2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Reconstructing Interactive 3D Scenes by Panoptic Mapping and CAD Model Alignments

Muzhi Han* Zeyu Zhang* Ziyuan Jiao Xu Xie Yixin Zhu Song-Chun Zhu Hangxin Liu

***Abstract*— In this paper, we rethink the problem of scene reconstruction from an embodied agent's perspective: While the classic view focuses on the reconstruction accuracy, our new perspective emphasizes the underlying functions and constraints such that the reconstructed scenes provide *actionable* information for simulating *interactions* with agents. Here, we address this challenging problem by reconstructing an interactive scene using RGB-D data stream, which captures (i) the semantics and geometry of objects and layouts by a 3D volumetric panoptic mapping module, and (ii) object affordance and contextual relations by reasoning over physical common sense among objects, organized by a graph-based scene representation. Crucially, this reconstructed scene replaces the object meshes in the dense panoptic map with part-based articulated CAD models for finer-grained robot interactions. In the experiments, we demonstrate that (i) our panoptic mapping module outperforms previous state-of-the-art methods, (ii) a high-performant physical reasoning procedure that matches, aligns, and replaces objects' meshes with best-fitted CAD models, and (iii) reconstructed scenes are physically plausible and naturally afford actionable interactions; without any manual labeling, they are seamlessly imported to ROS-based simulators and virtual environments for complex robot task executions.**[1]

I. Introduction

Perception of the human-made scenes and the objects within inevitably leads to the course of actions [1, 2]; such a task-oriented view [3, 4] is the basis for a robot to interact with the environment and accomplish complex tasks. In stark contrast, such a crucial perspective is largely missing in the robot mapping and scene reconstruction literature: Prevailing semantic mapping or Simultaneous Localization and Mapping (SLAM) methods often produce a metric map of the scene with semantic or instance annotations; they only emphasize mapping accuracy but omit the essence of robot task execution—actions that a semantic entity could afford and associated physical constraints embedded among entities.

Such a lack of the scene's functional representation leads to a gap between the reconstructed semantic scenes and Task and Motion Planning (TAMP), which prevents a robot from directly interacting with the reconstructed scenes to accomplish complex tasks. Take the reconstructed scene in Fig. 1 as the example, wherein the robot is tasked to pick up a frozen meal from the fridge, microwave and serve it. To properly plan and execute inside the reconstructed scene, a robot ought to acquire (i) semantics and geometry of objects (*e.g.*, this piece of point cloud is a fridge), (ii) actions an object affords (*e.g.*, a fridge can be open), and (iii) constraints among these entities (*e.g.*, no objects should float in the air). Although modern semantic mapping and SLAM methods can partially address (i) [5, 6], existing solutions for (ii) [4, 7, 8] and (iii) [9–14] have not yet been fully integrated into a robot scene reconstruction framework, resulting in non-interactive reconstructed scenes. This deficiency precludes the feasibility of directly applying TAMP on the reconstructed scenes either using traditional [15, 16] or learning-based [17, 18] methods; the robot can hardly verify whether its plan is valid or the potential outcomes of its actions are satisfied before executing in the physical world.

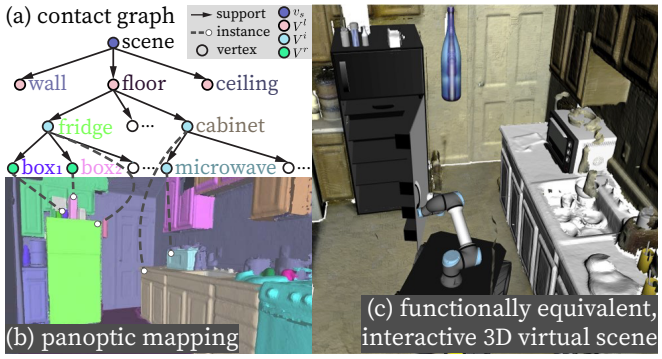


Fig. 1: **The reconstruction of an interactive 3D scene.** (a) A contact graph is constructed by the supporting relations that emerged from (b) panoptic mapping. By reasoning their affordance, functional objects within the scene are matched and aligned with part-based interactive CAD models. (c) The reconstructed scene enables a robot simulates its task execution with comparable outcomes in the physical world.

Although researchers have attempted to devise manual pipelines (*e.g.*, iGibson [19], SAPIEN [20]) to either convert the reconstructed real-world scenes or directly build virtual environments from scratch, creating such simulation environments is a non-trivial and time-consuming task. The simulated environment should be sufficiently similar to the reality, and the objects to be interacted with should afford sufficiently similar functionality. Only by satisfying the above conditions could the outcomes of interactions in simulation be similar to those in the physical world. Due to the enormous workload to create/convert each scene, the number of available scenes to date is still quite limited. A challenge naturally arises: Can we reconstruct a scene that can be automatically imported into various simulators for interactions and task executions?

In this paper, we propose a new task of reconstructing *functionally equivalent* and interactive scenes, capable of being directly imported into simulators for robot training

* Muzhi Han and Zeyu Zhang contributed equally to this work.

UCLA Center for Vision, Cognition, Learning, and Autonomy (VCLA) at the Statistics Department. Emails: {muzhihan, zeyuzhang, zyjiao, xiexu, yixin.zhu, hx.liu}@ucla.edu, sczhu@stat.ucla.edu.

The work reported herein was supported by ONR N00014-19-1-2153, ONR MURI N00014-16-1-2007, and DARPA XAI N66001-17-2-4029.

[1]The code is available at `https://github.com/hmz-15/Interactive-Scene-Reconstruction`.

and testing of complex task execution. We argue that a scene's functionality is composed of the functions afforded by objects within the scene. Therefore, the essence of our scene reconstruction lies in defining functionally equivalent objects, which should preserve four characteristics with decreasing importance: (i) its semantic class and spatial relations with nearby objects, (ii) its affordance, *e.g.*, what interactions it offers, (iii) a similar geometry in terms of size and shape, and (iv) a similar appearance.

Existing approaches oftentimes represent reconstructed semantic scene and its entities as sparse landmarks [21, 22], surfels [5, 23], or volumetric voxels [24, 25]. However, these representations are inadequate to serve as a *functional representation* of the scene and its entities: They merely provide occupancy information (*i.e.*, where the fridge is) without any actionable information for robot interactions or planning (*e.g.*, whether or how the fridge can be open).

To address the above issues, we devise three primary components in our system; see an illustration in Fig. 2:

(A) A robust 3D volumetric panoptic mapping module, detailed in Section III, accurately segments and reconstructs 3D objects and layouts in clustered scenes even with noisy per-frame image segmentation results. The term "panoptic," introduced in [26], refers to jointly segmenting *stuff* and *things*. In this paper, we regard objects as *things* and layout as *stuff*. Our system produces a volumetric panoptic map using a novel per-frame panoptic fusion and a customized data fusion procedure; see examples in Fig. 1b and Fig. 2a.

(B) A physical common sense reasoning module, detailed in Section IV, replaces object meshes obtained from the panoptic map with interactive rigid or articulated CAD models. This step is achieved by a ranking-based CAD matching and an optimization-based CAD alignment, which accounts for both geometric and physical constraints. We further introduce a global physical violation check to ensure that every CAD replacement is physically plausible.

(C) A graphical representation, *contact graph* cg, (Fig. 1a, Fig. 2c, and Section II) is built and maintained simultaneously, in which the nodes of a cg represent objects and layouts, and the edges of a cg denote the support and proximal relations. We further develop an interface to convert a cg to a Unified Robot Description Format (URDF) such that the reconstructed functionally equivalent scene (see Fig. 1C) can be directly imported into simulators for robot interactions and task executions; see Section V for experimental results.

Related Work: Existing approaches to generate simulated interactive environments fall into three categories: (i) **manual efforts**, such as those in Gazebo [27] and V-REP [28] for robotics, AI2THOR [29] and Gibson [30] for embodied AI, and iGibson [19], SAPIEN [20], and VR-Gym [31] with part-based articulated objects (*e.g.*, a cabinet with a door); (ii) **scene synthesis** that produces a massive amounts scenes with the help of CAD databases [32–34]; (iii) **large-scale scene dataset** with aligned CAD models, such as SUNCG [35] and 3D-FRONT [36]. However, without tedious manual work, all of these prior approaches fail to replicate a real scene in simulation with diverse interactions.

Modern **semantic mapping** [6, 24, 37] and **object SLAM** [22, 25] methods can effectively reconstruct an indoor scene at an object-level. Physical cues, such as support and collision, has been further integrated to estimate and refine the object pose [38–40]. In parallel, computer vision algorithms predict **3D instance segmentation** in densely reconstructed scenes [41, 42], and then fit CAD models by crowdsourcing [43] or by computing the correspondences between the reconstructed scenes and CAD models [44, 45]. However, the above work fails to go beyond semantics to (i) capture the interactive nature of the objects, or (ii) meaningfully represent a physically plausible scene. As such, the reconstructed scenes still fail to be imported into simulators to afford robot interactions and task executions.

Constructing a proper scene or a map **representation** remains an open problem [46]. Typical semantic mapping and SLAM methods only output a flat representation, difficult to store or process high-level semantics for robot interactions and task executions. Meanwhile, graph-based representations, *e.g.*, scene grammar [11, 13, 14, 34, 47, 48] and 3D scene graph [49–51], provide structural and contextual information. In particular, Rosinol *et al.* [51] also incorporate actionable information for robot navigation tasks. Our work devises a *contact graph* with supporting and proximal relations, which imposes kinematic constraints for more complex robot manipulation.

II. Contact-based Scene Representation

We devise a graph-based representation, *contact graph* cg, to represent a 3D indoor scene. Formally, a contact graph $cg = (pt, E)$ contains (i) a parse tree (*pt*) that captures the hierarchical relations among the scene entities [47], and (ii) the proximal relations E among entities represented by undirected edges; see an example of *pt* in Fig. 1a.

A. Representation

Scene Parse Tree $pt = (V, S)$ has been used to represent the hierarchical decompositional relations (*i.e.*, the edge set S) among entities (*i.e.*, the node set V) in various task domains, including 2D images and 3D scenes [11, 13, 14, 33, 34, 48, 52], videos and activities [4, 8, 53], robot manipulations [54–58], and theory of mind [59]. In this paper, we adopt *pt* to represent supporting relations among entities, dynamically built and maintained during the reconstruction; for instance in Fig. 1a, the `cabinet` is the parent node of the `microwave`. Supporting relation is quintessential in scene understanding with physical common sense as it reflects the omnipresent physical plausibility; *i.e.*, if the cabinet were moved, the microwave would move together with it or fall onto the ground. This counterfactual perspective goes beyond occupancy information (*i.e.*, the physical location of an object); in effect, it further provides actionable information and the potential outcome of actions for robot interactions and task executions in the scene.

Scene Entity Nodes $V = \{v_s\} \cup V^L \cup V^R \cup V^A$ include: (i) the scene node v_s, severing as the root of pt, (ii) layout node set V^L, including floor, ceiling, and the wall that bound the 3D scene, (iii) rigid object set V^R, wherein each object has no articulated part (*e.g.*, a table), and (iv) articulated

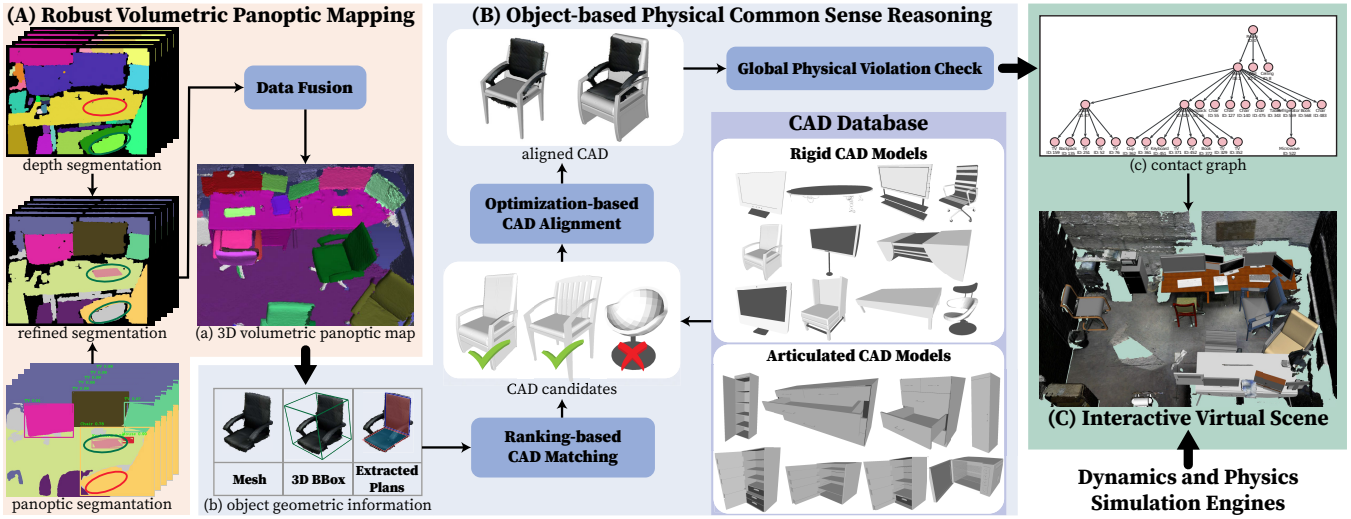


Fig. 2: **System architecture for reconstructing a functionally equivalent scene.** (A) Per-frame segmentation and cross-frame data fusion produce (a) a 3D volumetric panoptic map with fine-grained semantics and geometry, served as the input for (B) physical common sense reasoning that matches, aligns, and replaces segmented object meshes with functionally equivalent CAD alternatives. Specifically, (b) by geometric similarity, a ranking-based matching algorithm selects a shortlist of CAD candidates, followed by an optimization-based process that finds a proper transformation and scaling between the CAD candidates and object mesh. A global physical violation check is further applied to finalize CAD replacements to ensure physical plausibility. (C) This CAD augmented scene can be seamlessly imported to existing simulators; (c) contact graph encodes the kinematic relations among the objects in the scene as the planning space for a robot.

object set V^A, wherein each object has articulated parts to be interacted for various robot tasks (*e.g.*, fridge, microwave). Each non-root node $v_i = \langle o_i, c_i, M_i, B_i(\boldsymbol{p}_i, \boldsymbol{q}_i, \boldsymbol{s}_i), \Pi_i \rangle$ encodes a unique instance label o_i, a semantic label c_i, a full geometry model M_i (a triangular mesh or a CAD model), a 3D bounding box B_i (parameterized by its position $\boldsymbol{p}_i$, orientation $\boldsymbol{q}_i$, and size $\boldsymbol{s}_i$, all in $\mathbb{R}^3$), and a set of surface planes $\Pi_i = \{\boldsymbol{\pi}_i^k, k = 1 \cdots |\Pi_i|\}$, where $\boldsymbol{\pi}_i^k$ is a homogeneous vector $[\boldsymbol{n}_i^{k\,T}, d_i^k]^T \in \mathbb{R}^4$ in the projective space [60] with unit plane normal vector $\boldsymbol{n}_i^k$, and any point $\boldsymbol{v} \in \mathbb{R}^3$ on the plane satisfies a constraint: $\boldsymbol{n}_i^{k\,T} \cdot \boldsymbol{v} + d_i^k = 0$.

Supporting Relations S is the set of directed edges in pt from parent nodes to their child nodes. Each edge $s_{p,c} \in S$ imposes physical common sense between the parent node v_p and the child node v_c. These constraints are necessary to ensure that v_p supports v_c in a physically plausible fashion: **(i) Geometrical plausibility:** v_p should have a plane $\boldsymbol{\pi}_p^s = [\boldsymbol{n}_p^{s\,T}, d_p^s]^T$ with $\boldsymbol{n}_p^s$ being opposite to the gravity direction, whereas bottom surface of v_c should contact the top of $\boldsymbol{\pi}_p^s$:

$$\exists \boldsymbol{\pi}_p^s \in \Pi_p, \boldsymbol{n}_p^{s\,T} \cdot \boldsymbol{g} \leqslant a_{th},\ s.t.\ \mathcal{D}(v_c, \boldsymbol{\pi}_p^s) = p_c^g - (-d_p^s + s_c^g/2) = 0, \quad (1)$$

where $\boldsymbol{g}$ is the unit vector along the gravity direction, $a_{th} = -0.9$ is a tolerance coefficient, d_p^s is the offset of the v_p's supporting plane, and p_c^g and s_c^g denote the position and size of the v_c's 3D bounding box along the gravity direction. **(ii) Sufficient contact area for stable support:** Formally,

$$\mathcal{A}(v_p, v_c) = \mathrm{A}(v_p \cap v_c)/\mathrm{A}(v_c) \geqslant b_{th}, \quad (2)$$

where $\mathrm{A}(v_c)$ is the bottom surface of the v_c's 3D bounding box, and $\mathrm{A}(v_p \cap v_c)$ is the area of the overlapping rectangle containing the mesh vertices of v_p near $\boldsymbol{\pi}_p^s$ within v_c's 3D bounding box. We set threshold $b_{th} = 0.5$ for a stable support.

Proximal Relations E introduce links among entities in the pt. They impose additional constraints by modeling spatial relations between two non-supporting but physically nearby objects v_1 and v_2: Their meshes should not penetrate with each other, *i.e.*, $\mathrm{Vol}(M_1 \cap M_2) = 0$. Note that the constraint only exists between two objects with overlapping 3D bounding boxes, *i.e.*, when $\mathrm{Vol}(B_1 \cap B_2) > 0$.

B. Constructing Contact Graph

Each node v_x in cg is constructed from a scene entity x in the panoptic map (see Section III) by: (i) acquiring its $o_x, c_x, M_x, B_x(\boldsymbol{p}_x, \boldsymbol{q}_x, \boldsymbol{s}_x)$, (ii) extracting surface planes Π_x by iteratively applying RANSAC [61] and removing plane inliers, and (iii) assigning x as v_x in cg.

Given a set of nodes constructed on-the-fly, we apply a bottom-up process to build up cg by detecting supporting relations among the entities. Specifically, given an entity v_c, we consider all entities $\{v_i\}$ whose 3D bounding boxes are spatially below it and have proper supporting planes $\boldsymbol{\pi}_i^k$ based on Eq. (1). The most likely supporting relation is chosen by maximizing the following score function:

$$S(v_c, v_i, \boldsymbol{\pi}_i^k) = \left\{1 - \min\left[1, \|\mathcal{D}(v_c, \boldsymbol{\pi}_i^k)\|\right]\right\} \times \mathcal{A}(v_i, v_c), \quad (3)$$

where the first term indicates the alignment between the v_c's bottom surface and the v_i's supporting planes, and the second term reflects an effective supporting area, both normalized to $[0, 1]$. B_i is further refined (see Eq. (1)) as it was computed based on incomplete object meshes. Meanwhile, the proximal relations are assembled by objects' pairwise comparison. At length, the cg of the scene is constructed based on the identified entities and their relations and grows on-the-fly.

III. Robust Panoptic Mapping

Robust and accurate mapping of scene entities within clustered environments is essential for constructing a cg and serving downstream tasks. Below, we describe our robust panoptic mapping module to generate volumetric object and layout segments in the form of meshes from RGB-D streams; see the pipeline in Fig. 2A. We follow the framework

proposed in [24] and **only highlight crucial technical modifications below**. The experiments demonstrate that our modifications significantly improve system performance.

Per-frame Segmentation: We combine the segmentation of both RGB and depth for performance improvement as in [24]. However, instead of merely labeling the depth segments with semantic-instance masks, we bilaterally fuse panoptic masks and geometric segments to output point cloud segments with both semantic and instance labels. We further perform an outlier removal for each object entity; far away segments are removed and assigned to the scene background.

This modification significantly improves the noisy per-frame segmentation; see Fig. 2a. In this example, fusing RGB and depth segments mutually improves the segments if they were obtained by each alone. The fusion (i) correctly segments the keyboard and divides the two monitors when depth segments fail, and (ii) geometrically refines the noisy panoptic mask of the chair to exclude the far-away ground.

Data Fusion: Compared to [24], we introduce two notable enhancements in data fusion. First, we use a triplet count $\Phi(l, c, o)$ to record the frequency that an instance label o, a semantic label c, and a geometric label l associated with the same point cloud segment; it is incrementally updated: $\Phi(l, c, o) = \Phi(l, c, o) + 1$. This modification improves consistency in semantic-instance fusion. Second, in addition to merging two geometric labels if they share voxels over a certain ratio, we also regulate two instance labels if the duration of association with a common geometric label exceeds a threshold. We further estimate a gravity-aligned, 3D-oriented bounding box for each object mesh [62]. In sum, our system simultaneously and comprehensively outputs a set of scene entities with their instance labels, semantic labels, 3D bounding boxes, and reconstructed meshes.

Implementation and Evaluation: We use an off-the-shelf panoptic segmentation model [63] pre-trained on the COCO panoptic class [64] for RGB images and a geometric segmentation method [65] for depth images. We compare our panoptic mapping module with the original Voxblox++ [24] on 8 sequences in the SceneNN dataset [66]. Our evaluation includes four criteria: (i) panoptic quality (PQ) [6, 26], (ii) segmentation quality (SQ), (iii) recognition quality (RQ) of 3D panoptic mapping on 8 *thing* classes and 2 *stuff* classes, and (iv) the mean average precision (mAP) computed using an intersection of union (IoU) with a threshold of 0.5 for 3D oriented bounding box estimation on *thing* classes. Since the supporting relations in cg could further refine the 3D bounding boxes (see Section II-B), we also include $\text{mAP}_{\texttt{re}}$.

Table I tabulates the class-averaged results, showing that our method consistently outperforms the baseline in both 3D panoptic mapping and 3D bounding box estimation; see Fig. 5b for some qualitative results. In general, refining objects' 3D bounding boxes with supporting relations introduces significant improvement in accuracy.

TABLE I: **Quantitative results of 3D panoptic mapping and 3D oriented bounding box estimation on 8 sequences in the SceneNN dataset [66].**

	Ours					Voxblox++ [24]			
Sequence ID	PQ	SQ	RQ	mAP	$\text{mAP}_{\texttt{re}}$	PQ	SQ	RQ	mAP
061	**43.0**	52.0	**46.3**	**33.6**	33.6	25.7	**53.1**	32.2	8.9
086	**27.3**	**39.6**	**34.6**	**33.8**	33.8	19.4	32.9	25.2	7.9
096	**12.5**	**21.4**	**14.6**	**14.6**	14.6	7.3	11.0	8.3	14.6
223	**49.5**	**60.2**	**63.3**	24.2	55.6	21.7	40.2	26.7	**61.4**
225	**35.4**	**46.9**	**44.8**	**31.5**	31.5	21.6	43.6	29.4	11.2
231	**37.8**	**45.9**	**45.4**	29.2	**31.3**	17.9	30.4	22.1	19.4
249	**24.4**	33.8	**34.4**	48.9	**71.9**	23.4	**36.4**	30.6	48.5
322	**68.4**	**71.1**	**80.0**	58.3	**83.3**	43.6	64.6	52.9	25.0

IV. Physical Reasoning for CAD Alignments

Due to occlusion or limited camera view, the reconstructed meshes of the scene are oftentimes incomplete. As such, the segmented object meshes are incomplete and non-interactive before recovering them as full 3D models; see examples in Fig. 3a and Fig. 4a. We introduce a multi-stage framework to replace a segmented object mesh with a functionally equivalent CAD model. This framework consists of an object-level, coarse-grained CAD matching and fine-grained CAD alignment, followed by a scene-level, global physical violation check; see an illustration in Fig. 2B.

A. CAD Pre-processing

We collected a CAD database consisting of both rigid and articulated CAD models, organized by semantic classes. The rigid CAD models are obtained from ShapeNetSem [67], whereas articulated parts are first assembled and then properly transformed into one model. Each CAD is transformed to have its origin and axes aligned with its canonical pose. Fig. 2B shows some instances of CAD models in the database. Similar to a segmented object entity, a CAD model y is parameterized by $o_y, c_y, M_y, B_y(\boldsymbol{p}_y, \boldsymbol{q}_y, \boldsymbol{s}_y)$, and Π_y.

B. Ranking-based CAD Matching

Take the chair in Fig. 2b as an example: Given a segmented object entity x, the algorithm retrieves all CAD models in the same semantic category (*i.e.*, chair) from the CAD database to best fit x's geometric information. Since the exact orientation of x is unknown, we uniformly discretize the orientation space into 24 potential orientations. For each rotated CAD model y that aligned to one of the 24 orientations, the algorithm computes a matching distance:

$$D(x, y) = \omega_1 \cdot d_s(x, y) + \omega_2 \cdot d_\pi(x, y) + \omega_3 \cdot d_b(y), \qquad (4)$$

where $\omega_1 = \omega_2 = 1.0$ and $\omega_3 = 0.2$ are the weights of three terms, set empirically. We detail these terms below.

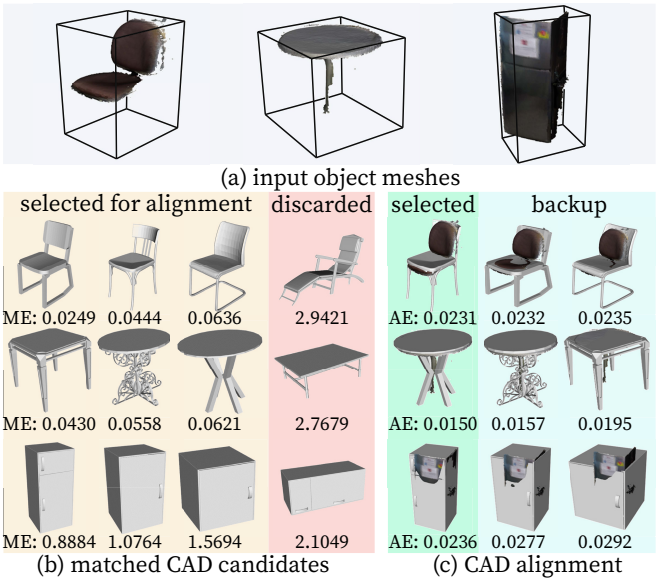
(i) d_s matches the relative sizes of 3D bounding boxes:

$$d_s(x, y) = \left\| \frac{\boldsymbol{s}_x}{\|\boldsymbol{s}_x\|_2} - \frac{\boldsymbol{s}_y}{\|\boldsymbol{s}_y\|_2} \right\|. \qquad (5)$$

(ii) d_π penalizes the misalignment between their surface planes in terms plane normal and relative distance:

$$d_\pi(x, y) = \min_{f_\Pi} \sum_{\boldsymbol{\pi}_i \in \Pi_x} \left[\left\| \frac{d({T_x}^T \boldsymbol{\pi}_i)}{\|\boldsymbol{s}_x\|_2} - \frac{d(f_\Pi(\boldsymbol{\pi}_i))}{\|\boldsymbol{s}_y\|_2} \right\| + 1 - \boldsymbol{n}(\boldsymbol{\pi}_i)^T \cdot \boldsymbol{n}(f_\Pi(\boldsymbol{\pi}_i)) \right], \qquad (6)$$

where T_x denotes the homogeneous transformation matrix from the map frame on the ground to the frame of the bounding box B_x, $d(\cdot)$ and $\boldsymbol{n}(\cdot)$ denote the offset and normal vector of a plane, and $f_\Pi : \Pi_x \to \Pi_y$ is a bijection function denoting the assignment of feature planes between x and y. Note that f_Π is also constrained to preserve supporting planes



(a) input object meshes

(b) matched CAD candidates (c) CAD alignment

Fig. 3: Examples of matching and aligning CAD candidates to (a) an input object mesh. (b) All CAD models within the same semantic class as the input object are retrieved for matching. Matching Error (ME) reflects both the similarity in shapes and the proximity in orientations. After selecting the CAD candidates with smallest MEs, (c) a fine-grained CAD alignment process selects the best CAD model with a proper transformation based on Alignment Error (AE).

as defined in Eq. (1). As computing d_π involves solving an optimal assignment problem, we adopt a variant of the Hungarian algorithm [68] to identify the best f_Π.

(iii) $d_b(y)$ is a bias term that adjusts the overall matching error for less preferable CAD candidates:

$$d_b(y) = 1 + \boldsymbol{g}^T \cdot \boldsymbol{z}(y), \tag{7}$$

where $\boldsymbol{z}(y)$ denotes the up-direction of the CAD model in the oriented CAD frame, and $\boldsymbol{g}$ is a unit vector along the gravity direction. In general, we prefer CAD candidates that stand upright to those leaning aside or upside down.

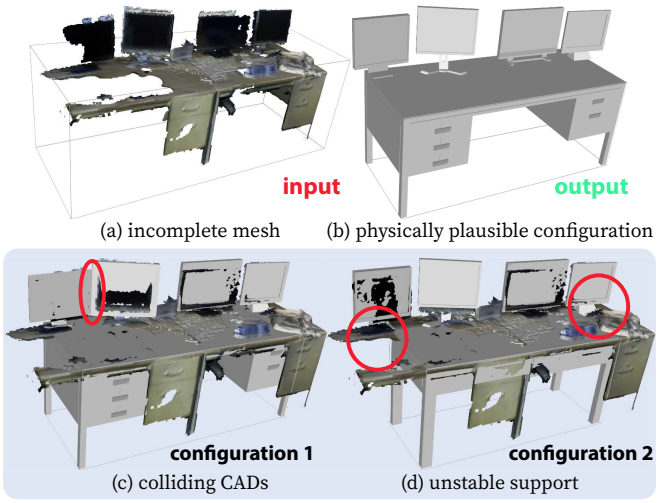
Fig. 3b illustrates the matching process. Empirically, we observe that the discarded CAD candidates of "chair" and "table" due to large Matching Error (ME) are indeed more visually distinct from the input object meshes. Moreover, the "fridge" model with a wrong orientation has a much larger ME and is thus discarded. These results demonstrate that our ranking-based matching process can select visually more similar CAD models with the correct orientation. Our system maintains the top 10 orientated CAD candidates with the lowest ME for the fine-grained alignment in the next stage.

C. Optimization-based CAD Alignment

Given a shortlist of CAD candidates, the overarching goal of this step to find an accurate transformation (instead of 24 discretized orientations) that aligns a given CAD candidate y to the original object entity x, achieved by estimating a homogeneous transformation matrix between x and y:

$$T = \begin{bmatrix} \alpha R & \boldsymbol{p} \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \; s.t. \; \min_T \mathcal{J}(x, T \circ y), \tag{8}$$

where $\circ$ denotes the transformation of a CAD candidate y, $\mathcal{J}$ is an alignment error function, α is a scaling factor,



(a) incomplete mesh (b) physically plausible configuration

(c) colliding CADs (d) unstable support

Fig. 4: Given (a) incomplete object meshes, our physical common sense reasoning for CAD replacement (b) generates a functionally equivalent and physically plausible configuration. Specifically, the CAD matching and alignment algorithms select and rank a shortlist of CAD candidates. A global physical violation check prunes invalid configurations such as (c) collision and (d) unstable support.

$R = Rot(\boldsymbol{z}, \theta)$ is a rotation matrix that only considers the yaw angle under the gravity-aligned assumption, and $\boldsymbol{p}$ is a translation. This translation is subject to the following constraint: $p^g = -d^s + \alpha \cdot s_y^g/2$, as the aligned CAD candidate is supported by a supporting plane $\boldsymbol{\pi} = [\boldsymbol{n}_\cdot^{s\,T}, d_\cdot^s]$.

The objective function $\mathcal{J}$ can be written in a least squares form and minimized by the Levenberg–Marquardt [69] method:

$$\mathcal{J} = \boldsymbol{e}_b^T \Sigma_b \boldsymbol{e}_b + \boldsymbol{e}_p^T \Sigma_p \boldsymbol{e}_p, \tag{9}$$

where $\boldsymbol{e}_b$ is the 3D bounding box error, $\boldsymbol{e}_p$ the plane alignment error, and Σ_b, Σ_p the error covariance matrices of the error terms. Specifically: (i) $\boldsymbol{e}_b$ aligns the height of the two 3D bounding boxes while constraining the ground-aligned rectangle of the transformed B_y inside that of B_x:

$$\boldsymbol{e}_b = [\mathrm{A}(T \circ y)) - \mathrm{A}(x, T \circ y), \alpha \cdot \boldsymbol{s}_y^g - \boldsymbol{s}_x^g]^T, \tag{10}$$

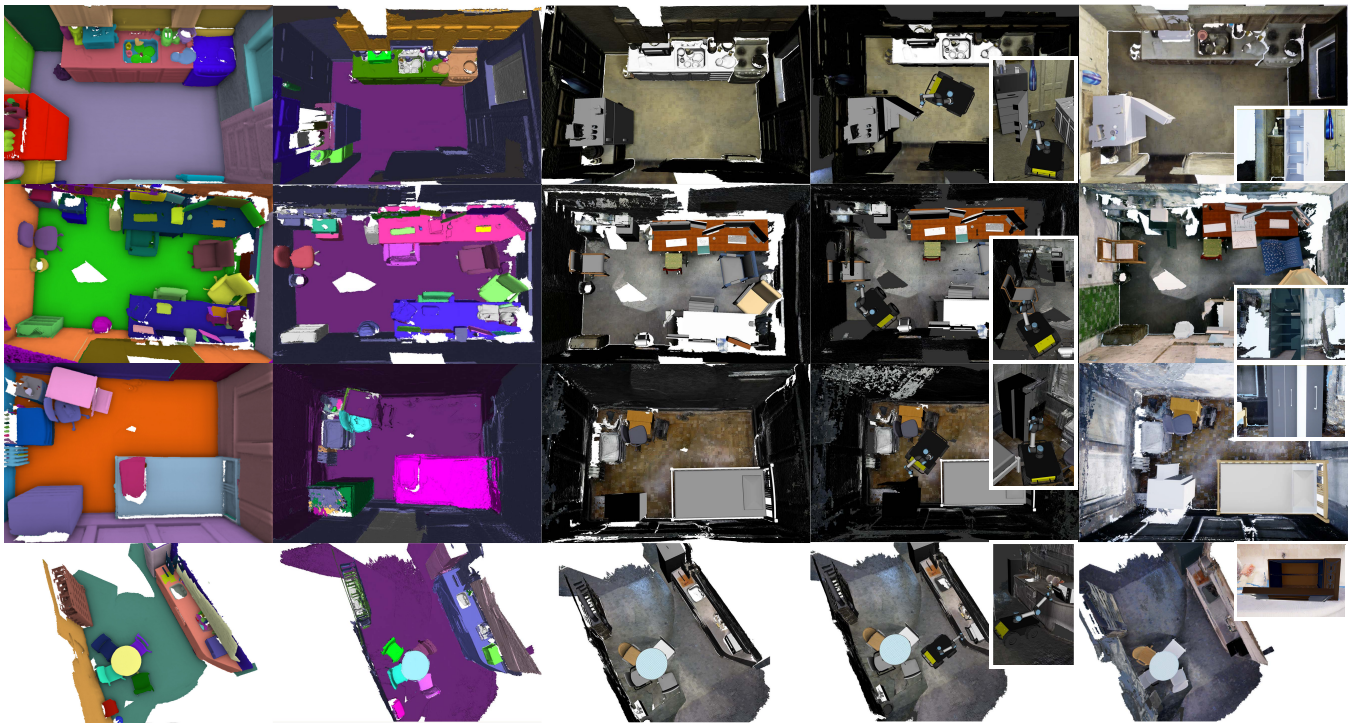
and (ii) $\boldsymbol{e}_p$ aligns all the matched feature planes as:

$$\begin{aligned} &\boldsymbol{e}_p = [\Delta\boldsymbol{\pi}_1, ..., \Delta\boldsymbol{\pi}_{|\Pi_x|}]^T, \\ &\Delta\boldsymbol{\pi}_i = [-d(\boldsymbol{\pi}_i) + d(T^{-T} \cdot f_\Pi(\boldsymbol{\pi}_i)), 1 - \boldsymbol{n}(\boldsymbol{\pi}_i)^T \cdot \boldsymbol{n}(T^{-T} \cdot f_\Pi(\boldsymbol{\pi}_i))]. \end{aligned} \tag{11}$$

We evaluate each aligned CAD candidate by computing an Alignment Error (AE), the root mean square distance between the object mesh vertices and the closest points on aligned CAD candidate; Fig. 3c shows both qualitative and quantitative results. The CAD candidate with the smallest AE will be selected, whereas others are potential substitutions if the selected CADs violate physical constraints, detailed next.

D. Global Physical Violation Check

Given a shortlist of matched and aligned CAD candidates, we validate supporting relations and proximal relations; see Fig. 4 for qualitative results. Specifically, for an object node v_p and its object entity x, we discard a CAD candidate y if it fails to satisfy Eq. (2) with any supporting child v_c of v_p. We also check the proximal constraint by first discarding CAD candidates that collide with the layout



(a) ground-truth segmentation (b) panoptic mapping (c) interactive scenes (d) robot interaction (e) VR interaction

Fig. 5: (a–b) Qualitative comparisons between the ground-truth segmentation [66] and segmentation results produced by the proposed panoptic mapping. (c) The reconstructed functionally equivalent scenes capture most of the objects and replaces them by actionable CAD models. (d–e) Both robots and human users can virtually enter the reconstructed scene for TAMP and VR applications, respectively.

entities, and then jointly selecting CAD candidates for each object entity to guarantee the object-object non-collision. The joint selection problem can be formulated as a constraint satisfaction problem. Starting with a CAD candidate with the minimum alignment error for each object entity, we adopt the min-conflict algorithm [70] to obtain a global solution.

V. Experiments and Results

We perform scene reconstruction experiments using RGB-D sequences in the SceneNN dataset [66] and import the results into various simulators for interaction; see Fig. 5. Compared to the ground-truth segmentation, our panoptic mapping system accurately recognizes and segments scene entities (Fig. 5b). Such an accurate mapping provides the basis for high-level physical reasoning to replace incomplete meshes with CAD models, resulting in a high-quality, functionally equivalent, interactive scene reconstruction, as shown in Fig. 5c. Note that our system's performance could be further improved as we only utilize pre-trained models in the mapping procedure without fine-tuning. The run-time for converting a 3D panoptic map into an interactive scene varies from 30 seconds to several minutes, depending on the number and categories of functional objects involved.

The reconstructed scene cg can be readily converted into a URDF and be imported into robot simulators. While it is straightforward to immigrate scene entities in cg to links and joints in the kinematic tree, supporting edges are altered to fixed/floating joints based on the semantics of the scene entity pairs (*e.g.*, a cup is connected to a table using a floating joint as it can be freely manipulated). Fig. 5c shows the reconstructed scenes in the ROS environment, which subsequently connects the reconstructed scenes and robot TAMP; see Fig. 5d. Fig. 5e demonstrates that the reconstructed scenes can be loaded into the VR environment [31] for interactions with both virtual agents and human users, which opens a new avenue for future studies.

VI. Conclusions

We proposed a new task of reconstructing interactive scenes that captures the semantic and associated actionable information of objects in a scene, instead of purely focusing on geometric reconstruction accuracy. We solved this new task by combining (i) a novel robust panoptic mapping that segments individuals objects and layouts, and (ii) a physical reasoning process to replace incomplete objects meshes with part-based CAD models, resulting in physically plausible and interactive scenes. We validated the capability of our system with both qualitative and quantitative results. Finally, we showed that various simulators (*e.g.*, ROS, VR environments) can seamlessly import the reconstructed scene to facilitate researches in robot TAMP and embodied AI.

This work also motivates three new research questions worth investigating in the future: (i) To sufficiently plan robot tasks, how well should the CAD models replicate the real objects? (ii) Although the proposed system can filter out dynamic entities based on their semantic segmentation (*e.g.*, humans) and a better data association can handle semi-dynamic objects, how could we incorporate the causal relations between environmental changes and human activities? (iii) Although the effects of acting in a sequential task could be updated as the kinematic information in cg, recognizing these effects in physical world introduces extra challenges.

REFERENCES

[1] J. J. Gibson, *The perception of the visual world.* Houghton Mifflin, 1950.

[2] J. J. Gibson, *The senses considered as perceptual systems.* Houghton Mifflin, 1966.

[3] K. Ikeuchi and M. Hebert, "Task oriented vision," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 1992.

[4] Y. Zhu, Y. Zhao, and S.-C. Zhu, "Understanding tools: Task-oriented object modeling, learning and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] D.-C. Hoang, A. J. Lilienthal, and T. Stoyanov, "Panoptic 3d mapping and object pose estimation using adaptively weighted semantic information," *Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 1962–1969, 2020.

[6] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[7] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2015.

[8] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu, "Inferring forces and learning human utilities from videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu, "Beyond point clouds: Scene understanding by reasoning geometry and physics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[10] B. Zheng, Y. Zhao, C. Y. Joey, K. Ikeuchi, and S.-C. Zhu, "Detecting potential falling objects by inferring human action and natural disturbance," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2014.

[11] Y. Zhao and S.-C. Zhu, "Scene parsing by integrating function, geometry and appearance models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[12] B. Zheng, Y. Zhao, J. Yu, K. Ikeuchi, and S.-C. Zhu, "Scene understanding by reasoning stability and safety," *International Journal of Robotics Research (IJRR)*, vol. 112, no. 2, pp. 221–238, 2015.

[13] S. Huang, S. Qi, Y. Xiao, Y. Zhu, Y. N. Wu, and S.-C. Zhu, "Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[14] Y. Chen, S. Huang, T. Yuan, S. Qi, Y. Zhu, and S.-C. Zhu, "Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[15] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2011.

[16] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2014.

[17] B. Kim, Z. Wang, L. P. Kaelbling, and T. Lozano-Pérez, "Learning to guide task and motion planning using score-space representation," *International Journal of Robotics Research (IJRR)*, vol. 38, no. 7, pp. 793–812, 2019.

[18] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Active model learning and diverse action sampling for task and motion planning," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[19] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese, "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments," *Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 713–720, 2020.

[20] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2012.

[22] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *Transactions on Robotics (T-RO)*, vol. 35, no. 4, pp. 925–938, 2019.

[23] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2017.

[24] 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," *Robotics and Automation Letters (RA-L)*, vol. 4, no. 3, pp. 3037–3044, 2019.

[25] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *Proceedings of International Conference on 3D Vision (3DV)*, 2018.

[26] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator.," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[28] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[29] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," 2017.

[30] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] X. Xie, H. Liu, Z. Zhang, Y. Qiu, F. Gao, S. Qi, Y. Zhu, and S.-C. Zhu, "Vrgym: A virtual testbed for physical and interactive ai," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, 2019.

[32] L. F. Yu, S. K. Yeung, C. K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: automatic optimization of furniture arrangement," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, 2011.

[33] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-centric indoor scene synthesis using stochastic grammar," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] C. Jiang, S. Qi, Y. Zhu, S. Huang, J. Lin, L.-F. Yu, D. Terzopoulos, and S.-C. Zhu, "Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars," *International Journal of Computer Vision (IJCV)*, vol. 126, no. 9, pp. 920–941, 2018.

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

[36] H. Fu, B. Cai, L. Gao, L. Zhang, C. Li, Q. Zeng, C. Sun, Y. Fei, Y. Zheng, Y. Li, Y. Liu, P. Liu, L. Ma, L. Weng, X. Hu, X. Ma, Q. Qian, R. Jia, B. Zhao, and H. Zhang, "3d-front: 3d furnished rooms with layouts and semantics," *arXiv preprint arXiv:2011.09127*, 2020.

[37] Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Real-time progressive 3d semantic segmentation for indoor scenes," in *Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[38] S. Yang and S. Scherer, "Monocular object and plane slam in structured environments," *Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 3145–3152, 2019.

[39] K. Wada, E. Sucar, S. James, D. Lenton, and A. J. Davison, "Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Z. Sui, H. Chang, N. Xu, and O. Chadwicke Jenkins, "Geofusion: Geometric consistency informed scene estimation in dense clutter," *Robotics and Automation Letters (RA-L)*, 2020.

[41] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] Q.-H. Pham, T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "Jsis3d: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

[44] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2cad: Learning cad model alignment in rgb-d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[45] A. Avetisyan, T. Khanova, C. Choy, D. Dash, A. Dai, and M. Nießner, "Scenecad: Predicting object alignments and layouts in rgb-d scans," in *The European Conference on Computer Vision (ECCV)*, August 2020.

[46] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *Transactions on Robotics (T-RO)*, vol. 32, no. 6, pp. 1309–1332, 2016.

[47] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2007.

[48] Y. Zhao and S.-C. Zhu, "Image parsing with stochastic scene grammar," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[49] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[51] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[52] S. Huang, S. Qi, Y. Zhu, Y. Xiao, Y. Xu, and S.-C. Zhu, "Holistic 3d scene parsing and reconstruction from a single rgb image," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[53] S. Qi, B. Jia, S. Huang, P. Wei, and S.-C. Zhu, "A generalized earley parser for human activity parsing and prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[54] M. Edmonds, F. Gao, X. Xie, H. Liu, S. Qi, Y. Zhu, B. Rothrock, and S.-C. Zhu, "Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[55] H. Liu, Y. Zhang, W. Si, X. Xie, Y. Zhu, and S.-C. Zhu, "Interactive robot knowledge patching using augmented reality," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2018.

[56] M. Edmonds, F. Gao, H. Liu, X. Xie, S. Qi, B. Rothrock, Y. Zhu, Y. N. Wu, H. Lu, and S.-C. Zhu, "A tale of two explanations: Enhancing human trust by explaining robot behavior," *Science Robotics*, vol. 4, no. 37, 2019.

[57] H. Liu, C. Zhang, Y. Zhu, C. Jiang, and S.-C. Zhu, "Mirroring without overimitation: Learning functionally equivalent manipulation actions," in *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[58] Z. Zhang, Y. Zhu, and S.-C. Zhu, "Graph-based hierarchical knowledge representation for robot task transfer from virtual to physical world," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[59] T. Yuan, H. Liu, L. Fan, Z. Zheng, T. Gao, Y. Zhu, and S.-C. Zhu, "Joint inference of states, robot knowledge, and human (false-) beliefs," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[60] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[61] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane slam for hand-held 3d sensors," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2013.

[62] G. Malandain and J.-D. Boissonnat, "Computing the diameter of a point set," *International Journal of Computational Geometry & Applications*, vol. 12, no. 06, pp. 489–509, 2002.

[63] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[64] 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 *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[65] F. Furrer, T. Novkovic, M. Fehr, A. Gawel, M. Grinvald, T. Sattler, R. Siegwart, and J. Nieto, "Incremental object database: Building 3d models from multiple partial observations," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2018.

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

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

[68] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[69] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*, pp. 105–116, Springer, 1978.

[70] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems," *Artificial intelligence*, vol. 58, no. 1-3, pp. 161–205, 1992.