3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans

Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, Luca Carlone
Laboratory for Information & Decision Systems (LIDS)
Massachusetts Institute of Technology
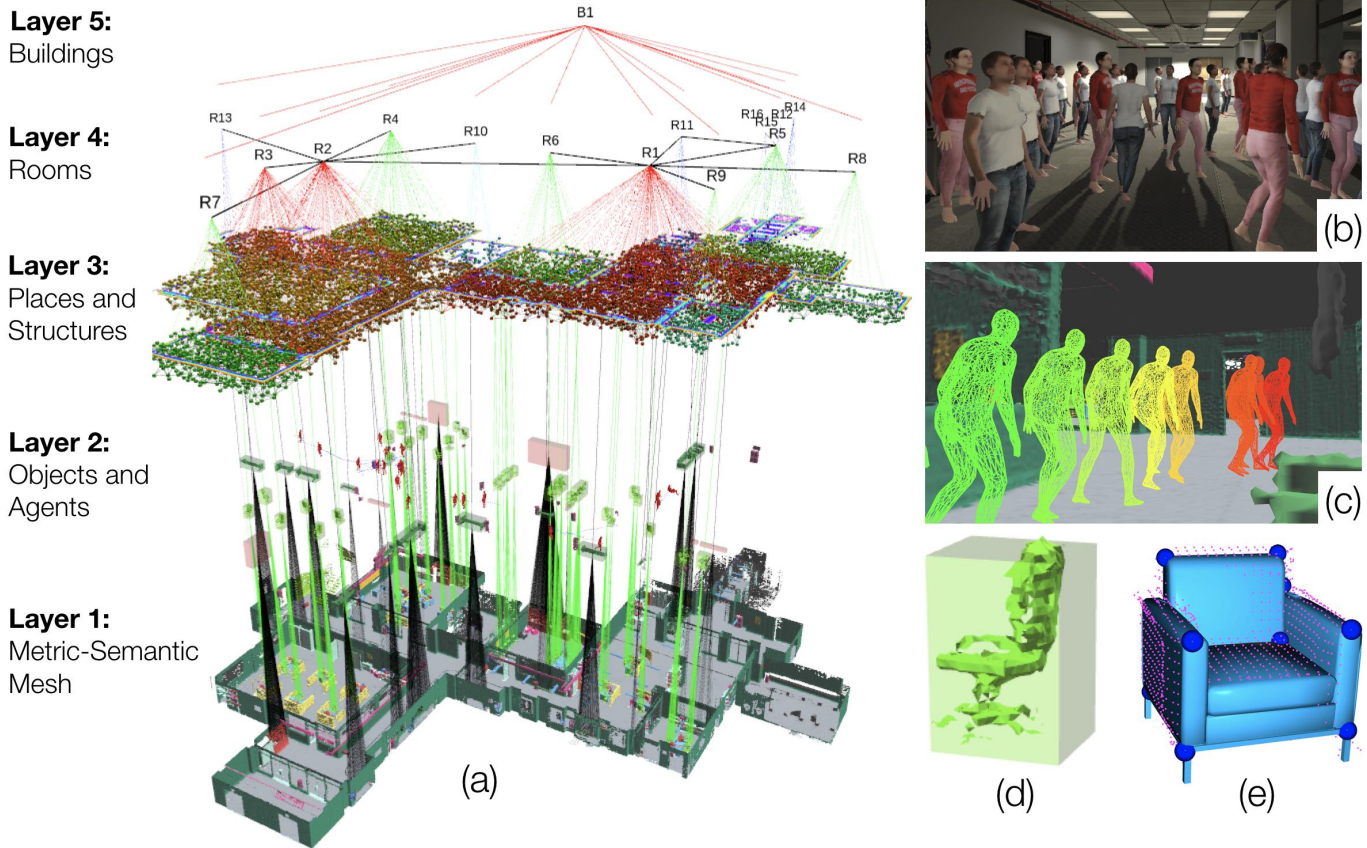{arosinol,agupta,mabate,jnshi,lcarlone}@mit.edu

Fig. 1: We propose **3D Dynamic Scene Graphs** (DSGs) as a unified representation for actionable spatial perception. (a) A DSG is a layered and hierarchical representation that abstracts a dense 3D model (*e.g.,* a metric-semantic mesh) into higher-level *spatial concepts* (*e.g.,* objects, agents, places, rooms) and models their spatio-temporal relations (e.g., "agent A is in room B at time t", traversability between places or rooms). We present a Spatial PerceptIon eNgine (SPIN) that reconstructs a DSG from visual-inertial data, and (a) segments places, structures (*e.g.,* walls), and rooms, (b) is robust to extremely crowded environments, (c) tracks dense mesh models of human agents in real time, (d) estimates centroids and bounding boxes of objects of unknown shape, (e) estimates the 3D pose of objects for which a CAD model is given.

Abstract—We present a unified representation for actionable spatial perception: *3D Dynamic Scene Graphs*. *Scene graphs* are directed graphs where nodes represent entities in the scene (*e.g.,* objects, walls, rooms), and edges represent relations (*e.g.,* inclusion, adjacency) among nodes. *Dynamic* scene graphs (DSGs) extend this notion to represent dynamic scenes with moving *agents* (*e.g.,* humans, robots), and to include actionable information that supports planning and decision-making (*e.g.,* spatio-temporal relations, topology at different levels of abstraction). Our second contribution is to provide the first fully automatic *Spatial PerceptIon eNgine* (SPIN) to build a DSG from visual-inertial data. We integrate state-of-the-art techniques for object and human detection and pose estimation, and we describe how to robustly infer object, robot, and human nodes in crowded scenes. To the best of our knowledge, this is the first paper that reconciles visual-inertial SLAM and dense human mesh tracking. Moreover, we provide algorithms to obtain hierarchical representations of indoor environments (*e.g.,* places, structures, rooms) and their relations. Our third contribution is to demonstrate the proposed spatial perception engine in a photo-realistic Unity-based simulator, where we assess its robustness and expressiveness. Finally, we discuss the implications of our proposal on modern robotics applications. 3D Dynamic Scene Graphs can have a profound impact on planning and decision-making, human-robot interaction, long-term autonomy, and scene prediction. A video abstract is available at **https://youtu.be/SWbofjhyPzI**.

I. INTRODUCTION

Spatial perception and 3D environment understanding are key enablers for high-level task execution in the real world. In order to execute high-level instructions, such as "search for survivors on the second floor of the tall building", a robot needs to ground semantic concepts (survivor, floor, building) into a spatial representation (*i.e.,* a metric map), leading to *metric-semantic* spatial representations that go beyond the map models typically built by SLAM and visual-inertial odometry (VIO) pipelines [15]. In addition, bridging low-level obstacle avoidance and motion planning with high-level task planning requires constructing a world model that captures reality at different levels of abstraction. For instance, while task planning might be effective in describing a sequence of actions to complete a task (*e.g.,* reach the entrance of the building, take the stairs, enter each room), motion planning typically relies on a fine-grained map representation (*e.g.,* a mesh or a volumetric model). Ideally, spatial perception should be able to build a *hierarchy of consistent abstractions* to feed both motion and task planning. The problem becomes even more challenging when autonomous systems are deployed in crowded environments. From self-driving cars to collaborative robots on factory floors, identifying obstacles is not sufficient for safe and effective navigation/action, and it becomes crucial to reason on the *dynamic* entities in the scene (in particular, *humans*) and predict their behavior or intentions [24].

The existing literature falls short of simultaneously addressing these issues (metric-semantic understanding, actionable hierarchical abstractions, modeling of dynamic entities). Early work on map representation in robotics (*e.g.,* [16, 28, 50, 51, 103, 113]) investigates hierarchical representations but mostly in 2D and assuming static environments; moreover, these works were proposed before the "deep learning revolution", hence they could not afford advanced semantic understanding. On the other hand, the quickly growing literature on metric-semantic mapping (*e.g.,* [8, 12, 30, 68, 88, 96, 100]) mostly focuses on "flat" representations (object constellations, metric-semantic meshes or volumetric models) that are not hierarchical in nature. Very recent work [5, 41] attempts to bridge this gap by designing richer representations, called *3D Scene Graphs*. A scene graph is a data structure commonly used in computer graphics and gaming applications that consists of a graph model where nodes represent entities in the scene and edges represent spatial or logical relationships among nodes. While the works [5, 41] pioneered the use of 3D scene graphs in robotics and vision (prior work in vision focused on 2D scene graphs defined in the image space [17, 33, 35, 116]), they have important drawbacks. Kim *et al.* [41] only capture objects and miss multiple levels of abstraction. Armeni *et al.* [5] provide a hierarchical model that is useful for visualization and knowledge organization, but does not capture *actionable* information, such as traversability, which is key to robot navigation. Finally, neither [41] nor [5] account for or model dynamic entities in the environment.

Contributions. We present a unified representation for actionable spatial perception: *3D Dynamic Scene Graphs* (DSGs, Fig. 1). A DSG, introduced in Section III, is a *layered* directed graph where nodes represent *spatial concepts* (*e.g.,* objects, rooms, agents) and edges represent pairwise spatio-temporal relations. The graph is *layered*, in that nodes are grouped into layers that correspond to different levels of abstraction of the scene (*i.e.,* a DSG is a hierarchical representation). Our choice of nodes and edges in the DSG also captures *places* and their connectivity, hence providing a strict generalization of the notion of topological maps [85, 86] and making DSGs an *actionable* representation for navigation and planning. Finally, edges in the DSG capture spatio-temporal relations and explicitly model dynamic entities in the scene, and in particular humans, for which we estimate both 3D poses over time (using a *pose graph* model) and a mesh model.

Our second contribution, presented in Section IV, is to provide the first fully automatic *Spatial PerceptIon eNgine* (SPIN) to build a DSG. While the state of the art [5] assumes an annotated mesh model of the environment is given and relies on a semi-automatic procedure to extract the scene graph, we present a pipeline that starts from visual-inertial data and builds the DSG without human supervision. Towards this goal (i) we integrate state-of-the-art techniques for object [111] and human [46] detection and pose estimation, (ii) we describe how to robustly infer object, robot, and human nodes in cluttered and crowded scenes, and (iii) we provide algorithms to partition an indoor environment into places, structures, and rooms. This is the first paper that integrates visual-inertial SLAM and human mesh tracking (we use SMPL meshes [64]). *The notion of* SPIN *generalizes SLAM, which becomes a module in our pipeline, and augments it to capture relations, dynamics, and high-level abstractions.*

Our third contribution, in Section V, is to demonstrate the proposed spatial perception engine in a Unity-based photo-realistic simulator, where we assess its robustness and expressiveness. We show that our SPIN (i) includes desirable features that improve the robustness of mesh reconstruction and human tracking (drawing connections with the literature on pose graph optimization [15]), (ii) can deal with both objects of known and unknown shape, and (iii) uses a simple-yet-effective heuristic to segment places and rooms in an indoor environment. More extensive and interactive visualizations are given in the video attachment (available at [90]).

Our final contribution, in Section VI, is to discuss several *queries* a DSG can support, and its use as an *actionable* spatial perception model. In particular, we discuss how DSGs can impact planning and decision-making (by providing a representation for hierarchical planning and fast collision checking), human-robot interaction (by providing an interpretable abstraction of the scene), long-term autonomy (by enabling data compression), and scene prediction.

II. RELATED WORK

Scene Graphs. Scene graphs are popular computer graphics models to describe, manipulate, and render complex scenes and are commonly used in game engines [106]. While in gaming applications, these structures are used to describe 3D environments, scene graphs have been mostly used in computer vision to abstract the content of 2D images. Krishna *et al.* [48] use a scene graph to model attributes and relations among

objects in 2D images, relying on manually defined natural language captions. Xu *et al.* [109] and Li *et al.* [59] develop algorithms for 2D scene graph generation. 2D scene graphs have been used for image retrieval [36], captioning [3, 37, 47], high-level understanding [17, 33, 35, 116], visual question-answering [27, 120], and action detection [60, 65, 114].

Armeni *et al.* [5] propose a *3D scene graph* model to describe 3D static scenes, and describe a semi-automatic algorithm to build the scene graph. In parallel to [5], Kim *et al.* [41] propose a 3D scene graph model for robotics, which however only includes objects as nodes and misses multiple levels of abstraction afforded by [5] and by our proposal.

Representations and Abstractions in Robotics. The question of world modeling and map representations has been central in the robotics community since its inception [15, 101]. The need to use hierarchical maps that capture rich spatial and semantic information was already recognized in seminal papers by Kuipers, Chatila, and Laumond [16, 50, 51]. Vasudevan *et al.* [103] propose a hierarchical representation of object constellations. Galindo *et al.* [28] use two parallel hierarchical representations (a spatial and a semantic representation) that are then *anchored* to each other and estimated using 2D lidar data. Ruiz-Sarmiento *et al.* [92] extend the framework in [28] to account for uncertain groundings between spatial and semantic elements. Zender *et al.* [113] propose a single hierarchical representation that includes a 2D map, a navigation graph and a topological map [85, 86], which are then further abstracted into a *conceptual map*. Note that the spatial hierarchies in [28] and [113] already resemble a scene graph, with less articulated set of nodes and layers. A more fundamental difference is the fact that early work (i) did not reason over 3D models (but focused on 2D occupancy maps), (ii) did not tackle dynamical scenes, and (iii) did not include dense (e.g., pixel-wise) semantic information, which has been enabled in recent years by deep learning methods.

Metric-Semantic Scene Reconstruction. This line of work is concerned with estimating metric-semantic (but typically non-hierarchical) representations from sensor data. While early work [7, 14] focused on offline processing, recent years have seen a surge of interest towards *real-time* metric-semantic mapping, triggered by pioneering works such as SLAM++ [96]. *Object-based approaches* compute an object map and include SLAM++ [96], XIVO [21], OrcVIO [98], QuadricSLAM [74], and [12]. For most robotics applications, an object-based map does not provide enough resolution for navigation and obstacle avoidance. *Dense approaches* build denser semantically annotated models in the form of point clouds [8, 22, 61, 100], meshes [30, 88, 91], surfels [104, 107], or volumetric models [30, 68, 72]. Other approaches use both objects and dense models, see Li *et al.* [55] and Fusion++ [69]. These approaches focus on static environments. Approaches that deal with moving objects, such as DynamicFusion [73], Mask-fusion [94], Co-fusion [93], and MID-Fusion [108] are currently limited to small table-top scenes and focus on objects or dense maps, rather than scene graphs.

Metric-to-Topological Scene Parsing. This line of work focuses on partitioning a metric map into semantically meaningful places (*e.g.,* rooms, hallways). Nüchter and Hertzberg [75] encode relations among planar surfaces (*e.g.,* walls, floor, ceiling) and detect objects in the scene. Blanco *et al.* [10] propose a hybrid metric-topological map. Friedman *et al.* [26] propose *Voronoi Random Fields* to obtain an abstract model of a 2D grid map. Rogers and Christensen [87] and Lin *et al.* [62] leverage objects to perform a joint object-and-place classification. Pangercic *et al.* [80] reason on the objects' functionality. Pronobis and Jensfelt [83] use a Markov Random Field to segment a 2D grid map. Zheng *et al.* [118] infer the topology of a grid map using a *Graph-Structured Sum-Product Network*, while Zheng and Pronobis [117] use a neural network. Armeni *et al.* [4] focus on a 3D mesh, and propose a method to parse a building into rooms. Floor plan estimation has been also investigated using single images [32], omnidirectional images [66], 2D lidar [56, 102], 3D lidar [71, 76], RGB-D [63], or from crowd-sourced mobile-phone trajectories [2]. The works [4, 71, 76] are closest to our proposal, but contrarily to [4] we do not rely on a Manhattan World assumption, and contrarily to [71, 76] we operate on a mesh model.

SLAM and VIO in Dynamic Environments. This paper is also concerned with modeling and gaining robustness against dynamic elements in the scene. SLAM and moving object tracking has been extensively investigated in robotics [6, 105], while more recent work focuses on joint visual-inertial odometry and target pose estimation [23, 29, 84]. Most of the existing literature in robotics models the dynamic targets as a single 3D point [18], or with a 3D pose and rely on lidar [6], RGB-D cameras [1], monocular cameras [58], and visual-inertial sensing [84]. Related work also attempts to gain robustness against dynamic scenes by using IMU motion information [34], or masking portions of the scene corresponding to dynamic elements [9, 13, 19]. To the best of our knowledge, the present paper is the first work that attempts to perform visual-inertial SLAM, segment dense object models, estimate the 3D poses of known objects, and reconstruct and track dense human SMPL meshes.

Human Pose Estimation. Human pose and shape estimation from a single image is a growing research area. While we refer the reader to [45, 46] for a broader review, it is worth mentioning that related work includes optimization-based approaches, which fit a 3D mesh to 2D image keypoints [11, 45, 54, 110, 112], and learning-based methods, which infer the mesh directly from pixel information [39, 45, 46, 79, 81, 99]. Human models are typically parametrized using the *Skinned Multi-Person Linear Model* (SMPL) [64], which provides a compact pose and shape description and can be rendered as a mesh with 6890 vertices and 23 joints.

III. 3D DYNAMIC SCENE GRAPHS

A 3D *Dynamic Scene Graph* (DSG, Fig. 1) is an actionable spatial representation that captures the 3D geometry and semantics of a scene at different levels of abstraction, and models objects, places, structures, and agents and their relations. More formally, a DSG is a *layered directed graph* where nodes represent *spatial concepts* (*e.g.,* objects, rooms, agents) and edges represent pairwise spatio-temporal relations (*e.g.,* "agent A is in room B at time t"). Contrarily to
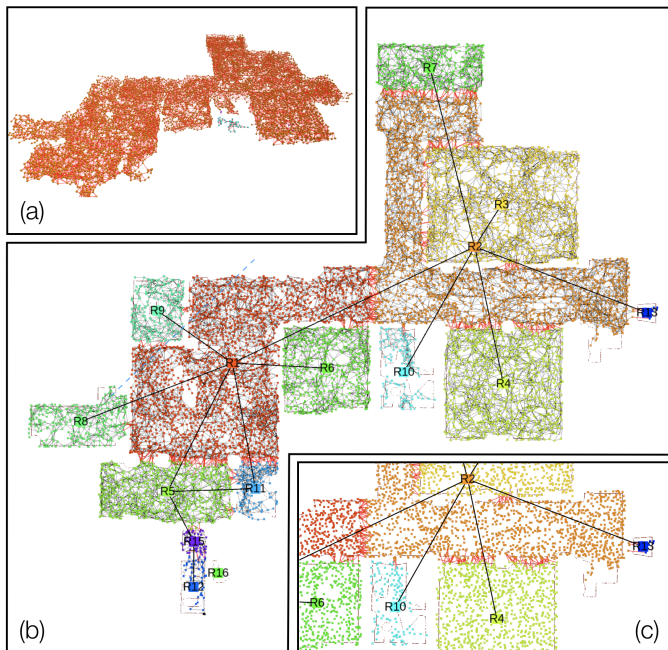
Fig. 2: **Places** and their connectivity shown as a graph. (a) Skeleton (places and topology) produced by [78] (side view); (b) **Room** parsing produced by our approach (top-down view); (c) Zoomed-in view; red edges connect different rooms.

knowledge bases [49], spatial concepts are semantic concepts that are *spatially grounded* (in other words, each node in our DSG includes spatial coordinates and shape or bounding-box information as attributes). A DSG is a *layered* graph, *i.e.,* nodes are grouped into layers that correspond to different levels of abstraction. Every node has a unique ID.

The DSG of a single-story indoor environment includes 5 layers (from low to high abstraction level): (i) Metric-Semantic Mesh, (ii) Objects and Agents, (iii) Places and Structures, (iv) Rooms, and (v) Building. We discuss each layer and the corresponding nodes and edges below.

A. Layer 1: Metric-Semantic Mesh

The lower layer of a DSG is a semantically annotated 3D mesh (bottom of Fig. 1(a)). The nodes in this layer are 3D points (vertices of the mesh) and each node has the following attributes: (i) 3D position, (ii) normal, (iii) RGB color, and (iv) a panoptic semantic label.[1] Edges connecting triplets of points (*i.e.,* a clique with 3 nodes) describe faces in the mesh and define the topology of the environment. Our metric-semantic mesh includes everything in the environment that is *static*, while for storage convenience we store meshes of dynamic objects in a separate structure (see "Agents" below).

B. Layer 2: Objects and Agents

This layer contains two types of nodes: objects and agents (Fig. 1(c-e)), whose main distinction is the fact that agents are time-varying entities, while objects are static.

Objects represent static elements in the environment that are not considered *structural* (*i.e.,* walls, floor, ceiling, pillars

are considered *structure* and are not modeled in this layer). Each object is a node and node attributes include (i) a 3D object pose, (ii) a bounding box, and (ii) its semantic class (*e.g.,* chair, desk). While not investigated in this paper, we refer the reader to [5] for a more comprehensive list of attributes, including materials and affordances. Edges between objects describe relations, such as co-visibility, relative size, distance, or contact ("the cup is on the desk"). Each object node is connected to the corresponding set of points belonging to the object in the Metric-Semantic Mesh. Moreover, nearby objects are connected to the same *place* node (see Section III-C).

Agents represent dynamic entities in the environment, including humans. While in general there might be many types of dynamic entities (*e.g.,* vehicles, bicycles in outdoor environments), without loss of generality here we focus on two classes: *humans* and *robots*.[2] Both human and robot nodes have three attributes: (i) a 3D pose graph describing their trajectory over time, (ii) a mesh model describing their (non-rigid) shape, and (iii) a semantic class (*i.e.,* human, robot). A pose graph [15] is a collection of time-stamped 3D poses where edges model pairwise relative measurements. The robot collecting the data is also modeled as an agent in this layer.

C. Layer 3: Places and Structures

This layer contains two types of nodes: places and structures. Intuitively, places are a model for the free space, while structures capture separators between different spaces.

Places (Fig. 2) correspond to positions in the free-space and edges between places represent traversability (in particular: presence of a straight-line path between places). Places and their connectivity form a *topological map* [85, 86] that can be used for path planning. Place attributes only include a 3D position, but can also include a semantic class (*e.g.,* back or front of the room) and an obstacle-free bounding box around the place position. Each object and agent in Layer 2 is connected with the nearest place (for agents, the connection is for each time-stamped pose, since agents move from place to place). Places belonging to the same room are also connected to the same room node in Layer 4. Fig. 2(b-c) shows a visualization with places color-coded by rooms.

Structures (Fig. 3) include nodes describing structural elements in the environment, *e.g.,* walls, floor, ceiling, pillars. The notion of structure captures elements often called "stuff" in related work [57], while we believe the name "structure" is more evocative and useful to contrast them to objects. Structure nodes' attributes are: (i) 3D pose, (ii) bounding box, and (iii) semantic class (*e.g.,* walls, floor). Structures may have edges to the rooms they enclose. Structures may also have edges to an object in Layer 3, *e.g.,* a "frame" (object) "is hung" (relation) on a "wall" (structure), or a "ceiling light is mounted on the ceiling".

---

[1]Panoptic segmentation [42, 57] segments both object (*e.g.,* chairs, tables, drawers) instances and structures (*e.g.,* walls, ground, ceiling).

[2]These classes can be considered instantiations of more general concepts: "rigid" agents (such as robots, for which we only need to keep track a 3D pose), and "deformable" agents (such as humans, for which we also need to keep track of a time-varying shape).
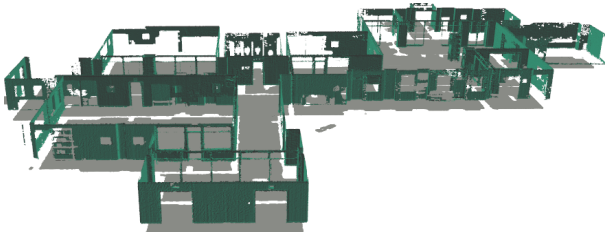
Fig. 3: **Structures**: exploded view of walls and floor.

D. Layer 4: Rooms

This layer includes nodes describing rooms, corridors, and halls. Room nodes (Fig. 2) have the following attributes: (i) 3D pose, (ii) bounding box, and (iii) semantic class (*e.g.,* kitchen, dining room, corridor). Two rooms are connected by an edge if they are adjacent (*i.e.,* there is a door connecting them). A room node has edges to the places (Layer 3) it contains (since each place is connected to nearby objects, the DSG also captures which object/agent is contained in each room). All rooms are connected to the building they belong to (Layer 5).

E. Layer 5: Building

Since we are considering a representation over a single building, there is a single *building node* with the following attributes: (i) 3D pose, (ii) bounding box, and (iii) semantic class (*e.g.,* office building, residential house). The building node has edges towards all rooms in the building.

F. Composition and Queries

Why should we choose this set of nodes or edges rather than a different one? Clearly, the choice of nodes in the DSG is not unique and is task-dependent. Here we first motivate our choice of nodes in terms of *planning queries* the DSG is designed for (see Remark 1 and the broader discussion in Section VI), and we then show that the representation is compositional, in the sense that it can be easily expanded to encompass more layers, nodes, and edges (Remark 2).

Remark 1 (Planning Queries): The proposed DSG is designed with task and motion planning queries in mind. The semantic node attributes (*e.g.,* semantic class) support planning from high-level specification ("pick up the red cup from the table in the dining room"). The geometric node attributes (*e.g.,* meshes, positions, bounding boxes) and the edges are used for motion planning. For instance, the places can be used as a topological graph for path planning, and the bounding boxes can be used for fast collision checking.

Remark 2 (Composition of DSGs): A second re-ensuring property of a DSG is its compositionality: one can easily concatenate more layers at the top and the bottom of the DSG in Fig. 1(a), and even add intermediate layers. For instance, in a multi-story building, we can include a "Level" layer between the "Building" and "Rooms" layers in Fig. 1(a). Moreover, we can add further abstractions or layers at the top, for instance going from buildings to neighborhoods, and then to cities.

IV. SPATIAL PERCEPTION ENGINE: BUILDING A 3D DSGs FROM SENSOR DATA

This section describes a *Spatial PerceptIon eNgine* (SPIN) that populates the DSG nodes and edges using sensor data. The input to our SPIN is streaming data from a stereo camera and an Inertial Measurement Unit (IMU). The output is a 3D DSG. In our current implementation, the metric-semantic mesh and the agent nodes are incrementally built from sensor data in real-time, while the remaining nodes (objects, places, structure, rooms) are automatically built at the end of the run.

Section IV-A describes how to obtain the metric-semantic mesh and agent nodes from sensor data. Section IV-B describes how to segment and localize objects. Section IV-C describes how to parse places, structures, and rooms.

A. From Visual-Inertial data to Mesh and Agents

Metric-Semantic Mesh. We use Kimera [88] to reconstruct a semantically annotated 3D mesh from visual-inertial data in real-time. Kimera is open source and includes four main modules: (i) Kimera-VIO: a visual-inertial odometry module implementing IMU preintegration and fixed-lag smoothing [25], (ii) Kimera-RPGO: a robust pose graph optimizer [67], (iii) Kimera-Mesher: a per-frame and multi-frame mesher [89], and (iv) Kimera-Semantics: a volumetric approach to produce a semantically annotated mesh and an Euclidean Signed Distance Function (ESDF) based on Voxblox [77]. Kimera-Semantics uses a panoptic 2D semantic segmentation of the left camera images to label the 3D mesh using Bayesian updates. We take the metric-semantic mesh produced by Kimera-Semantics as Layer 1 in the DSG in Fig. 1(a).

Robot Node. In our setup the only robotic agent is the one collecting the data, hence Kimera-RPGO directly produces a time-stamped pose graph describing the poses of the robot at discrete time stamps. Since our robot moves in crowded environments, we replace the Lukas-Kanade tracker in the VIO front-end of [88] with an IMU-aware optical flow method, where feature motion between frames is predicted using IMU motion information, similar to [34]. Moreover, we use a 2-point RANSAC [43] for geometric verification, which directly uses the IMU rotation to prune outlier correspondences in the feature tracks. To complete the robot node, we assume a CAD model of the robot to be given (only used for visualization).

Human Nodes. Contrary to related work that models dynamic targets as a point or a 3D pose [1, 6, 18, 58, 84], we track a dense time-varying mesh model describing the shape of the human over time. Therefore, to create a human node our SPIN needs to detect and estimate the shape of a human in the camera images, and then track the human over time. For shape estimation, we use the Graph-CNN approach of Kolotouros *et al.* [46], which directly regresses the 3D location of the vertices of an SMPL [64] mesh model from a single image. An example is given in Fig. 4(a-b). More in detail, given a panoptic 2D segmentation, we crop the left camera image to a bounding box around each detected human, and we use the approach [46] to get a 3D SMPL. We then extract the full pose in the original perspective camera frame ([46] uses a weak perspective camera model) using PnP [119].

To track a human, our SPIN builds a pose graph where each node is assigned the pose of the torso of the human at a discrete time. Consecutive poses are connected by a factor [20] modeling a zero velocity prior. Then, each detection at time t is modeled as a prior factor on the pose at time t. For each node

of the pose graph, our SPIN also stores the 3D mesh estimated by [46]. For this approach to work reliably, outlier rejection and data association become particularly important. The approach of [46] often produces largely incorrect poses when the human is partially occluded. Moreover, in the presence of multiple humans, one has to associate each detection d_t to one of the human pose graphs $h_{1:t-1}^{(i)}$ (including poses from time 1 to $t-1$ for each human $i = 1, 2, \ldots$). To gain robustness, our SPIN (i) rejects detections when the bounding box of the human approaches the boundary of the image or is too small (≤ 30 pixels in our tests), and (ii) adds a measurement to the pose graph only when the human mesh detected at time t is "consistent" with the mesh of one of the humans at time $t-1$. To check consistency, we extract the skeleton at time $t-1$ (from the pose graph) and t (from the current detection) and check that the motion of each joint (Fig. 4(c)) is physically plausible in that time interval (*i.e.,* we leverage the fact that the joint and torso motion cannot be arbitrarily fast). We use a conservative bound of 3m on the maximum allowable joint displacement in a time interval of 1 second. If no pose graph meets the consistency criterion, we initialize a new pose graph with a single node corresponding to the current detection.

Besides using them for tracking, we feed back the human detections to Kimera-Semantics, such that dynamic elements are not reconstructed in the 3D mesh. We achieve this by only using the free-space information when ray casting the depth for pixels labeled as humans, an approach we dubbed *dynamic masking* (see results in Fig. 5).
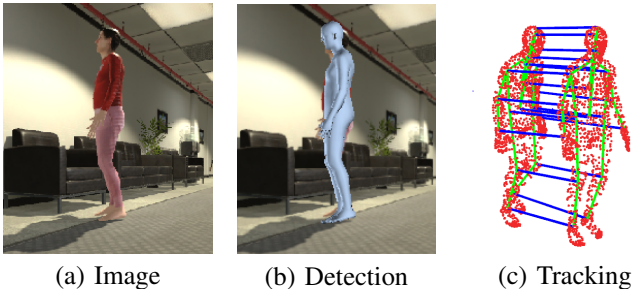

(a) Image   (b) Detection   (c) Tracking

Fig. 4: **Human nodes**: (a) Input camera image from Unity, (b) SMPL mesh detection and pose/shape estimation using [46], (c) Temporal tracking and consistency checking on the maximum joint displacement between detections.

B. From Mesh to Objects

Our spatial perception engine extracts static objects from the metric-semantic mesh produced by Kimera. We give the user the flexibility to provide a catalog of CAD models for some of the object classes. If a shape is available, our SPIN will try to fit it to the mesh (paragraph "Objects with Known Shape" below), otherwise will only attempt to estimate a centroid and bounding box (paragraph "Objects with Unknown Shape").

Objects with Unknown Shape. The metric semantic mesh from Kimera already contains semantic labels. Therefore, our SPIN first exacts the portion of the mesh belonging to a given object class (*e.g.,* chairs in Fig. 1(d)); this mesh potentially contains multiple object instances belonging to the same class. Then, it performs Euclidean clustering using PCL [95] (with a distance threshold of twice the voxel size used in Kimera-Semantics, which is 0.1m) to segment the object mesh into instances. From the segmented clusters, our SPIN obtains a centroid of the object (from the vertices of the corresponding mesh), and assigns a canonical orientation with axes aligned with the world frame. Finally, it computes a bounding box with axes aligned with the canonical orientation.

Objects with Known Shape. For objects with known shape, our SPIN isolates the mesh corresponding to an object instance, similarly to the unknown-shape case. However, if a CAD model for that class of objects is given, our SPIN attempts fitting the known shape to the object mesh. This is done in three steps. First, we extract 3D keypoints from the CAD model of the object, and the corresponding object mesh from Kimera. The 3D keypoints are extracted by transforming each mesh to a point cloud (by picking the vertices of the mesh) and then extracting 3D Harris corners [95] with 0.15m radius and 10^{-4} non-maximum suppression threshold. Second, we match every keypoint on the CAD model with any keypoint on the Kimera model. Clearly, this step produces many incorrect putative matches (outliers). Third, we apply a robust open-source registration technique, TEASER++ [111], to find the best alignment between the point clouds in the presence of extreme outliers. The output of these three steps is a 3D pose of the object (from which it is also easy to extract an axis-aligned bounding box), see result in Fig. 1(e).

C. From Mesh to Places, Structures, and Rooms

This section describes how our SPIN leverages existing techniques and implements simple-yet-effective methods to parse places, structures, and rooms from Kimera's 3D mesh.

Places. Kimera uses Voxblox [77] to extract a global mesh and an ESDF. We also obtain a topological graph from the ESDF using [78], where nodes sparsely sample the free space, while edges represent straight-line traversability between two nodes. We directly use this graph to extract the places and their topology (Fig. 2(a)). After creating the places, we associate each object and agent pose to the nearest place to model a proximity relation.

Structures. Kimera's semantic mesh already includes different labels for walls, ground floor, and ceiling, so isolating these three structural elements is straightforward (Fig. 3). For each type of structure, we then compute a centroid, assign a canonical orientation (aligned with the world frame), and compute an axis-aligned bounding box.

Rooms. While floor plan computation is challenging in general, (i) the availability of a 3D ESDF and (ii) the knowledge of the gravity direction given by Kimera enable a simple-yet-effective approach to partition the environment into different rooms. The key insight is that an horizontal 2D section of the 3D ESDF, cut below the level of the detected ceiling, is relatively unaffected by clutter in the room. This 2D section gives a clear signature of the room layout: the voxels in the section have a value of 0.3m almost everywhere (corresponding to the distance to the ceiling), except close to the walls, where the distance decreases to 0m. We refer to this 2D ESDF (cut at 0.3m below the ceiling) as an *ESDF section*.

To compensate for noise, we further truncate the ESDF section to distances above 0.2m, such that small openings

between rooms (possibly resulting from error accumulation) are removed. The result of this partitioning operation is a set of disconnected 2D ESDFs corresponding to each room, that we refer to as *2D ESDF rooms*. Then, we label all the "Places" (nodes in Layer 3) that fall inside a 2D ESDF room depending on their 2D (horizontal) position. At this point, some places might not be labeled (those close to walls or inside door openings). To label these, we use majority voting over the neighborhood of each node in the topological graph of "Places" in Layer 3; we repeat majority voting until all places have a label. Finally, we add an edge between each place (Layer 3) and its corresponding room (Layer 4), see Fig. 2(b-c), and add an edge between two rooms (Layer 4) if there is an edge connecting two of its places (red edges in Fig. 2(b-c)). We also refer the reader to the video attachment.

V. EXPERIMENTS IN PHOTO-REALISTIC SIMULATOR

This section shows that the proposed SPIN (i) produces accurate metric-semantic meshes and robot nodes in crowded environments (Section V-A), (ii) correctly instantiates object and agent nodes (Section V-B), and (iii) reliably parses large indoor environments into rooms (Section V-C).

Testing Setup. We use a photo-realistic Unity-based simulator to test our spatial perception engine in a 65m×65m simulated office environment. The simulator also provides the 2D panoptic semantic segmentation for Kimera. Humans are simulated using the realistic 3D models provided by the SMPL project [64]. The simulator provides ground-truth poses of humans and objects, which are only used for benchmarking. Using this setup, we create 3 large visual-inertial datasets, that we release as part of the uHumans dataset [90]. The datasets, labeled as uH_01, uH_02, uH_03, include 12, 24, and 60 humans, respectively. We use the human pose and shape estimator [46] out of the box, without any domain adaptation or retraining.

A. Robustness of Mesh Reconstruction in Crowded Scenes

Here we show that IMU-aware feature tracking and the use of a 2-point RANSAC in Kimera enhance VIO robustness. Moreover, we show that this enhanced robustness, combined with *dynamic masking* (Section IV-A), results in robust and accurate metric-semantic meshes in crowded environments.

Enhanced VIO. Table I reports the absolute trajectory errors of Kimera with and without the use of 2-point RANSAC and when using 2-point RANSAC and IMU-aware feature tracking (label: DVIO). Best results (lowest errors) are shown in bold. The left part of the table (MH_01–V2_03) corresponds to tests on the (static) EuRoC dataset. The results confirm that in absence of dynamic agents the proposed approach performs on-par with the state of the art, while the use of 2-point RANSAC already boosts performance. The last three columns (uH_01–uH_03), however, show that in the presence of dynamic entities, the proposed approach dominates the baseline (Kimera-VIO).

Dynamic Masking. Fig. 5 visualizes the effect of dynamic masking on Kimera's metric-semantic mesh reconstruction. Fig. 5(a) shows that without dynamic masking a human walking in front of the camera leaves a "contrail" (in cyan) and creates artifacts in the mesh. Fig. 5(b) shows that dynamic

TABLE I: VIO errors in centimeters on the EuRoC (MH and V) and uHumans (uH) datasets.

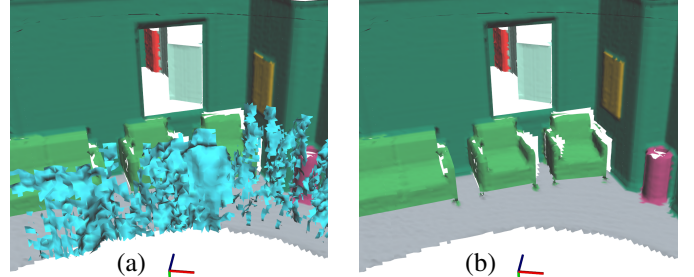| Seq.	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03	uH_01	uH_02	uH_03
5-point	9.3	10	11	42	21	6.7	12	17	**5**	**8.1**	30	92	145	160
2-point	9.0	10	**10**	31	**16**	4.7	**7.5**	**14**	5.8	9	**20**	78	79	111
DVIO	**8.1**	**9.8**	14	**23**	20	**4.3**	7.8	17	6.2	11	30	**59**	**78**	**88**

Fig. 5: 3D mesh reconstruction (a) without and (b) with *dynamic masking*.

masking avoids this issue and leads to clean mesh reconstructions. Table II reports the RMSE mesh error (see *accuracy metric* in [89]) with and without dynamic masking (label: "with DM" and "w/o DM"). To assess the mesh accuracy independently from the VIO accuracy, we also report the mesh error when using ground-truth poses (label: "GT Poses" in the table), besides the results with the VIO poses (label: "DVIO Poses"). The "GT Poses" columns in the table show that even with a perfect localization, the artifacts created by dynamic entities (and visualized in Fig. 5(a)) significantly hinder the mesh accuracy, while dynamic masking ensures highly accurate reconstructions. The advantage of dynamic masking is preserved when VIO poses are used.

TABLE II: Mesh error in meters with and without dynamic masking (DM).

Seq.	GT Pose w/o DM	GT Poses with DM	DVIO Poses w/o DM	DVIO Poses with DM
uH_01	0.089	0.060	0.227	0.227
uH_02	0.133	0.061	0.347	0.301
uH_03	0.192	0.061	0.351	0.335

B. Parsing Humans and Objects

Here we evaluate the accuracy of human tracking and object localization on the uHumans datasets.

Human Nodes. Table III shows the average localization error (mismatch between the torso estimated position and the ground truth) for each human on the uHumans datasets. The first column reports the error of the detections produced by [46] (label: "Single-img."). The second column reports the error for the case in which we filter out detections when the human is only partially visible in the camera image, or when the bounding box of the human is too small ($\leq$ 30 pixels, label: "Single-img. filtered"). The third column reports errors with the proposed pose graph model discussed in Section IV-A (label: "Tracking"). The approach [46] tends to produce incorrect estimates when the human is occluded. Filtering out detections improves the localization performance, but occlusions due to objects in the scene still result in significant errors. Instead, the proposed approach ensures accurate human tracking.

TABLE III: Human and object localization errors in meters.

	Humans			Objects	
Seq.	Single-img. [46]	Single-img. filtered	Tracking (proposed)	Unknown Objects	Known Objects
uH_01	1.07	0.88	0.65	1.31	0.20
uH_02	1.09	0.78	0.61	1.70	0.35
uH_03	1.20	0.97	0.63	1.51	0.38

Object Nodes. The last two columns of Table III report the average localization errors for objects of unknown and known shape detected in the scene. In both cases, we compute the localization error as the distance between the estimated and the ground truth centroid of the object (for the objects with known shape, we use the centroid of the fitted CAD model). We use CAD models for objects classified as "couch". In both cases, we can correctly localize the objects, while the availability of a CAD model further boosts accuracy.

C. Parsing Places and Rooms

The quality of the extracted places and rooms can be seen in Fig. 2. We also compute the average precision and recall for the classification of places into rooms. The ground truth labels are obtained by manually segmenting the places. For uH_01 we obtain an average precision of 99.89% and an average recall of 99.84%. Incorrect classifications typically occur near doors, where room misclassification is inconsequential.

VI. DISCUSSION: QUERIES AND OPPORTUNITIES

We highlight the actionable nature of a 3D Dynamic Scene Graph by providing examples of queries it enables.

Obstacle Avoidance and Planning. Agents, objects, and rooms in our DSG have a bounding box attribute. Moreover, the hierarchical nature of the DSG ensures that bounding boxes at higher layers contain bounding boxes at lower layers (*e.g.,* the bounding box of a room contains the objects in that room). This forms a *Bounding Volume Hierarchy* (BVH) [53], which is extensively used for collision checking in computer graphics. BVHs provide readily available opportunities to speed up obstacle avoidance and motion planning queries where collision checking is often used as a primitive [40].

DSGs also provide a powerful tool for high-level planning queries. For instance, the (connected) subgraph of places and objects in a DSG can be used to issue the robot a high-level command (*e.g.,* object search [38]), and the robot can directly infer the closest place in the DSG it has to reach to complete the task, and can plan a feasible path to that place.

The multiple levels of abstraction afforded by a DSG have the potential to enable hierarchical and multi-resolution planning approaches [52, 97], where a robot can plan at different levels of abstraction to save computational resources.

Human-Robot Interaction. As already explored in [5, 41], a scene graph can support user-oriented tasks, such as interactive visualization and *Question Answering*. Our Dynamic Scene Graph extends the reach of [5, 41] by (i) allowing visualization of human trajectories and dense poses (see visualization in the video attachment), and (ii) enabling more complex and time-aware queries such as "where was this person at time t?", or "which object did this person pick in Room A?".

Furthermore, DSGs provide a framework to model plausible interactions between agents and scenes [31, 70, 82, 115]. We believe DSGs also complement the work on natural language grounding [44], where one of the main concerns is to reason over the variability of human instructions.

Long-term Autonomy. DSGs provide a natural way to "forget" or retain information in long-term autonomy. By construction, higher layers in the DSG hierarchy are more compact and abstract representations of the environment, hence the robot can "forget" portions of the environment that are not frequently observed by simply pruning the corresponding branch of the DSG. For instance, to forget a room in Fig. 1, we only need to prune the corresponding node and the connected nodes at lower layers (places, objects, etc.). More importantly, the robot can selectively decide which information to retain: for instance, it can keep all the objects (which are typically fairly cheap to store), but can selectively forget the mesh model, which can be more cumbersome to store in large environments. Finally, DSGs inherit memory advantages afforded by standard scene graphs: if the robot detects N instances of a known object (*e.g.,* a chair), it can simply store a *single* CAD model and cross-reference it in N nodes of the scene graph; this simple observation enables further data compression.

Prediction. The combination of a dense metric-semantic mesh model and a rich description of the agents allows performing short-term predictions of the scene dynamics and answering queries about possible future outcomes. For instance, one can feed the mesh model to a physics simulator and roll out potential high-level actions of the human agents;

VII. CONCLUSION

We introduced *3D Dynamic Scene Graphs* as a unified representation for actionable spatial perception, and presented the first *Spatial PerceptIon eNgine* (SPIN) that builds a DSG from sensor data in a fully automatic fashion. We showcased our SPIN in a photo-realistic simulator, and discussed its application to several queries, including planning, human-robot interaction, data compression, and scene prediction. This paper opens several research avenues. First of all, many of the queries in Section VI involve nontrivial research questions and deserve further investigation. Second, more research is needed to expand the reach of DSGs, for instance by developing algorithms that can infer other node attributes from data (*e.g.,* material type and affordances for objects) or creating new node types for different environments (*e.g.,* outdoors). Third, this paper only scratches the surface in the design of spatial perception engines, thus leaving many questions unanswered: is it advantageous to design SPINs for other sensor combinations? Can we estimate a scene graph incrementally and in real-time? Can we design distributed SPINs to estimate a DSG from data collected by multiple robots?

REFERENCES

[1] A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. Di Stefano, and M. Vincze. Multimodal cue integration through hypotheses verification for rgb-d object recognition and 6dof pose estimation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2104–2111, 2013. 3, 5

[2] M. Alzantot and M. Youssef. Crowdinside: Automatic construction of indoor floorplans. In *Proc. of the 20th International Conference on Advances in Geographic Information Systems*, pages 99–108, 2012. 3

[3] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 382–398, 2016. 3

[4] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, 2016. 3

[5] 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 *Intl. Conf. on Computer Vision (ICCV)*, pages 5664–5673, 2019. 2, 3, 4, 8

[6] A. Azim and O. Aycard. Detection, classification and tracking of moving objects in a 3d environment. In *2012 IEEE Intelligent Vehicles Symposium*, pages 802–807, 2012. 3, 5

[7] S. Y.-Z. Bao and S. Savarese. Semantic structure from motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011. 3

[8] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Intl. Conf. on Computer Vision (ICCV)*, 2019. 2, 3

[9] B. Bescos, J. M. Fácil, J. Civera, and J. Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 3

[10] J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal. Subjective local maps for hybrid metric-topological slam. *Robotics and Autonomous Systems*, 57:64–74, 2009. 3

[11] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3d human pose and shape from a single image. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conf. on Computer Vision (ECCV)*, 2016. 3

[12] S. Bowman, N. Atanasov, K. Daniilidis, and G. Pappas. Probabilistic data association for semantic slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1722–1729, 2017. 2, 3

[13] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari. Semantic monocular slam for highly dynamic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 393–400, 2018. 3

[14] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conf. on Computer Vision (ECCV)*, pages 44–57, 2008. 3

[15] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics*, 32(6):1309–1332, 2016. arxiv preprint: 1606.05830. 2, 3, 4

[16] R. Chatila and J.-P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 138–145, 1985. 2, 3

[17] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 33–40, 2013. 2, 3

[18] M. Chojnacki and V. Indelman. Vision-based dynamic target trajectory and ego-motion estimation using incremental light bundle adjustment. *International Journal of Micro Air Vehicles*, 10(2):157–170, 2018. 3, 5

[19] L. Cui and C. Ma. Sof-slam: A semantic visual slam for dynamic environments. *IEEE Access*, 7:166528–166539, 2019. 3

[20] F. Dellaert and M. Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1-2):1–139, 2017. 5

[21] J. Dong, X. Fei, and S. Soatto. Visual-inertial-semantic scene representation for 3D object detection. 2017. 3

[22] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena. SegMap: 3d segment mapping using data-driven descriptors. In *Robotics: Science and Systems (RSS)*, 2018. 3

[23] K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang. Tightly-coupled visual-inertial localization and 3D rigid-body target tracking. *IEEE Robotics and Automation Letters*, 4(2):1541–1548, 2019. 3

[24] M. Everett, Y. F. Chen, and J. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning, 05 2018. 2

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration theory for fast and accurate visual-inertial navigation. *IEEE Trans. Robotics*, 33(1):1–21, 2017. 5

[26] S. Friedman, H. Pasula, and D. Fox. Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In *Intl. Joint Conf. on AI (IJCAI)*, page 2109âĂŞ2114, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. 3

[27] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. 2016. arXiv preprint arXiv:1606.01847. 3

[28] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González. Multi-hierarchical semantic maps for mobile robotics. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3492–3497, 2005. 2, 3

[29] P. Geneva, J. Maley, and G. Huang. Schmidt-EKF-based visual-inertial moving object tracking. *ArXiv Preprint: 1903.0863*, 2019. 3

[30] 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, 2019. 2, 3

[31] M. Hassan, V. Choutas, D. Tzionas, and M. J. Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2282–2292, 2019. 8

[32] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1849–1856, 2009. 3

[33] 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 *European Conf. on Computer Vision (ECCV)*, pages 187–203, 2018. 2, 3

[34] M. Hwangbo, J. Kim, and T. Kanade. Inertial-aided klt feature tracking for a moving camera. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1909–1916, 2009. 3, 5

[35] C. Jiang, S. Qi, Y. Zhu, S. Huang, J. Lin, L.-F. Yu, D. Terzopoulos, and S. Zhu. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *Intl. J. of Computer Vision*, 126(9):920–941, 2018. 2, 3

[36] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and F.-F. Li. Image retrieval using scene graphs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 3

[37] J. Johnson, B. Hariharan, L. van der Maaten, F.-F. Li, L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, 2017. 3

[38] D. Joho, M. Senk, and W. Burgard. Learning search heuristics for finding objects in structured environments. *Robotics and Autonomous Systems*, 59(5):319–328, 2011. 8

[39] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[40] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Intl. J. of Robotics Research*, 30(7):846–894, 2011. 8

[41] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Transactions on Cybernetics*, PP:1–13, 08 2019. doi: 10.1109/TCYB.2019.2931042. 2, 3, 8

[42] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. Panoptic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[43] L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an IMU. In *British Machine Vision Conf. (BMVC)*, pages 16.1–16.11, 2011. 5

[44] T. Kollar, S. Tellex, M. Walter, A. Huang, A. Bachrach, S. Hemachandra, E. Brunskill, A. Banerjee, D. Roy, S. Teller, and N. Roy. Generalized grounding graphs: A probabilistic framework for understanding grounded commands. *ArXiv Preprint: 1712.01097*, 11 2017. 8

[45] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. *arXiv e-prints*, art. arXiv:1909.12828, Sep 2019. 3

[46] N. Kolotouros, G. Pavlakos, and K. Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 5, 6, 7, 8

[47] J. Krause, J. Johnson, R. Krishna, and F.-F. Li. A hierarchical approach for generating descriptive image paragraphs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3337–3345, 2017. 3

[48] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL https://arxiv.org/abs/1602.07332. 2

[49] S. Krishna. *Introduction to Database and Knowledge-Base Systems*. World Scientific Publishing Co., Inc., 1992. ISBN 9810206194. 4

[50] B. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978. 2, 3

[51] B. Kuipers. The Spatial Semantic Hierarchy. *Artificial Intelligence*, 119:191–233, 2000. 2, 3

[52] D. T. Larsson, D. Maity, and P. Tsiotras. Q-Search trees: An information-theoretic approach towards hierarchical abstractions for agents with computational limitations. 2019. 8

[53] T. Larsson and T. Akenine-Möller. A dynamic bounding volume hierarchy for generalized collision detection. *Comput. Graph.*, 30(3):450–459, 2006. 8

[54] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[55] C. Li, H. Xiao, K. Tateno, F. Tombari, N. Navab, and G. D. Hager. Incremental scene understanding on dense SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 574–581, 2016. 3

[56] J. Li and R. Stevenson. Indoor layout estimation by 2d lidar and camera fusion. 2020. arXiv preprint arXiv:2001.05422. 3

[57] J. Li, A. Raventos, A. Bhargava, T. Tagawa, and A. Gaidon. Learning to fuse things and stuff. *ArXiv*, abs/1812.01192, 2018. 4

[58] P. Li, T. Qin, and S. Shen. Stereo vision-based semantic 3D object and ego-motion tracking for autonomous driving. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *European Conf. on Computer Vision (ECCV)*, pages 664–679, 2018. 3, 5

[59] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *Intl. Conf. on Computer Vision (ICCV)*, 2017. 3

[60] X. Liang, L. Lee, and E. Xing. Deep variation structured reinforcement learning for visual relationship and attribute detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4408–4417, 2017. 3

[61] K.-N. Lianos, J. L. Schönberger, M. Pollefeys, and T. Sattler. Vso: Visual semantic odometry. In *European Conf. on Computer Vision (ECCV)*, pages 246–263, 2018. 3

[62] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. 12 2013. doi: 10.1109/ICCV.2013. 179. 3

[63] C. Liu, J. Wu, and Y. Furukawa. FloorNet: A unified framework for floorplan reconstruction from 3D scans. In *European Conf. on Computer Vision (ECCV)*, pages 203–219, 2018. 3

[64] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 3, 5, 7

[65] C. Lu, R. Krishna, M. Bernstein, and F. Li. Visual relationship detection with language priors. In *European Conf. on Computer Vision (ECCV)*, pages 852–869, 2016. 3

[66] R. Lukierski, S. Leutenegger, and A. J. Davison. Room layout estimation from rapid omnidirectional exploration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6315–6322, 2017. 3

[67] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2916–2923, 2018. 5

[68] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017. 2, 3

[69] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger. Fusion++: Volumetric object-level SLAM. In *Intl. Conf. on 3D Vision (3DV)*, pages 32–41, 2018. 3

[70] A. Monszpart, P. Guerrero, D. Ceylan, E. Yumer, and N. J. Mitra. imapper: interaction-guided scene mapping from monocular videos. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 8

[71] C. Mura, O. Mattausch, A. J. Villanueva, E. Gobbetti, and R. Pajarola. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics*, 44:20–32, 2014. ISSN 0097-8493. 3

[72] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *arxiv preprint: 1903.01177*, 2019. 3

[73] R. Newcombe, D. Fox, and S. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352, 2015. 3

[74] L. Nicholson, M. Milford, and N. Sünderhauf. QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM. *IEEE Robotics and Automation Letters*, 4:1–8, 2018. 3

[75] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56:915–926, 2008. 3

[76] S. Ochmann, R. Vock, R. Wessel, M. Tamke, and R. Klein. Automatic generation of structural building descriptions from 3d point cloud scans. In *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 1–8, 2014. 3

[77] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1366–1373. IEEE, 2017. 5, 6

[78] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto. Sparse 3D topological graphs for micro-aerial vehicle planning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018. 4, 6

[79] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. *Intl. Conf. on 3D Vision (3DV)*, pages 484–494, 2018. 3

[80] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz. Semantic object maps for robotic housework - representation, acquisition and use. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4644–4651, 10 2012. ISBN 978-1-4673-1737-5. doi: 10.1109/IROS. 2012.6385603. 3

[81] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3d human pose and shape from a single color image. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 459–468, 2018. 3

[82] S. Pirk, V. Krs, K. Hu, S. D. Rajasekaran, H. Kang, Y. Yoshiyasu, B. Benes, and L. J. Guibas. Understanding and exploiting object interaction landscapes. *ACM Transactions on Graphics (TOG)*, 36(3):1–14, 2017. 8

[83] A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. 2012. IEEE Intl. Conf. on Robotics and Automation (ICRA). 3

[84] K. Qiu, T. Qin, W. Gao, and S. Shen. Tracking 3-D motion of dynamic objects using monocular visual-inertial sensing. *IEEE Trans. Robotics*, 35(4):799–816, 2019. ISSN 1941-0468. doi: 10.1109/TRO. 2019.2909085. 3, 5

[85] A. Ranganathan and F. Dellaert. Inference in the space of topological maps: An MCMC-based approach. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2004. 2, 3, 4

[86] E. Remolina and B. Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152(1):47–104, 2004. 2, 3, 4

[87] J. Rogers and H. I. Christensen. A conditional random field model for place and object classification. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1766–1772, 2012. 3

[88] A. Rosinol, M. Abate, Y. Chang, and L. Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. *arXiv preprint arXiv: 1910.02490*, 2019. 2, 3, 5

[89] A. Rosinol, T. Sattler, M. Pollefeys, and L. Carlone. Incremental Visual-Inertial 3D Mesh Generation with Structural Regularities. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2019. 5, 7

[90] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone. uHumans dataset. 2020. URL http://web.mit.edu/sparklab/datasets/uHumans. 2, 7

[91] R. Rosu, J. Quenzel, and S. Behnke. Semi-supervised semantic mapping through label propagation with semantic texture meshes. *Intl. J. of Computer Vision*, 06 2019. 3

[92] J.-R. Ruiz-Sarmiento, C. Galindo, and J. Gonzalez-Jimenez. Building

multiversal semantic maps for mobile robot operation. *Knowledge-Based Systems*, 119:257–272, 2017. 3

[93] M. Rünz and L. Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4471–4478. IEEE, 2017. 3

[94] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20. IEEE, 2018. 3

[95] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011. 6

[96] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2, 3

[97] D. Schleich, T. Klamt, and S. Behnke. Value iteration networks on multiple levels of abstraction. In *Robotics: Science and Systems (RSS)*, 2019. 8

[98] M. Shan, Q. Feng, and N. Atanasov. Object residual constrained visual-inertial odometry. In *technical report, https://moshanatucsd.github.io/orcvio_githubpage/*, 2019. 3

[99] V. Tan, I. Budvytis, and R. Cipolla. Indirect deep structured learning for 3D human body shape and pose prediction. In *British Machine Vision Conf. (BMVC)*, 2017. 3

[100] K. Tateno, F. Tombari, and N. Navab. Real-time and scalable incremental segmentation on dense slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4465–4472, 2015. 2, 3

[101] S. Thrun. Robotic mapping: a survey. In *Exploring artificial intelligence in the new millennium*, pages 1–35. Morgan Kaufmann, Inc., 2003. 3

[102] E. Turner and A. Zakhor. Floor plan generation and room labeling of indoor environments from laser range data. In *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 1–12, 2014. 3

[103] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart. Cognitive maps for mobile robots: An object based approach. In *Proceedings of the IROS Workshop From Sensors to Human Spatial Concepts (FS2HSC 2006)*, 2006. 2, 3

[104] J. Wald, K. Tateno, J. Sturm, N. Navab, and F. Tombari. Real-time fully incremental scene understanding on mobile platforms. *IEEE Robotics and Automation Letters*, 3(4):3402–3409, 2018. 3

[105] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *Intl. J. of Robotics Research*, 26(9):889–916, 2007. 3

[106] R. Wang and X. Qian. *OpenSceneGraph 3.0: Beginner's Guide*. Packt Publishing, 2010. ISBN 1849512825. 2

[107] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison. ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015. 3

[108] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger. MID-Fusion: Octree-based object-level multi-instance dynamic slam. pages 5231–5237, 2019. 3

[109] D. Xu, Y. Zhu, C. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Intl. Conf. on Computer Vision (ICCV)*, 2017. 3

[110] H. Yang and L. Carlone. In perfect shape: Certifiably optimal 3D shape reconstruction from 2D landmarks. *arXiv preprint arXiv: 1911.11924*, 2019. 3

[111] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *arXiv preprint arXiv:2001.07715*, 2020. 2, 6

[112] A. Zanfir, E. Marinoiu, and C. Sminchisescu. Monocular 3D pose and shape estimation of multiple people in natural scenes: The importance of multiple scene constraints. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2148–2157, 2018. 3

[113] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008. From Sensors to Human Spatial Concepts. 2, 3

[114] H. Zhang, Z. Kyaw, S.-F. Chang, and T. Chua. Visual translation embedding network for visual relation detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, page 5, 2017. 3

[115] Y. Zhang, M. Hassan, H. Neumann, M. J. Black, and S. Tang. Generating 3d people in scenes without people. *arXiv preprint arXiv:1912.02923*, 2019. 8

[116] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3119–3126, 2013. 2, 3

[117] K. Zheng and A. Pronobis. From pixels to buildings: End-to-end probabilistic deep networks for large-scale semantic mapping. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019. 3

[118] K. Zheng, A. Pronobis, and R. P. N. Rao. Learning Graph-Structured Sum-Product Networks for probabilistic semantic maps. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3

[119] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2344–2351, 2013. 5

[120] Y. Zhu, O. Groth, M. Bernstein, and F.-F. Li. Visual7w: Grounded question answering in images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4995–5004, 2016. 3