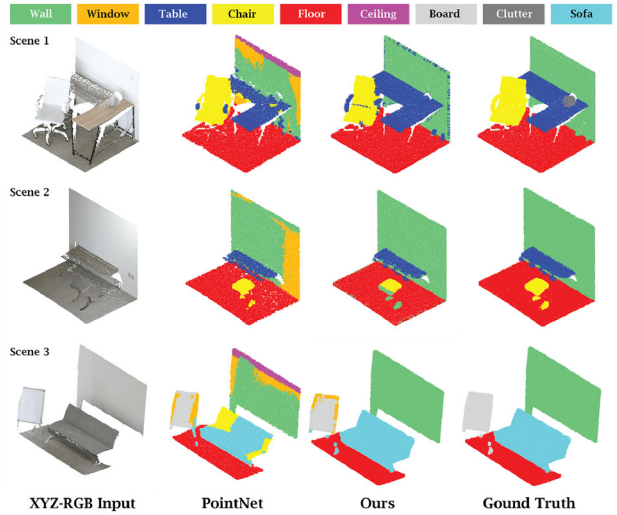


Fig. 8 Qualitative model evaluation of the PointNet segmentation approach and our proposed approach on the point cloud of scanned scenes

digital content through a graphical user interface (GUI) with four buttons for moving up, moving down, moving right, and moving left. In the second set-up, the participant tested the semantic-based input. To move digital content, the participant can define the target location by referring to the surface semantic label through natural language. The system then parses instructions and automatically projects the content to the target surface. The task is considered complete if the observer confirms the virtual content is placed at the specified location. We conducted ten iterations of tests and recorded the 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 = 5.88, Std = 1.05) was significantly faster (p<.001) overall compared to the traditional projection mapping system (mean = 12.89, Std = 4.21).

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 semantic label directly. The distance between the current and the target surface does not affect the completion time. The completion time depends on (1) the processing time for instruction recognition and (2) the error rate of the Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) component. For a traditional projection mapping system, the users need to move the content incrementally using a keyboard. The relo-

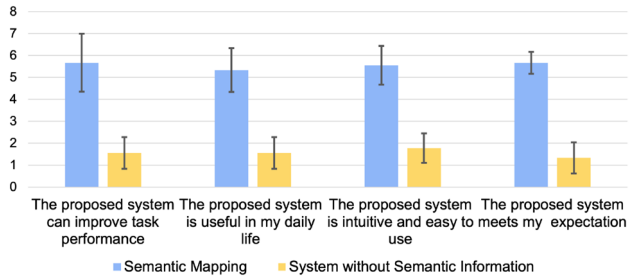


Fig. 9 Results of the Likert-scale survey questions

cation of virtual content can be 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 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.

As depicted in Fig. 9 , all four questions got positive results from our participants. More specifically, all participants agreed that our system can improve their task performance (mean 5.67, Std 1.32). More than half of the participants believe our system is useful in their daily lives (mean 5.33, Std 1.0). 7 out of 9 participants agreed that our system is intuitive and easy to use (mean 5.56, Std 0.88). And all participants agreed that the performance of our system met their expectations (mean 5.67, Std 0.50). Compared to a speech input projection system without semantic information, our system is significantly better from all four perspectives ($p < 0.001$ for all four questions).

6 Conclusion and future work

This study presents a computational framework for a semantic-based interactive system for digital content placement in immersive environments. Enabled by the system, users can directly place virtual content onto a physical surface by referring to its semantic label. Compared to other interactive modalities, the integration of semantic-based input can provide benefits such as efficiency, intuitiveness, adaptability, and connectedness. To construct a semantic representation of the physical environment, this work proposes a novel pipeline for automatically annotating the physical environment with semantic labels. The pipeline incorporates the geometric properties of point clouds into a learning-based architecture for embedding both the global and local contextual information. Based on the technical evaluation result, the pipeline improves the mean IoU and accuracy in point cloud segmentation.

To test the usability, we evaluated the system's accuracy and conducted a user study. We compared the proposed semantic-based projection mapping system with the graphics-based projection mapping system in the user study. In the semantic-based scenario, a user assigns digital content to a location via a speech interface. For instance, a user can place a digital clock on a table by saying *"project time to the table."* In the graphics-based scenario, a user moves projected digital content to the target location via a GUI with four buttons for move-up, move-down, move-right, and move-left. 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.

In future work, we would like to leverage semantic representations of physical environments to construct a high-level scene understanding. For instance, by constructing a semantic graph to encode semantic relationships between physical objects and users, the system may be able to process complex unstructured queries and identify an optimal location for digital content placement.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declarations

Conflict of Interest The authors declare that they have no conflicts of interest.

Data availability Datasets for this research are openly available at locations cited in the reference section [50,51].

References

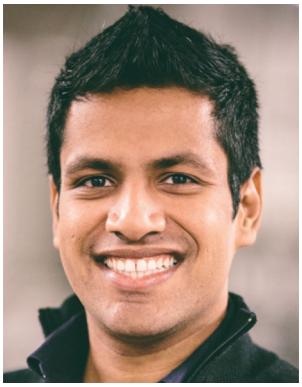
1. Xiao, R., Harrison, C., and Hudson, S. E.: WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 879–888, (2013)
2. Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., and Shapira, L.: RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In: Proceedings of the 27th annual ACM symposium on User interface software and technology, pp. 637–644, (2014)
3. Fender, A., Herholz, P., Alexa, M., and Müller, J.: OptiSpace: automated placement of interactive 3D projection mapping content. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–11, (2018)
4. Fender, A., and Müller, J.: 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, pp. 303–314, (2019)

5. Weimer, D., Ganapathy, S.K.: A synthetic visual environment with hand gesturing and voice input. ACM SIGCHI Bull. **20**, 235–240 (1989)
6. Marsh, E., Wauchope, K., and Gurney, J.: Human-machine dialogue for multi-modal decision support systems. In: Proceedings of the AAAI Spring Symposium on Multi-Media Multi-Modal Systems. Citeseer, (1994)
7. Liu, J.: Semantic mapping: a semantics-based approach to virtual content placement for immersive environments. In: 2021 17th International Conference on Intelligent Environments (IE). IEEE, pp. 1–8, (2021)
8. Raskar, R., Welch, G., and Fuchs, H.: Spatially augmented reality. In: Proceedings of the International Workshop on Augmented Reality: Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes, pp. 63–72, (1999)
9. Raskar, R., Welch, G., Low, K.-L., and Bandyopadhyay, D.: Shader lamps: animating real objects with image-based illumination. In: Eurographics Workshop on Rendering Techniques, pp. 89–102, (2001)
10. Raskar, R., Baar, J. V., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C.: iLamps: geometrically aware and self-configuring projectors. pp. 7–es, (2006)
11. Wilson, A. D.: Depth-sensing video cameras for 3d tangible tabletop interaction. In: Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07), pp. 201–204, (2007)
12. Rekimoto, J., and Saitoh, M.: Augmented surfaces: a spatially continuous work space for hybrid computing environments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 378–385, (1999)
13. Harrison, C., Benko, H., and Wilson, A. D.: OmniTouch: wearable multitouch interaction everywhere. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 441–450, (2011)
14. Lucente, M., Zwart, G.-J., and George, A. D.: Visualization space: a testbed for deviceless multimodal user interface. In: Intelligent Environments Symposium, vol. 98, (1998)
15. Kaiser, E., Olwal, A., McGee, D., Benko, H., Corradini, A., Li, X., Cohen, P., and Feiner, S.: Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In: Proceedings of the 5th International Conference on Multimodal interfaces, pp. 12–19, (2003)
16. Cohen, P. R.: The role of natural language in a multimodal interface. In: Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology, pp. 143–149, (1992)
17. Billinghurst, M.: Put that where? voice and gesture at the graphics interface. Acm Siggraph Comput. Graphics **32**(4), 60–63 (1998)
18. Bell, B., Feiner, S., and Höllerer, T.: View management for virtual and augmented reality. In: Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, pp. 101–110, (2001)
19. Fender, A., Lindlbauer, D., Herholz, P., Alexa, M., and Müller, J.: 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, pp. 611–621, (2017)
20. Bell, B., Feiner, S., and Höllerer, T.: View management for virtual and augmented reality. In: Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, pp. 101–110, (2001)
21. Tatzgern, M., Orso, V., Kalkofen, D., Jacucci, G., Gamberini, L., Schmalstieg, D.: Adaptive information density for augmented reality displays. In: IEEE Virtual Reality (VR) **2016**, 83–92 (2016)
22. Lindlbauer, D., Feit, A. M., and Hilliges, O.: Context-aware online adaptation of mixed reality interfaces. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, pp. 147–160, (2019)
23. Ahuja, K., Pareddy, S., Xiao, R., Goel, M., and Harrison, C.: Lightanchors: appropriating point lights for spatially-anchored augmented reality interfaces. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, pp. 189–196, (2019)
24. Grasset, R., Langlotz, T., Kalkofen, D., Tatzgern, M., Schmalstieg, D.: Image-driven view management for augmented reality browsers. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR) **2012**, 177–186 (2012)
25. Gal, R., Shapira, L., Ofek, E., Kohli, P.: FLARE: fast layout for augmented reality applications. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR) **2014**, pp. 207–212 (2014)
26. Nuernberger, B., Ofek, E., Benko, H., and Wilson, A. D.: Snaptoreality: aligning augmented reality to the real world. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1233–1244, (2016)
27. Du, R., Turner, E., Dzitsiuk, M., Prasso, L., Duarte, I., Dourgarian, J., Afonso, J., Pascoal, J., Gladstone, J., Cruces, N. et al.: Depthlab: real-time 3d interaction with depth maps for mobile augmented reality. In: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, pp. 829–843, (2020)
28. Winograd, T.: Understanding natural language. Cogn. Psychol. **3**(1), 1–191 (1972)
29. Bolt, R. A.: Put-that-there voice and gesture at the graphics interface. In: Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, pp. 262–270, (1980)
30. Long, J., Shelhamer, E., and Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440, (2015)
31. Dai, J., He, K., and Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3150–3158, (2016)
32. Li, Y., Qi, H., Dai, J., Ji, X., and Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2359–2367, (2017)
33. Barreira, J., Bessa, M., Barbosa, L., Magalhães, L.: A context-aware method for authentically simulating outdoors shadows for mobile augmented reality. IEEE Trans. Vis. Comput. Graph. **24**(3), 1223–1231 (2017)
34. Chen, L., Tang, W., John, N. W., Wan, T. R., and Zhang, J. J.: Context-aware mixed reality: a learning-based framework for semantic-level interaction. In: Computer Graphics Forum, vol. 39, pp. 484–496, (2020)
35. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., and Davison, A.: 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, pp. 559–568, (2011)
36. Sappa, A. D., and Devy, M.: Fast range image segmentation by an edge detection strategy. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling, pp. 292–299, (2001)
37. Jagannathan, A., Miller, E.L.: Three-dimensional surface mesh segmentation using curvedness-based region growing approach. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12), 2195–2204 (2007)
38. Schnabel, R., Wahl, R., and Klein, R.: Efficient RANSAC for point-cloud shape detection. In: Computer Graphics Forum, vol. 26, pp. 214–226, (2007)
39. Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., and Su, H.:

Shapenet: an information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015
40. Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1746–1754, (2017)
41. Silberman, N., Hoiem, D., Kohli, P., and Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, pp. 746–760, (2012)
42. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y.: Matterport3d: learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
43. Qi, C. R., Su, H., Mo, K., and Guibas, L. J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
44. Qi, C. R., Yi, L., Su, H., and Guibas, L. J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5099–5108 (2017)
45. Fayolle, P.-A., Pasko, A.: Segmentation of discrete point clouds using an extensible set of templates. Visual Comput. **29**(5), 449–465 (2013)
46. Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., and Pajarola, R.: Object detection and classification from large-scale cluttered indoor scans. In: Computer Graphics Forum, vol. 33, no. 2. Wiley Online Library, pp. 11–21 (2014)
47. Sun, Y., Miao, Y., Chen, J., Pajarola, R.: Pgcnet: patch graph convolutional network for point cloud segmentation of indoor scenes. Visual Comput. **36**(10), 2407–2418 (2020)
48. Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., and Shapira, L.: RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, pp. 637–644 (2014)
49. Besl, P. J., and McKay, N. D.: Method for registration of 3-D shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures, vol. 1611, pp. 586–606 (1992)
50. Armeni, I., Sax, S., Zamir, A. R., and Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
51. Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1534–1543 (2016)
52. Bashir, U., Abbas, M., Ali, J.M.: The g2 and c2 rational quadratic trigonometric bézier curve with two shape parameters with applications. Appl. Math. Comput. **219**(20), 183–197 (2013)
53. Maqsood, S., Abbas, M., Miura, K.T., Majeed, A., Iqbal, A.: Geometric modeling and applications of generalized blended trigonometric bézier curves with shape parameters. Adv. Diff. Equ. **2020**(1), 1–18 (2020)
54. BiBi, S., Abbas, M., Misro, M.Y., Hu, G.: A novel approach of hybrid trigonometric bézier curve to the modeling of symmetric revolutionary curves and symmetric rotation surfaces. IEEE Access **7**, 779–792 (2019)
55. Liao, C.-W., Huang, J.S.: Stroke segmentation by bernstein-bezier curve fitting. Pattern Recogn. **23**(5), 475–484 (1990)
56. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. Robot. Autonomous Syst. **56**(11), 927–941 (2008)
57. Berkmann, J., Caelli, T.: Computation of surface geometry and segmentation using covariance techniques. IEEE Trans. Pattern Anal. Mach. Intell. **16**(11), 1114–1116 (1994)
58. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: IEEE Visualization, VIS **2002**, 163–170 (2002)
59. Katz, S., Tal, A., and Basri, R.: Direct visibility of point sets. pp. 24–es (2007)
60. Chen, Q., Zhuo, Z., and Wang, W.: Bert for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909 (2019)
61. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
62. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
63. Li, C., Li, L., and Qi, J.: A self-attentive model with gate mechanism for spoken language understanding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3824–3833 (2018)
64. Zhou, J., and Xu, W.: End-to-end learning of semantic role labeling using recurrent neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Vol 1: Long Papers), pp. 1127–1137 (2015)
65. Pennington, J., Socher, R., and Manning, C. D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
66. Zhou, Q.-Y., Park, J., and Koltun, V.: Open3D: a modern library for 3D data processing. arXiv:1801.09847 (2018)
67. Scharstein, D., and Szeliski, R.: High-accuracy stereo depth maps using structured light. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings., vol. 1. IEEE, pp. I–I (2003)
68. Besl, P. J., and McKay, N. D.: Method for registration of 3-d shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures, vol. 1611. International Society for Optics and Photonics, pp. 586–606 (1992)

Jingyang (Leo) Liu is a Ph.D. student in Computational Design at Carnegie Mellon University (CMU). Prior to CMU, He received the M.S. in matter design computation from Cornell University, where he worked as a Research Associate responsible for National Science Foundation (NSF) supported research on programmable kirigami - an approach to transform 3D geometries into 2D patterns involving only folding and cutting. His current research interest focus on computational geometry with its applications in robotic perception, motion planning, and spatial computing within the built environment.

Yunzhi Li is a Ph.D. student at Carnegie Mellon University's Human-Computer Interaction Institute. Prior to CMU, he received his M.Sc. from Georgia Institute of Technology and B.Eng. from University of Chinese Academy of Sciences. Yunzhi's current research interests are healthcare applications, assistive technologies, and novel sensing techniques.



Mayank Goel My research focuses on designing, implementing, and testing new sensing systems. I typically focus on repurposing and extending the capabilities of sensors and devices around us. This approach allows us to add various functionalities to our daily-use devices with negligible hardware modifications. I am interested in solving problems in various domains, including health sensing, technologies for global development, and novel interactions. Many times solutions in such domains need to be used and evaluated by the end user in their daily lives. Consequently, I collaborate closely with medical professionals, bio-engineers, and designers to develop end-to-end solutions that can be immediately used and evaluated outside the laboratory environment.