

# Retargetable AR: Context-aware Augmented Reality in Indoor Scenes based on 3D Scene Graph

Tomu Tahara\*

Takashi Seno

Gaku Narita

Tomoya Ishikawa

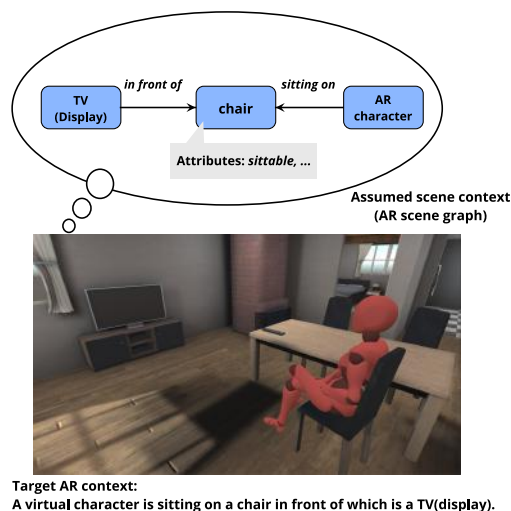
R&D Center  
Sony Corporation

Figure 1: **Retargetable AR**. The context of an AR scene in which "A virtual character is sitting on a chair in front of which is a TV" is expressed as an abstract AR scene graph based on the relationships among objects (left). The framework can retarget the AR scene to various real scenes by comparing the AR scene graph with 3D scene graphs constructed in each of the scenes (right).

## ABSTRACT

In this paper, we present **Retargetable AR**—a novel AR framework that yields an AR experience that is aware of scene contexts set in various real environments, achieving natural interaction between the virtual and real worlds. To this end, we characterize scene contexts with relationships among objects in 3D space, not with coordinates transformations. A context assumed by an AR content and a context formed by a real environment where users experience AR are represented as abstract graph representations, i.e. scene graphs. From RGB-D streams, our framework generates a volumetric map in which geometric and semantic information of a scene are integrated. Moreover, using the semantic map, we abstract scene objects as oriented bounding boxes and estimate their orientations. With such a scene representation, our framework constructs, in an online fashion, a 3D scene graph characterizing the context of a real environment for AR. The correspondence between the constructed graph and an AR scene graph denoting the context of AR content provides a semantically registered content arrangement, which facilitates natural interaction between the virtual and real worlds. We performed extensive evaluations on our prototype system through quantitative evaluation of the performance of the oriented bounding box estimation, subjective evaluation of the AR content arrangement based on constructed 3D scene graphs, and an online AR demonstration. The results of these evaluations showed the effectiveness of our framework, demonstrating that it can provide a context-aware

\*e-mail:{tomu.tahara, takashi.seno, gaku.narita, tomoya.ishikawa}@sony.com

AR experience in a variety of real scenes.

**Index Terms:** Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Artificial intelligence—Computer vision—Computer vision tasks—Scene understanding

## 1 INTRODUCTION

With the recent advances in computer vision and graphics technologies, natural fusion of the virtual and real worlds by Augmented Reality (AR) / Mixed Reality (MR) systems is reaching a practical level. To integrate virtual objects into real environments accurately and naturally, a deep understanding of physical environments becomes even more crucial for AR/MR systems.

In conventional approaches, scene understanding in AR has emphasized geometric and photometric registration between the virtual and real worlds [26, 57]. Geometric registration means that virtual content is displayed in a real environment without misalignments. For example, a virtual entity that is aligned on the ground without floating in the air or that is occluded by real objects gives users a more realistic impression. Photometric registration makes lighting and shading of virtual objects the same as the real ones, which facilitates the natural representation of virtual objects as well. Recently, the increased accuracy of camera tracking technology [14, 45], such as visual SLAM (Simultaneous Localization and Mapping), and lighting estimation [3, 42] has relaxed these registration problems, and it can be said that methods to resolve the difficulties have reached a more practical level [22, 35].

While such relaxation allows ubiquitous AR experiences in a range of settings from specially designed environments to ordinary scenes such as living rooms at home, the need for semantic registration to allow more realistic arrangements or interactions of virtual

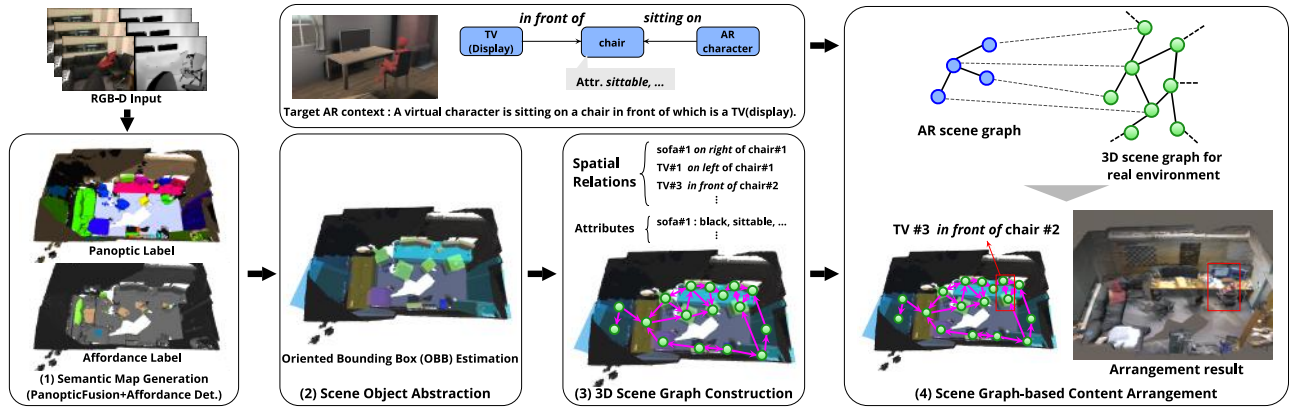


Figure 2: Overview of the proposed framework.

content in different real environments has been raised [30]. Semantic registration means that a defined context for virtual content is preserved and matched to that of a real AR scene as well. When an AR content developer designs an AR space including a virtual lamp placed on a table, for example, a semantically registered AR space requires placing a virtual light not on a detected plane in the scene, but on the recognized table. We focus on the understanding of the semantic information and propose a new AR framework that enables virtual content to interact with a real environment while being aware of its context.

Although object recognition techniques such as semantic segmentation and semantic mapping have been attracting more and more attention by the research community in order to understand real scenes deeply, there is little prior work in which AR applications were developed while being aware of rich semantic information. Commonly used SDKs (ARCore [35], ARKit [22], and MRTK [11]) for AR application development provide functions to arrange virtual objects or interact with virtual and real objects using geometry-based scene understanding. They detect planar regions in a scene from its measured geometric information and assign object categories such as walls, floors, and tables to them. Such an approach is capable of placing virtual objects onto detected regions. However, it is difficult to deal with interactions in more complex situations that require us to consider the contexts around objects, such as the relations among them, or to discriminate individual objects.

On the other hand, some conventional approaches have adopted coordinate sharing between a real space for an AR experience and a virtual space for content creation for context-aware AR [6, 25, 31, 47]. The most fundamental approaches utilize AR markers [25] or image feature tracking [47]. Such methods set the poses of the virtual object in advance while considering how to present AR content at the time of clue detection. In this case, AR content designers create AR applications by understanding AR scene contexts based on geometric and semantic information of the scene instead of AR systems. Also, the recently proposed Visual Positioning System (VPS) [4, 5] enables content authoring in large-scale scenes. However, these methods have difficulty in dealing with layout change of objects in the scenes, or unknown scenes such as the living rooms of individual users.

In this paper, we present *Retargetable AR*—a novel AR framework that yields a context-aware AR experience in various scenes based on scene contexts defined by AR content designers and enables natural interaction between the virtual and physical worlds. The essential feature of the framework is to represent the context of AR content as an abstract 3D scene graph, not as some specific coordinate transformation. The 3D scene graph symbolizes scene context in the form of a 3D directed graph. The 3D graph consists of nodes corresponding to individual objects and directed edges

associated with relations between objects. Then, by matching the context of AR content with that of the real scene in the 3D scene graphs, we can achieve semantic-aware interactions using virtual content without specific scene coordinates. In addition, motivated by AR content like a virtual character who takes natural actions in a real environment as humans do (Fig. 1), our framework recognizes additional high-level semantics in real scenes and includes them in the node attributes, which contributes to natural AR representation with interactions.

To realize such an AR system, it is necessary to recognize not only geometric and semantic information of individual objects but also relations between objects in an online manner. Our framework rapidly constructs a 3D scene graph by adopting step-by-step scene abstraction. As shown in Fig. 2, our framework has a four-step structure: (1) dense 3D reconstruction and semantic information recognition for a scene, (2) abstraction of scene objects as oriented bounding boxes, (3) construction of a 3D scene graph representing the scene context, and (4) arranging virtual content in a variety of indoor scenes using 3D scene graphs.

At the first step of the framework, we generate a 3D semantic map (i.e., a 3D panoptic map with additional affordance information) for the scene geometry and semantic information representation. We used PanopticFusion [40] as a basis, with modifications for integrating multiple types of image recognition results from RGB-D data into the 3D map (Fig. 2(1)). The 3D semantic map has voxel-wise resolution and stores object recognition labels and high-level semantic information, like affordances for efficient processing of later steps and a natural interaction representation.

Next, we extract a 3D scene graph as a scene context from the 3D semantic map. To this end, considering computational efficiency, we abstract the individual objects from a voxel representation to oriented bounding boxes (OBBs) and estimate the orientations of OBBs from geometric heuristics and affordances (Fig. 2(2)). Then, by estimating experimentally defined spatial relations between each object, the system achieves connectivity of the graph (Fig. 2(3)).

The final step includes 3D scene graph comparison between the context-represented graph of the AR content and that of the real environment for an AR experience and obtains a context-aware content arrangement from the results (Fig. 2(4)). Our framework processes the four steps described above in an online fashion according to the RGB-D data input and incrementally updates the 3D semantic map and its corresponding 3D scene graph.

The proposed online framework is capable of recognizing individual objects in a real scene, their categories, and high-level semantic information and scene context as spatial relations, which will lead to the realization of rich AR representations in varied environments. As far as we know, our method achieves AR content arrangements

in various real environments based on the deep scene understanding by image-based object recognition and 3D scene graph generation, for the first time.

The contributions of this paper are as follows:

- We introduce a novel AR framework for context-aware virtual content arrangement in a variety of real environments. Our approach abstracts a scene as a graph representation and places contents in a context-aware way via a graph matching approach, without the need for a pre-defined environment.
- We introduce an online 3D scene graph construction method using a 3D semantic map based on image-based object recognition. Our method also includes an affordance detection module which makes object orientation estimation easier and allows virtual objects to interact with real environments more naturally.
- We experimentally demonstrate that our framework is capable of arranging content in a context-aware way to a variety of interior scenes which have different furniture layouts.

## 2 RELATED WORK

**3D Scene Understanding via Semantic Mapping** Semantic mapping reconstructs the 3D structure of a scene and recognizes its semantics simultaneously, which provides crucial clues for seamless connection between the virtual and physical spaces in an AR experience. With the great successes achieved by deep convolutional networks in recent years, some works [39, 46] have proposed achieving semantic 3D scene reconstruction by combining 2D semantic segmentation and Visual SLAM, and this approach has realized category-level 3D object recognition. Then, the simultaneous recognition of semantic categories and instances of 3D scene objects has been achieved by using, for example, an object-oriented map representation [38, 43, 50], and a combination [19] of image-based instance segmentation, a famous example of which is Mask R-CNN [21], and Visual SLAM.

The most related semantic mapping system, PanopticFusion [40] performed a panoptic segmentation task [28] (i.e., recognizing scene objects at the level of *stuff* and *things* from RGB images) in 3D scenes and reconstructed 3D panoptic maps for the first time. This approach enables holistic scene understanding by integrating recognition results into 3D volumes. This work also demonstrates that geometric information and semantic information at the level of stuff and things of a real environment are useful in AR applications with virtual-real interactions, like a situation in which a virtual character locomotes on the floor and sits on chairs and sofas.

From the perspective of AR, Chen *et al.* [9] have presented a context-aware AR framework with the idea of representing scene context based on dense 3D reconstruction and semantic understanding of scene materials. In contrast, we recognize not only the attributes of individual objects but also the relations between them as the scene context and achieve rich AR expressions with a deeper understanding of the holistic scene context.

**AR/VR Experiences in Unknown Environments** There are previous works to adapt and generate AR/VR experiences in unknown, different environments in real-time [10, 18, 20, 41, 52]. The closest work of our framework is FLARE [18]. FLARE generates an optimal layout for AR applications based on the planar geometry of scenes and given constraints. While FLARE focuses on generating simple layouts for AR contents based on geometric information, since our framework utilizes both geometric and semantic information, making it suitable for dealing with a more complex AR scenario/storytelling that is aware of the semantics of individual objects and their relationships. On the other hand, real-time VR systems like DreamWalker [52], VRoamer [10] provide continuous, immersive experiences in uncontrolled physical environments. These works

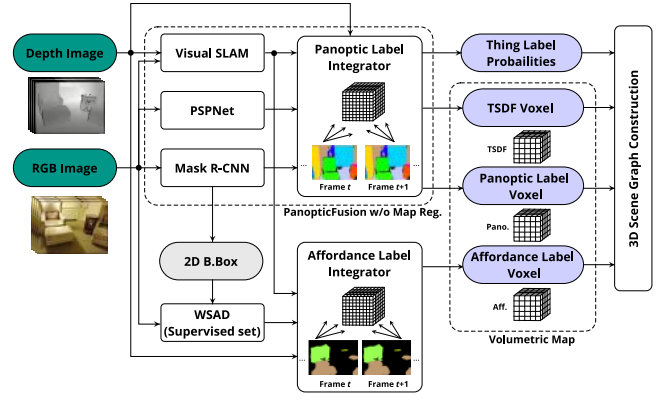


Figure 3: Overview of semantic map generation module.

are closely related to ours in terms of providing consistency-aware experiences across virtual-real worlds in real-time.

**Scene Representation as Graph** Abstract representations of scene contexts as graph or hierarchical structures based on semantic relations between objects or structures have been utilized in the computer vision and graphics communities due to their compactness and high applicability. In graphics, abstract representations of scene contexts as relation graph have often been used in the hierarchical analysis of indoor 3D scene models [17], indoor scene synthesis based on various inputs like texts [8], examples [16], and generative models [49].

In computer vision, the large-scale annotated dataset Visual Genome [29] has been introduced in recent years, and has spurred research on scene graph generation tasks [33, 51] which enable deep understanding of real scenes from a single RGB image. Scene graphs are suitable for various vision tasks, such as image retrieval [24], image generation [23], visual question answering [54], and image captioning [53], and scene graph generation is a promising research topic that is attracting more and more attention in these communities. The closest works to our approach are recent 3D scene graph generation methods for indoor environments using RGB-D sensing data [1, 27, 58].

In our framework, in terms of scalability and flexibility, we represent a scene context as a 3D scene graph, similarly to these methods. The graph is constructed in an online manner based on a 3D semantic map reconstructed from incoming RGB-D images of the real environment. To the best of our knowledge, this is the first AR framework based on a 3D scene graph generated online.

## 3 PROPOSED FRAMEWORK

The proposed framework takes in streams of RGB-D images from off-the-shelf scanners and generates a 3D semantic map from the images (Fig. 2). With a 3D scene graph constructed from the map and a scene graph characterizing the context of AR content defined by content designers, the framework outputs context-aware AR content arrangements according to a real environment. Diverse types of semantics (e.g., affordance, material, texture) could be considered as additional semantic information here. In this paper, we select affordance, which is closely related to interactions in real environments, as an additional semantic. Each module will be described in details in the following subsections.

### 3.1 Semantic Map Generation

#### 3.1.1 Multi-label Integration Module

As shown in Fig. 3, we map a Truncated Signed Distance Function (TSDF) representing the scene geometry and object recognition results into 3D voxel spaces based on camera poses estimated from RGB-D data and depth values, the same as PanopticFusion [40].



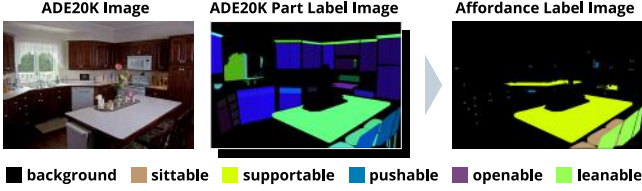


Figure 4: Indoor scene affordance examples converted from ADE20K parts labels.

TSDF enables implicit volumetric representation of the scene by storing the truncated signed distance to the object surface [12], and allows easily extracting meshes of objects, which are used for the next object abstraction step, using Marching Cube algorithm [36].

In the PanopticFusion pipeline, each voxel contains truncated signed distances to the object surface, and panoptic labels, representing stuff labels and things labels uniformly, are integrated into it together with truncated signed weights. By contrast, to perform efficient addition and integration of higher-level semantic information, we adopt a structure for preserving TSDF, panoptic labels, and additional semantic labels from other recognition results in independent voxel spaces. This data structure allows for efficient multi-label integration. While TSDF voxels are updated in the whole scene, including spaces that do not contain objects, through raycasting, it is sufficient for the voxels storing semantic labels to be updated only around the object surface, which reduces the update processing cost and the number of voxels for hashing and enables us to integrate geometry and semantics separately. Besides, this approach easily provides 3D semantic meshes by obtaining the correspondences of voxel spaces preserving each semantic label and mesh generated from the TSDF. To simplify the implementation, we omit the map regularization introduced in the original method, which is based on a Conditional Random Field (CRF).

### 3.1.2 Affordance Detection Module

Recognizing affordances (i.e. interactions that real objects offer virtual contents) will contribute to natural interactions. In recent years, some learning-based methods that detect object affordances from RGB images via neural networks [15, 37, 44] have been proposed. However, most of the open datasets with annotated affordance labels are mainly built for supporting robot manipulation in different scenes and are not matched with our purpose, like assisting interactions with virtual content in indoor scenes such as the living rooms of users. Therefore, inspired by the method [37] of Luddecke *et al.*, we built an ADE20K Affordance dataset by converting the part labels of ADE20K [56] into affordance labels, assuming the indoor scene interactions shown in Fig. 4. We use the fully supervised setting of Weakly Supervised Affordance Detection (WSAD) [44] as an affordance detection head trained on the created dataset. The affordance detector is applied to RGB images in the input RGB-D streams and outputs pixel-wise affordance segmentation of these images. Note that the segmentation process is applied not on the whole image but the cropped regions of interest using the bounding boxes estimated for individual objects by Mask R-CNN.

## 3.2 Scene Object Abstraction

In general, dense reconstruction and labeling of scenes are desirable for virtual content locomotion or interaction in the scenes. However, with respect to comparing the real environments with the context assumed by the virtual content designers, it is not suitable to deal with the scene representation as dense voxels due to the processing complexity and computational cost involved. Also, to ensure the online performance of the framework, it is necessary to represent the scene context information in a more tractable way.

Here, taking the 3D scene graph construction process at the later step into account, we abstract individual object poses and positions

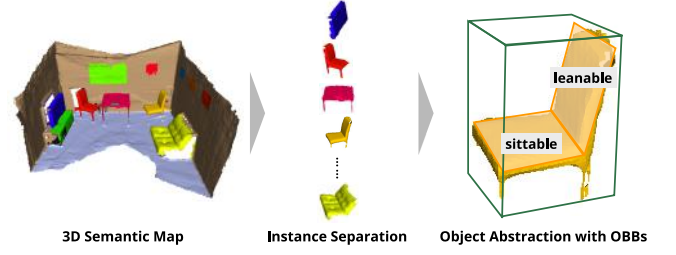


Figure 5: Abstracting scene objects with Oriented Bounding Boxes (OBBs) per instance. Affordance parts of objects will be abstracted as well, if recognized.

as a lightweight 3D representation on the basis of the obtained map. In particular, separating the 3D structures per object, we estimate instances and parts level Oriented Bounding Boxes (OBBs). Since the semantic map has recognition labels for individual objects, as Fig. 5 shows, we can easily separate the 3D structure of each recognized instance from the map. Then we estimate OBBs for separated instances to abstract their positions, orientations, and sizes. The positions and sizes of OBBs are determined as minimum bounding boxes using center points and points extracted from meshes of separated instances. The orientations are estimated using geometric heuristics described later. The OBBs for the regions with affordance are also estimated and stored as parts separately with the instance ones. Our framework assumes that most of the objects in general indoor environments are standing upright, so that OBB instances are estimated as vertical axis-aligned (z-axis aligned) ones. An OBB corresponding to a part of an object does not employ this assumption.

We use several geometric heuristics to predict the orientations of the OBBs. First, we select the front and back axis of an OBB from its extension for each category. For instance, most of the furniture objects in an indoor scene, which we mainly focus on, are longer in the left/right direction than the front/back direction. In such a situation, it is appropriate to set an axis corresponding to the front and back to be perpendicular to the direction of the longer extension. In some categories, such as beds, we set it otherwise. Next, we utilize the observation that most of the recognized objects have more measured surfaces on the front side than the back. We determine the orientation of an OBB as the normal orientation of its surface that is closest to the major direction of normals for recognized objects.

For some categories, our framework is able to utilize affordance information to estimate the front side of an OBB more accurately. Specifically, the orientations of some objects that have *sittable* and *leanable* areas, such as chairs and sofas, are robustly and quickly estimated from the positions of the OBBs of those affordance labels. The OBB for *sittable* is usually located in front of the *leanable* one, and the direction to the *sittable* area, i.e. the seat area, from the *leanable* area, i.e. the backrest area, is always the same as the front orientation. We calculate the direction vector corresponding to the front orientation using the positions of *sittable* and *leanable* OBBs and set the orientation of an instance OBB as the normal orientation of its surface that is closest to the direction vector.

## 3.3 3D Scene Graph Construction

### 3.3.1 Graph Definition

To represent scene context, we introduce a 3D directed graph  $G = (V, E)$  that encodes recognized objects as nodes  $v_i \in V$ , and one or more relations  $r(v_i, v_j) \in R$  between nodes as edges  $e_{ij} \in E$ . The node  $v_i$  corresponds to an instance OBB described in Sect. 3.2 and holds diverse information concerning each object as attributes  $a_{v_i} \in A$ . The attributes could include diverse information, such as color, size, and material, in addition to affordance, which our framework employs, depending on the assumed context for the AR content.

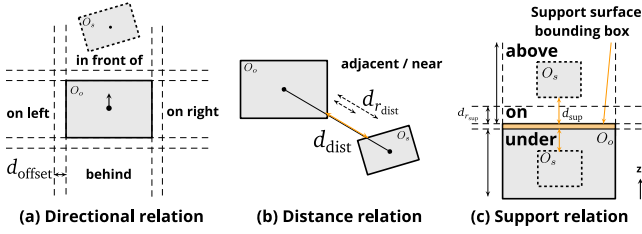


Figure 6: Spatial relations. The relations are defined with a pair of object bounding boxes.

### 3.3.2 Spatial Relations

The proposed framework utilizes the graph representation defined in Sect. 3.3.1 and extracts relations between nodes  $r(v_i, v_j) \in R$  as spatial relations between abstracted objects in real environments. We adopt a geometric rule-based approach to extract spatial relations using the compactly parameterized shape and pose of OBBs. These relations allow to efficiently construct a 3D scene graph. In particular, we extract spatial relations  $r(v_s, v_o)$  between the subject  $v_s$  and the object  $v_o$  as directed-relations  $r(O_s, O_o)$  derived from their OBBs  $O_s, O_o$ .

Taking the online performance into account, we define the following fundamental spatial relations  $r(O_s, O_o)$  which just need relatively lightweight processing based on the ones proposed by Wang *et al.* [49] and Li *et al.* [32].

**Directional Relation ( $r_{\text{dir}}$ ).** Denotes the direction in which  $O_s$  is located relative to  $O_o$ . As shown in Fig. 6(a), when the center point of  $O_s$  is placed in regions between lines defined by extensions of  $O_o$  in the horizontal plane (XY-plane), we represent relations according to the orientation of  $O_o$  as  $\{r_{\text{dir}}(O_s, O_o) | r_{\text{dir}} = \text{in front of, behind, on left, on right}\}$ .  $d_{\text{offset}}$  here means an offset parameter for extending  $O_o$  and enlarging valid areas.

**Distance Relation ( $r_{\text{dist}}$ ).** Denotes the distance between  $O_s$  and  $O_o$ . As shown in Fig. 6(b), when the distance between the surfaces of  $O_s$  and  $O_o$  on the line connecting the centers of  $O_s$  and  $O_o$  in the horizontal plane (XY-plane) is shorter than a defined threshold  $d_{r_{\text{dist}}}$ , we represent the relations as  $\{r_{\text{dist}}(O_s, O_o) | r_{\text{dist}} = \text{near, adjacent}\}$ . Note that distance relations are bidirectional; that is,  $r_{\text{dist}}(O_s, O_o)$  and  $r_{\text{dist}}(O_o, O_s)$  always works at the same time.

**Support Relation ( $r_{\text{sup}}$ ).** Denotes spatial relations on the vertical axis (z-axis) between  $O_s$  and  $O_o$ . As shown in Fig. 6(c), the relations are defined by the distance between the support surface bounding box of  $O_o$  and top or bottom planes of  $O_s$  with threshold  $d_{r_{\text{sup}}}$  and represented as  $\{r_{\text{sup}}(O_s, O_o) | r_{\text{sup}} = \text{on, above, under}\}$ . Here, the center of  $O_s$  is included in  $O_o$  in the horizontal plane. The support surface bounding box is specified per  $O_o$  to extract the support relations of each object pair. When the object corresponding to  $O_o$  has *sittable* or *supportable* areas, we use their OBBs as the support surface bounding boxes; otherwise we utilize  $O_o$  itself.

All relations  $r(O_s, O_o) \in R$  defined above are investigated for each pair of OBBs ( $O_i, O_j$ ). If none of the relations are detected, the object pair does not have edges between them in the 3D scene graph.

### 3.4 Scene Graph-based Content Arrangement

**Content Description.** As described above, our framework represents the context of AR content as a scene graph, called an AR scene graph. Therefore, to achieve natural behaviors of AR contents in real scenes in which they are semantically registered to their AR scenarios, it is required for content designers to define in advance what situations of scenes, or contexts, are needed for the scenarios.

AR scene graphs are also configured as  $G' = (V', E')$  following the format defined in Sect. 3.3.1. In  $G'$ , a virtual content is also represented as a node  $v'_c \in V'$  and is connected with other nodes via relations  $r_{\text{act}} \in R$  denoting their interactions. Suppose the AR context is defined as "A virtual character is sitting on a chair in front of which is a TV", as shown in Fig. 1. First of all, nodes for required components, "TV", "chair" and "virtual character" are added as  $v'_0, v'_1, v'_2 \in V'$ . Then we associate TV  $v'_0$  and chair  $v'_1$  with *in front of*( $v'_0, v'_1$ ) as "A TV is in front of a chair" and connect an edge  $e'_{01} \in E'$ . Finally, the virtual character  $v'_c$  and chair  $v'_1$  are associated with  $r_{\text{act}} = \text{sitting on}(v'_c, v'_1)$  and  $e'_{c1}$  is created in  $G'$  for representing the situation that "virtual character" is "sitting on" a "chair". Our framework compares an AR scene graph  $G'$  defined in this manner with a real environment graph  $G$  constructed in Sect. 3.3.1 and outputs content arrangements regarding the results.

**Content Arrangement.** A 3D scene graph  $G$ , representing a real AR environment in an abstract way, is incrementally updated according to the 3D semantic map update. Whenever  $G$  is updated, the content arrangement module compares the structure of  $G$  with AR scene graph  $G'$ . For the comparison, we use graph  $G''$  derived from  $G'$ , which a node  $v'_c$  corresponding to a virtual content, and its connected edges with relation  $r_{\text{act}}$  are removed. When an obtained graph  $G$  for an AR scene includes  $G''$ , that is, the context-representing graph  $G''$  becomes a subgraph of the scene graph  $G$  extracted from the real environment, it enables us to identify objects for AR in the 3D semantic map from the graph correspondences. Thus we can obtain the content placement in a context-aware way that satisfies the assumed context shown in  $G'$ .

## 4 IMPLEMENTATION OF A PROTOTYPE SYSTEM

To verify the effectiveness of our framework, we constructed prototype system (Fig. 7) that consists of two PCs: (1) A PC for semantic map generation and 3D scene graph construction with the input RGB-D streams; (2) A PC for AR content arrangement with the semantic map and input 3D scene graph. RGB-D sequences are captured by off-the-shelf RGB-D sensors, in our case ASUS Xtion Pro Live, and camera poses are known or are estimated by Visual SLAM.

Here, we clarify the details of the settings for the learning in each recognition module included in the semantic map generation module. We used PSPNet [55] and Mask R-CNN [21] as head CNNs for 2D semantic and instance segmentation, the same as in the original PanopticFusion [40]. PSPNet was fine-tuned with ScanNet v2 [13] using pre-trained weights for ADE20K [56]. Mask R-CNN used different recognition categories for quantitative evaluation of the semantic mapping module and subjective evaluation by participants. For quantitative evaluation, it was fine-tuned with ScanNet v2 [13] using pre-trained weights for COCO [34]. In contrast, for subjective evaluation, we added *tv* and *remote* as additional categories to ScanNet's 20 categories, for the purpose of enhancing the AR experience. To this end, we established and utilized the ScanNet20+2 dataset by extracting 4547 images including TVs and remote controls from COCO [34] and merged them with ScanNet's 17161 images. The 80 categories of COCO were converted into ScanNet20+2 categories following Table 1. We used the same pre-trained weights and parameters for training described in the original paper [40].

As described above, we used the fully supervised settings of WSAD [44] as the affordance detection module and trained it on the ADE20K Affordance dataset we built by converting part labels of ADE20K [56] into affordance labels for indoor scenes. The ADE20K Affordance dataset consists of 3388 images (2980 training images, 408 test images). We extracted indoor scene images including one or more regions converted to the defined affordance labels according to Table 2 and cropped them to  $321 \times 321$  pixels according to an open implementation [44]. The parameters for training

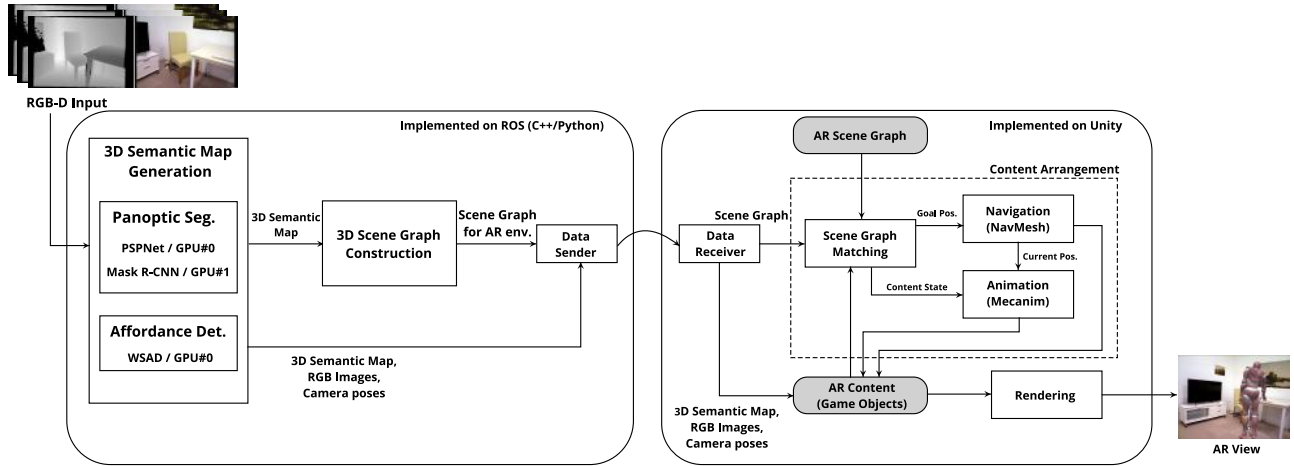


Figure 7: Configuration of prototype system.

Table 1: Conversion table of ScanNet20+2 from COCO80.

ScanNet20+2	COCO80
bed	bed
chair	chair
sofa	coach
table	dining table
refrigerator	refrigerator
toilet	toilet
sink	sink
tv	tv
otherfurniture	bench
remote	remote
background, otherprops	other categories not shown on the table

Table 2: Conversion table of ADE20K parts to affordances

Affordance	ADE20K Parts
sittable	seat, seat_cushion
supportable	bed/base, bed/bedspring, shelf, countertop, work_surface
pushable	button(s), switch, light_switch(es), electric(al).switch, push_button(s)
openable	knob, handle
leanable	chair/back, chair/back_pillow, sofa/back, sofa/back_pillow

were the same as the open implementation [44]. In the evaluations, we performed the whole process of the semantic map generation and 3D scene graph generation with a PC equipped with an Intel Core i7-7800X CPU running at 3.6GHz and two NVIDIA GeForce GTX1080Ti GPUs.

A generated semantic map and a 3D scene graph, together with RGB images and camera poses for AR, were sent in the form ROS messages to the PC for AR content arrangement via our modified rosbridge-based transmission module. After receiving the data from the PC for semantic map generation and 3D scene graph construction, the PC for AR content arrangement compared the AR scene graph with the 3D scene graph to control the AR contents. In particular, we used built-in functions in Unity to animate the AR character and to navigate the character on the *floor* area of the semantic map. The PC for AR content arrangement was equipped with an Intel Core i7-7800X CPU running at 3.5GHz and an NVIDIA GeForce GTX1080 Ti GPU. This configuration example is the same as the one used in the qualitative evaluations and subjective evaluation described later.

## 5 EXPERIMENTS

### 5.1 Evaluation of Semantic Mapping

To evaluate the semantic mapping performance of our prototype system quantitatively, we performed experiments on both the 3D panoptic segmentation task and the affordance detection task with the system.

For the 3D panoptic segmentation task, we employed 3D panoptic quality (PQ) in the ScanNet dataset as a metric and compared our system with PanopticFusion [40] without CRF (Table 3). PQ is

Table 3: 3D panoptic segmentation results on ScanNet (v2) open test set.

method	metric	all	things	stuff
PanopticFusion w/o CRF [40]	PQ	29.7	26.7	56.7
	SQ	71.2	71.4	69.5
	RQ	41.1	36.8	79.6
PanopticFusion w/ multi-label integration (our prototype system)	PQ	30.5	27.5	56.9
	SQ	71.3	71.4	70.1
	RQ	42.2	38.1	79.2

Table 4: Pixel accuracy and mean IoU on ADE20K Affordance dataset.

overall	pixel accuracy (%)						mIoU(%)
	avg	sittable	supportable	pushable	openable	leanable	
64.0	44.4	67.0	62.0	7.0	17.0	69.0	44.2

defined as the multiplication of recognition quality (RQ) and segmentation quality (SQ) [28]. Table 3 shows that there was little difference in the PQ/RQ/SQ scores between PanopticFusion and our system. Therefore, this shows that the proposed framework can integrate an additional recognition module such as an affordance detector without sacrificing its object recognition performance.

In the affordance detection task, we evaluated the affordance segmentation results with pixel accuracy and mean Intersection-over-Union (mIoU) on test images of the ADE20K Affordance dataset introduced in Sect. 4. The pixel accuracy for the overall image was 64.0 %, whereas the average score of classes turned out to be 44.4 %. These results reflected the fact that the regions with *pushable* or *openable* labels, corresponding to parts such as buttons and knobs, were so small (Fig. 4) that they were more challenging to detect than others. Other labels, *sittable*, *leanable* and *supportable* obtained good estimation results with accuracies over 60 %. Fig. 8 shows examples of detection results for affordance masks.

### 5.2 Evaluation of Scene Object Abstraction

We quantitatively evaluated the detection performance of OBBs using the Scan2CAD dataset [2] built for CAD model retrieval and alignment tasks. The Scan2CAD dataset contains information associating similar 3D CAD models of ShapeNet [7] with real scene 3D scans of ScanNet via 3D keypoint correspondences and also includes annotations for 3D OBBs of CAD models aligned with the scans.

Setting the annotations of 3D OBBs as the ground truth, we evaluated median orientation errors of the predicted OBBs and mIoU/precision/recall for True Positive detection results. In our evaluation process, panoptic and affordance labels were estimated online and mapped on the ground truth geometries of ScanNet, which yielded 3D semantic maps for 3D scene graph construction. To associate categories defined in Scan2CAD with those of Scan-



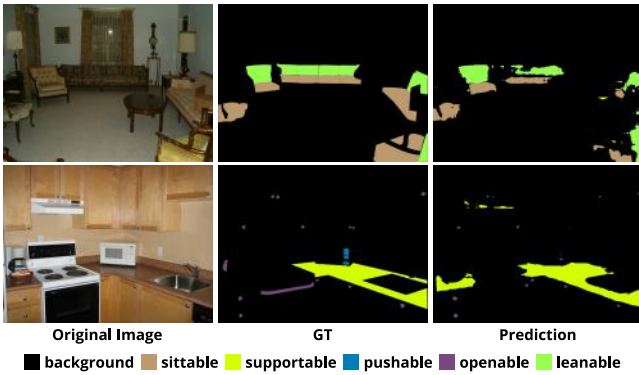


Figure 8: Examples of affordance detection results.

Table 5: Median orientation error in Scan2CAD dataset.

	IoU th.	bath	bkshf	cab	chair	tv	sofa	tabl
Median orientation error [deg.]	0.25	2.16	2.24	2.10	4.01	3.10	1.82	1.55
	0.50	1.21	1.94	1.25	2.74	1.87	1.78	1.31

Net20+2 described in Sect. 4, we treated the *display* category of Scan2CAD as the *TV* category of ScanNet20+2, and excluded *trash-bin/other*, which do not have suitable counterpart categories, from the evaluation targets. In addition, we used z-axis aligned poses for the 3D OBB ground truth.

Table 5 shows median orientation errors of successfully detected OBBs. They were in the same category as the ground truth and had IoUs of 0.25/0.5 or more. The results confirmed that the proposed system could estimate the orientations of the OBBs with errors of several degrees for the majority of OBBs.

We defined True Positive OBBs as successfully detected OBBs with orientation errors of 20 degrees or less and calculated the TP IoU/precision/recall, as shown in Table 6. The results show that *sofa* was recognized particularly well. It suggests that the orientation estimation by the affordance and the large object size improved its detection performance.

Although the orientations of *chair* were similarly estimated by using the affordance, it can be inferred that due to the lack of shape information in some parts of the chair (Fig. 9 bottom-right), which resulted from incomplete scanning, its IoU decreased, resulting in a lower Recall rate. Both *sofa* and *chair* in particular showed better Precision rates, which means that their orientations were predicted more accurately by using affordances rather than other categories. Fig. 9 shows examples of detected OBBs in the ScanNet scenes.

### 5.3 Subjective Evaluation of Context-Aware Content Arrangement

To evaluate whether or not the proposed framework was able to obtain context-aware content arrangements using 3D scene graphs, we performed a subjective evaluation of AR content arrangements in various scenes.

First, we set an AR scene graph  $G'$ . Then, from RGB-D data and camera poses of the ScanNet dataset, we extracted a 3D semantic map and a 3D scene graph  $G$  by using the prototype system. For a scene satisfying the AR scene graph  $G'$ , we placed AR content on 3D scans based on the comparison result of  $G$  and  $G'$ , achieving a context-aware arrangement (CA). Apart from that arrangement procedure, we performed random arrangement (RA) of the AR content so that the categories of objects with which the virtual content interacted matched with those of  $G'$ . We presented the arrangement results in the same scene from the two approaches above for the AR scene graph  $G'$  to participants. The participants gave subjective evaluations for the set context-awareness of the arrangement result by using a five-point Likert scale: from "strongly unexpressed" to "strongly expressed". As participants, we selected 12 people who did not have much knowledge of the proposed framework. Note

Table 6: TP IoU, precision, and recall evaluation in Scan2CAD dataset.

	IoU th.	bath	bkshf	cab	chair	tv	sofa	tabl	avg
TP IoU(%)	0.25	53.97	50.88	58.43	51.46	46.45	66.59	55.38	54.74
	0.50	63.73	61.36	70.06	63.38	56.74	68.44	65.93	64.23
Precision(%)	0.25	50.00	44.07	15.49	68.14	39.46	72.00	32.39	45.94
	0.50	28.95	23.73	10.20	31.59	15.68	66.67	20.45	28.18
Recall(%)	0.25	31.67	36.79	30.38	45.20	38.22	47.79	28.93	37.00
	0.50	18.33	19.81	20.00	20.95	15.18	44.25	18.26	22.40



Figure 9: Examples of predicted OBBs with the ground truths. Blue and red boxes are ground truth and predicted OBBs, respectively. OBBs without explicit orientation are colored yellow. Color gradient (white to blue/red) means the forward direction of the OBBs. (top: scene0025\_02, middle: scene0144\_00, bottom: scene0414\_00)

that the participants were not informed of the ways of arranging contents for the two results, and the order of the presented contents was randomized.

In our experiments, we set the following two types of scene graphs as  $G'$  (Fig. 10):

**Context A** "A character is sitting on a chair in front of which is a TV." (Fig. 10 top)

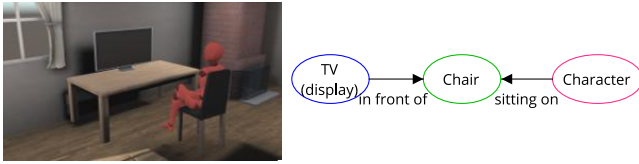
**Context B** "A lamp is on a table to the left of a sofa." (Fig. 10 bottom)

To clarify contexts provided by  $G'$ , we presented the above text descriptions that describe a scene graph in natural language to participants, together with  $G'$ .

There were a total of 33 scenes in the ScanNet dataset satisfying the contexts above: 27 for Context A and 6 for Context B. For these scenes, we performed both CA by the prototype system and RA using the categories of interaction objects. The arrangement results were presented to the participants for evaluation.

Fig. 11 shows 3D scans obtained by the prototype system and example results for CA and RA on those scans. The votes for the degrees of expression against each context and the whole result by CA/RA are shown in Fig. 12. In the following, we describe the evaluation results with the notations "Strongly unexpressed" as 1 point, and "Strongly expressed" as 5 points.

For the results by CA, the rates of answers that the participants evaluated as context-expressed was 79% for Context A, 78% for B, and 79% for the whole. In contrast, the results by RA were



**Context A: "A character is sitting on a chair in front of which is a TV."**



**Context B: "A lamp is on a table to the left of a sofa."**

Figure 10: Contexts for subjective evaluation. Scene examples (left) and the scene graphs (right) representing the contexts were provided to subjects.

39% for Context A, 7% for B, and 33% for the whole. In addition, we performed the Wilcoxon Rank Sum Test for the CA/RA results and confirmed that both the results for each scene and the total showed statistically significant differences (p-values were less than 0.01). These findings verified that the CA, the context-aware content arrangement using the scene graph obtained by the proposed framework, was effective for expressing the contexts of AR contents in real environments.

Focusing on the evaluation of each scene, the CA achieved statistically significant evaluation (p-values were less than 0.05) in 19 scenes out of the total 33 and showed better degrees of expression than RA. In many of such scenes, as shown in lines 1, 2, 4, and 5 in Fig. 11, CA can express the scene contexts (Fig. 10) more appropriately than RA. However, there was no significant difference in some scenes where the AR scene graph could be included by selecting any object to be interacted with in the scenes, as shown in line 3 in Fig. 11.

#### 5.4 Online Context-Aware AR Demonstration

We conducted experiments to demonstrate that our prototype system was capable of achieving an AR application in various indoor spaces depending on defined contexts. In these demonstrations, we set the two contexts described in Fig. 13 in advance and verified whether each AR context could be expressed in three different real scenes. Here, by applying Context 2 after Context 1 is expressed in the scenes, we expressed a series of interactions of a virtual character: a character "pushes a remote control on a table" to turn on the TV and then "sits on a chair in front of which is a TV" to watch the TV. As real environments, we employed an open space scene, a meeting room scene, and a living room scene with a TV, a table, a remote control and a chair. Note that our prototype system works without using absolute coordinates, and since it arranges AR contents based on the recognized 3D semantic map and scene graph, the AR program for the experiment did not need any setting change at all in each scene.

Fig. 14 shows the online AR demonstrations in each scene (see the supplementary video for the details of each scene). These demonstrations confirmed that, in each of the scenes, the proposed framework could recognize individual objects and their affordances, and spatial relations between objects in an online fashion. In addition, they showed that the framework could obtain context-aware content arrangements based on the recognition results. Fig. 15 illustrates the scene graph recognized in the open space scene and its relationship between Context 1 and Context 2. Table 7 shows the runtime analysis for each component of the proposed framework in executing the AR application in the open space scene.

With this experiment, we demonstrated that the proposed method is capable of achieving AR expressions in various physical environments, according to the set contexts, and expressing a series of

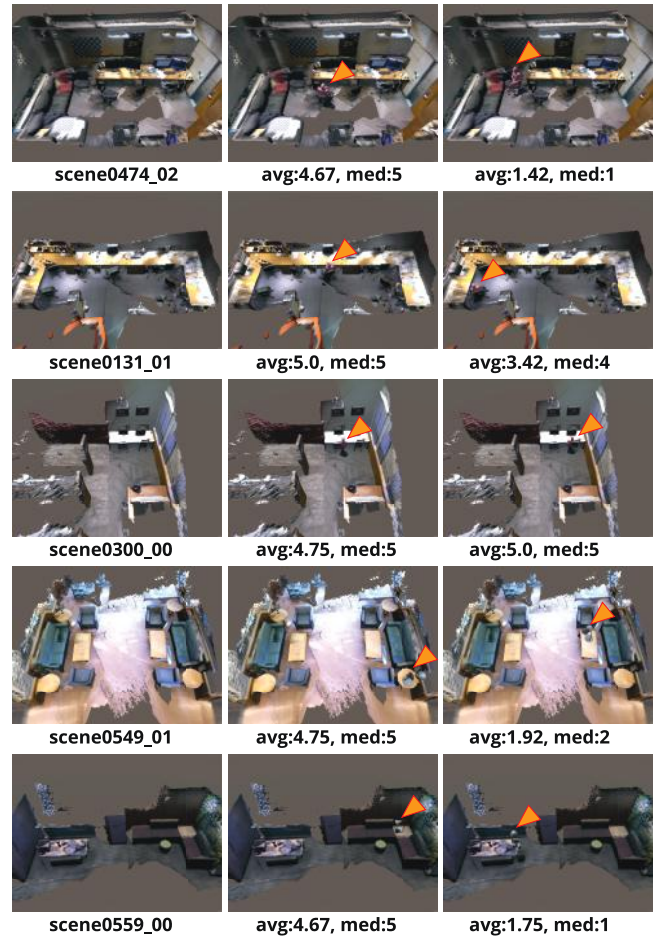


Figure 11: 3D scans (left column), context-aware content arrangement (middle column), and random content arrangement (right column) for subjective evaluation. Average and median scores were estimated from subjects' responses for each scene. Orange triangles indicate locations of arranged contents.

actions by switching the contexts. We consider that the proposed framework will enable more storytelling, space-adaptive AR applications that go beyond conventional AR applications, like virtual contents appearing in the space randomly.

#### 6 CONCLUSIONS

In this work, we have presented Retargetable AR, a novel online AR framework that produces AR experiences facilitating natural interaction between virtual and real worlds according to the pre-set context of AR contents. To express the scene context, the Retargetable AR generates a 3D semantic map in which geometric and semantic information densely reconstructed and labeled from RGB-D data scanned in the scene is integrated, and constructs a 3D scene graph based on spatial relations between objects in an online manner. Then it compares the context-representing graph with the AR scene graph corresponding to the context of the AR content as the 3D scene graph is updated and obtains the correspondences between graph nodes, achieving context-aware content arrangements in various scenes. The performance and effectiveness of the Retargetable AR were extensively verified through quantitative evaluations of the semantic map and its predicted 3D OBBs, subjective evaluation of the content arrangement results, and online context-aware AR demonstrations in real environments. To the best of our knowledge, the proposed approach is the first that considers the semantic registration between the virtual world and the physical world via dense semantic scene understanding, providing an AR experience with natural and realistic



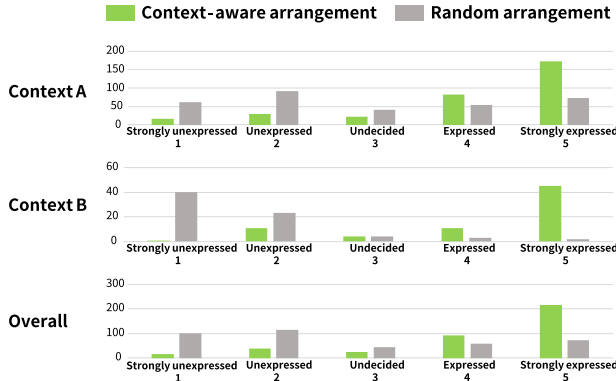


Figure 12: Evaluation scores from subjects for each context and overall, in context-aware and random arrangements.

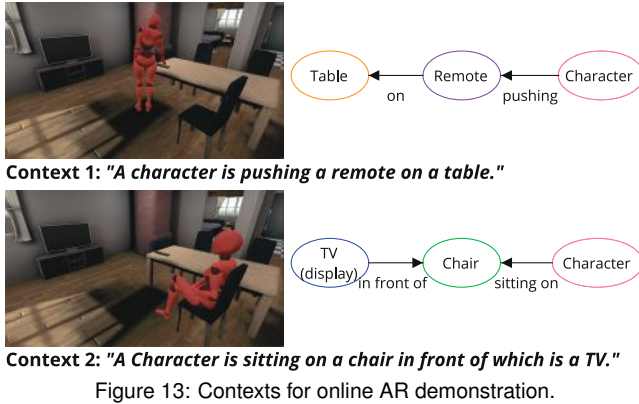


Figure 13: Contexts for online AR demonstration.

interactions.

Finally, we discuss the limitations and open challenges that the proposed framework did not address or include. In our work, though we present context-aware AR in static scenes, it is desirable to support AR in dynamic scenes and to use the framework in actual environments. We believe that updating the semantic map while keeping its online performance through the detection of moving objects and relocalization, or object tracking, etc. may be important.

The Retargetable AR employed geometric heuristics to estimate the orientations of objects. However, it is too difficult to accurately estimate them by using heuristics because the objects have a wide variety of geometries, even those of the same category. We showed examples in which the combination of semantic information, such as affordance, and the geometric information enabled accurate orientation prediction (e.g., *chair*, *sofa*). As another approach to address this problem, we believe that further improvement of a category-level 6DoF object pose estimation method [48] using deep neural networks could contribute to solving this problem. Moreover, it is necessary to explore reasonable ways of disentangling the ambiguity of the orientation of symmetric objects and their directional relationships. (E.g., what is a suitable way to define or estimate the orientations of L-shaped sofas and round tables, and what kind of directional relations should be used for expressing their contexts?)

How to create an AR application with an AR scene graph in a more efficient way or how to scale the graph definition to express various kinds of contexts are also major issues to be addressed. In this paper, we recognized affordance as an additional attribute of an object, extracted nine kinds of spatial relations, and then defined AR scene graphs by hand based on them to create AR applications. However, the introduced spatial relations, of course, have limited ability to express contexts, and thus more research is needed to explore more suitable relations, attributes and representations for reflecting the intention of the creators in AR applications more

Table 7: Run-time analysis of our prototype system running in the open space.

Frequency	Component	time
Every RGB-D frames	TSDF integration	103 ms
Every Mask R-CNN frames	PSPNet	124 ms
	Mask R-CNN	266 ms
	Affordance detection	492 ms
	Panoptic label integration	58 ms
	Affordance label integration	56 ms
	Panoptic label tracking	6 ms
Every 1 sec.	Probability integration	~1 ms
	Mesh extraction	14 ms
	Scene object abstraction	173 ms
	Scene graph construction	~2 ms
Triggered timing	Content arrangement	~1 ms

accurately and variously. In addition, the development of authoring tools that will enable us to easily and efficiently generate or develop AR applications that use such attributes or relations for representing contexts is desirable in the future.

## REFERENCES

- [1] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3D scene graph: A structure for unified semantics, 3D space, and camera. In *Proc. ICCV*, pp. 5663–5672. IEEE, 2019. doi: 10.1109/ICCV.2019.00576
- [2] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proc. CVPR*, 2019.
- [3] B. Boom, S. Orts-Escolano, X. Ning, S. McDonagh, P. Sandilands, and R. Fisher. Interactive light source position estimation for augmented reality with an rgb-d camera. *Computer Animation and Virtual Worlds*, 28(1), 1 2017. doi: 10.1002/cav.1686
- [4] E. Brachmann and C. Rother. Learning less is more - 6d camera localization via 3D surface regression. In *Proc. CVPR*, pp. 4654–4662, 2018.
- [5] S. Brahmabhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *Proc. CVPR*, 2018.
- [6] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc. IEEE International Symposium on Wearable Computers*, pp. 15–22, 2008.
- [7] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3D model repository. *CoRR*, abs/1512.03012, 2015.
- [8] A. X. Chang, M. Savva, and C. D. Manning. Learning spatial knowledge for text to 3D scene generation. In *Proc. EMNLP*, pp. 2028–2038. Association for Computational Linguistics, Doha, Qatar, 2014. doi: 10.3115/v1/D14-1217
- [9] L. Chen, W. Tang, N. John, T. R. Wan, and J. J. Zhang. Context-aware mixed reality: A framework for ubiquitous interaction. *arXiv preprint arXiv:1803.05541*, 2018.
- [10] L.-P. Cheng, E. Ofek, C. Holz, and A. D. Wilson. Vroamer: Generating on-the-fly vr experiences while walking inside large, unknown real-world building environments. In *Proc. VR*, pp. 359–366. IEEE, 2019.
- [11] M. Corp. and contributors. MixedRealityToolkit-Unity, 2016. <https://github.com/Microsoft/MixedRealityToolkit-Unity/>.
- [12] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, pp. 303–312, 1996.
- [13] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. CVPR*, 2017.
- [14] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration. *ACM Transactions on Graphics 2017 (TOG)*, 2017.
- [15] T.-T. Do, A. Nguyen, and I. Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *Proc. ICRA*, pp. 1–5. IEEE, 2018.

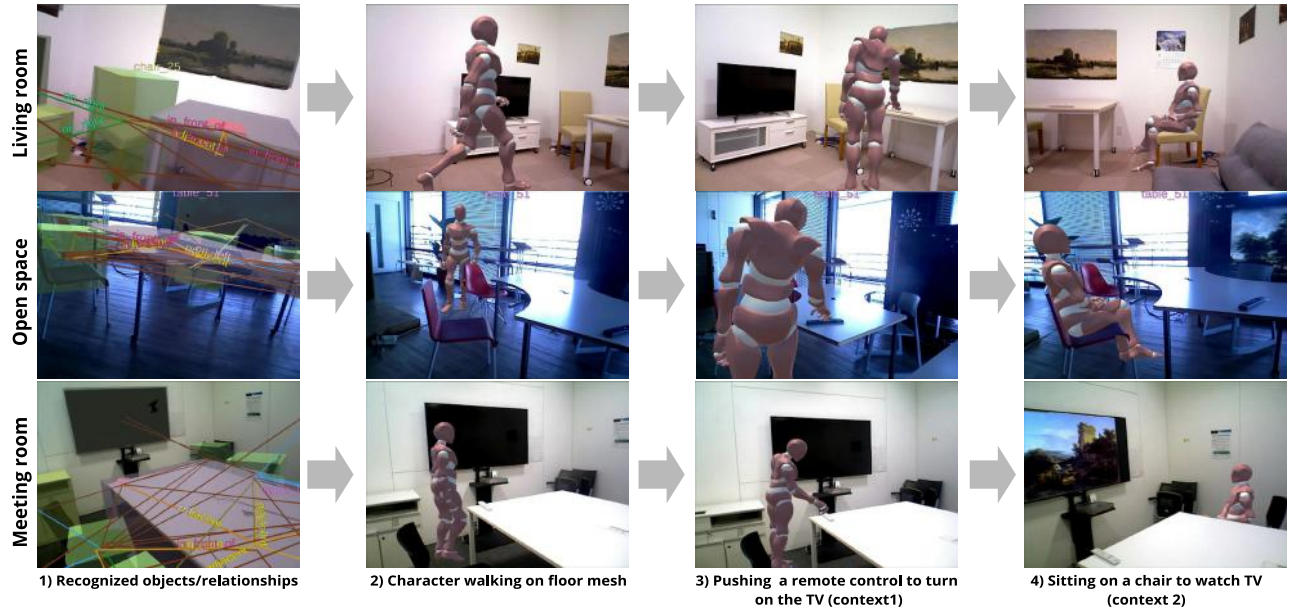


Figure 14: Online context-aware AR demo in various scenes.

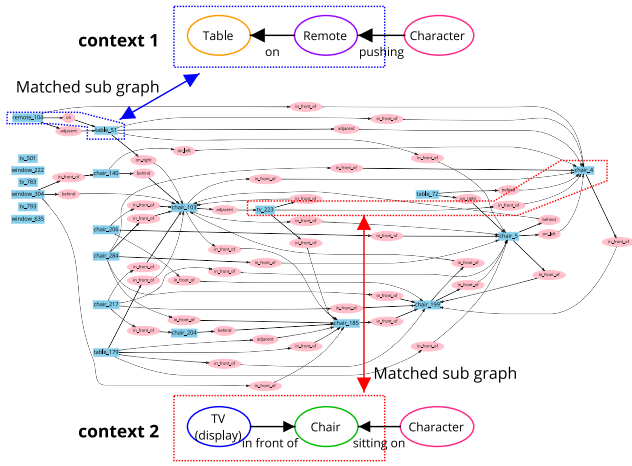


Figure 15: Matched sub graphs between generated scene graphs and AR scene graphs in the open space of Fig. 14. Note that "near" relationships are not visualized.

- [16] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3D object arrangements. *ACM TOG*, 31(6):1–11, 2012.
- [17] M. Fisher, M. Savva, and P. Hanrahan. Characterizing structural relationships in scenes using graph kernels. In *ACM TOG*, vol. 30, p. 34. ACM, 2011.
- [18] R. Gal, L. Shapira, E. Ofek, and P. Kohli. Flare: Fast layout for augmented reality applications. In *Proc. ISMAR*, pp. 207–212. IEEE, 2014.
- [19] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto. Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, July 2019. doi: 10.1109/LRA.2019.2923960
- [20] J. Hartmann, C. Holz, E. Ofek, and A. D. Wilson. Realitycheck: Blending virtual environments with situated physical reality. In *Proc. CHI*, pp. 1–12, 2019.
- [21] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proc. ICCV*, pp. 2961–2969, 2017.
- [22] A. Inc. ARKit, 2017. <https://developer.apple.com/augmented-reality/>.
- [23] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proc. CVPR*, pp. 1219–1228, 2018. doi: 10.1109/CVPR.2018.00133
- [24] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proc. CVPR*, pp. 3668–3678, 2015. doi: 10.1109/CVPR.2015.7298990
- [25] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. IEEE and ACM International Workshop on Augmented Reality (IWAR)*, pp. 85–94, 1999. doi: 10.1109/IWAR.1999.803809
- [26] K. Kim, M. Billinghurst, G. Bruder, H. B. Duh, and G. F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2947–2962, Nov 2018. doi: 10.1109/TVCG.2018.2868591
- [27] U. Kim, J. Park, T. Song, and J. Kim. 3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Transactions on Cybernetics*, pp. 1–13, Aug. 2019. doi: 10.1109/TCYB.2019.2931042
- [28] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proc. CVPR*, pp. 9404–9413, 2019.
- [29] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. doi: 10.1007/s11263-016-0981-7
- [30] Y. Lang, W. Liang, and L. Yu. Virtual agent positioning driven by scene semantics in mixed reality. In *Proc. VR*, pp. 767–775, 2019. doi: 10.1109/VR.2019.8798018
- [31] V. Lepetit. Instant outdoor localization and slam initialization from 2.5d maps. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–10, 2015.
- [32] A. Li, J. Sun, J. Yue-Hei Ng, R. Yu, V. I. Morariu, and L. S. Davis. Generating holistic 3D scene abstractions for text-based image retrieval. In *Proc. CVPR*, pp. 193–201, 2017.
- [33] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *Proc. ICCV*, pp. 1270–1279, 2017. doi: 10.1109/ICCV.2017.142
- [34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in

- context. In *Proc. ECCV*, pp. 740–755. Springer, 2014.
- [35] G. LLC. ARCore, 2017. <https://developers.google.com/ar/>.
  - [36] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.
  - [37] T. Luddecke and F. Worgotter. Learning to segment affordances. In *Proc. ICCVW*, pp. 769–776, 2017.
  - [38] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric object-level slam. In *Proc. 3DV*, pp. 32–41. IEEE, 2018.
  - [39] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3D semantic mapping with convolutional neural networks. In *Proc. ICRA*, pp. 4628–4635. IEEE, 2017.
  - [40] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *Proc. IROS*, pp. 4205–4212, 2019. doi: 10.1109/IROS40897.2019.8967890
  - [41] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson. Snaptoreality: Aligning augmented reality to the real world. In *Proc. CHI*, pp. 1233–1244, 2016.
  - [42] K. Rohmer, J. Jendersie, and T. Grosch. Natural environment illumination: Coherent interactive augmented reality for mobile and non-mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2474–2484, Nov 2017. doi: 10.1109/TVCG.2017.2734426
  - [43] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *Proc. ISMAR*, pp. 10–20. IEEE, 2018. doi: 10.1109/ISMAR.2018.00024
  - [44] J. Sawatzky, A. Srikantha, and J. Gall. Weakly supervised affordance detection. In *Proc. CVPR*, pp. 2795–2804, 2017.
  - [45] T. M. Schöps, T. Sattler, and M. Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proc. CVPR*. IEEE, 2019.
  - [46] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proc. CVPR*, pp. 6243–6252, 2017.
  - [47] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *Proc. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 125–134, 2008. doi: 10.1109/ISMAR.2008.4637338
  - [48] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proc. CVPR*, pp. 2642–2651, 2019.
  - [49] K. Wang, Y.-A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM TOG*, 38(4):1–15, 2019.
  - [50] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *Proc. ICRA*, pp. 5231–5237. IEEE, 2019.
  - [51] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Proc. CVPR*, pp. 3097–3106, 2017. doi: 10.1109/CVPR.2017.330
  - [52] J. Yang, C. Holz, E. Ofek, and A. D. Wilson. Dreamwalker: Substituting real-world walking experiences with a virtual reality. In *Proc. ACM UIST*, pp. 1093–1107, 2019.
  - [53] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *Proc. ECCV*, pp. 684–699, 2018.
  - [54] C. Zhang, W. Chao, and D. Xuan. An empirical study on leveraging scene graphs for visual question answering. In *Proc. BMVC*, 2019.
  - [55] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proc. CVPR*, pp. 2881–2890, 2017.
  - [56] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proc. CVPR*, 2017.
  - [57] F. Zhou, H. B.-L. Duh, and M. Billinghamurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proc. ISMAR*, ISMAR ’08, p. 193–202. IEEE Computer Society, USA, 2008. doi: 10.1109/ISMAR.2008.4637362
  - [58] Y. Zhou, Z. While, and E. Kalogerakis. Scenegraphnet: Neural message passing for 3D indoor scene augmentation. In *Proc. ICCV*. IEEE, 2019. doi: 10.1109/ICCV.2019.00748