

Multi-Robot Collaborative Dense Scene Reconstruction

SIYAN DONG[†], Shandong University and AICFVE Beijing Film Academy

KAI XU^{‡*}, National University of Defense Technology and AICFVE Beijing Film Academy

QIANG ZHOU, Shandong University and AICFVE Beijing Film Academy

ANDREA TAGLIASACCHI, Google Inc., University of Waterloo, and University of Victoria

SHIQING XIN, Shandong University

MATTHIAS NIESSNER, Technical University of Munich

BAOQUAN CHEN^{*}, Peking University

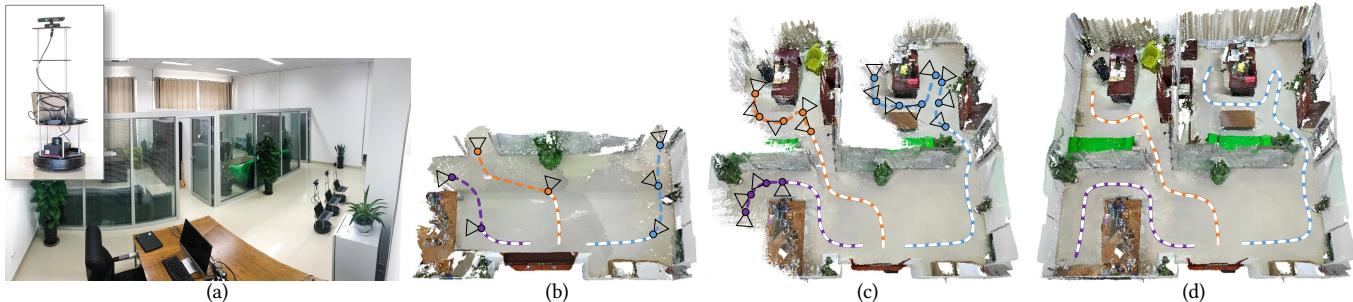


Fig. 1. We introduce an algorithm for multi-robot collaborative dense reconstruction of unknown indoor scenes (a). Given the partially scanned scene, we extract a set of task view points based on the uncertainty in the current reconstruction and assign them to the robots (b-c). The task assignment is formulated as an Optimal Mass Transport (OMT). For each robot, a smooth movement path is planned based on the tasks assigned to it. Our algorithm enables the robots to efficiently coordinate with each other, evenly distribute their scanning effort, and efficiently achieve a full coverage and high-quality reconstruction (d).

We present an autonomous scanning approach which allows multiple robots to perform collaborative scanning for dense 3D reconstruction of unknown indoor scenes. Our method plans scanning paths for several robots, allowing them to efficiently coordinate with each other such that the collective scanning coverage and reconstruction quality is maximized while the overall scanning effort is minimized. To this end, we define the problem as a dynamic task assignment and introduce a novel formulation based on Optimal Mass Transport (OMT). Given the currently scanned scene, a set of task views are extracted to cover scene regions which are either unknown or uncertain. These task views are assigned to the robots based on the OMT optimization. We then compute for each robot a smooth path over its assigned tasks by solving an approximate traveling salesman problem. In order to showcase our algorithm, we implement a multi-robot auto-scanning system. Since our method is computationally efficient, we can easily run it in real

time on commodity hardware, and combine it with online RGB-D reconstruction approaches. In our results, we show several real-world examples of large indoor environments; in addition, we build a benchmark with a series of carefully designed metrics for quantitatively evaluating multi-robot autoscanning. Overall, we are able to demonstrate high-quality scanning results with respect to reconstruction quality and scanning efficiency, which significantly outperforms existing multi-robot exploration systems.

CCS Concepts: • Computing methodologies → Shape analysis.

Additional Key Words and Phrases: Autonomous scene reconstruction, multi-robot, collaborative reconstruction, Optimal Mass Transport

ACM Reference Format:

Siyang Dong, Kai Xu, Qiang Zhou, Andrea Tagliasacchi, Shiqing Xin, Matthias Nießner, and Baoquan Chen. 2019. Multi-Robot Collaborative Dense Scene Reconstruction. *ACM Trans. Graph.* 38, 4, Article 84 (July 2019), 16 pages. <https://doi.org/10.1145/3306346.3322942>

[†]Siyang Dong and Kai Xu contribute equally. *Corresponding authors: Kai Xu (kevin.kai.xu@gmail.com) and Baoquan Chen (baoquan.chen@gmail.com).

Authors' addresses: Siyang Dong, Shandong University, AICFVE Beijing Film Academy; Kai Xu, National University of Defense Technology, AICFVE Beijing Film Academy; Qiang Zhou, Shandong University, AICFVE Beijing Film Academy; Andrea Tagliasacchi, Google Inc., University of Waterloo, University of Victoria; Shiqing Xin, Shandong University; Matthias Nießner, Technical University of Munich; Baoquan Chen, Peking University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Association for Computing Machinery.

0730-0301/2019/7-ART84 \$15.00

<https://doi.org/10.1145/3306346.3322942>

1 INTRODUCTION

Reconstructing and mapping indoor environments is critical to a large variety of applications, ranging from 3D content creation for augmented and virtual reality to localization for domestic robot navigation. On the hardware side, we have witnessed the emergence and proliferation of commodity range sensors (e.g., Microsoft Kinect, Intel RealSense, etc.) that capture depth data in real-time. On the software side, researchers have made incredible progress developing online RGB-D reconstruction methods [Izadi et al. 2011; Newcombe et al. 2011] that are able to reconstruct large environments [Chen et al. 2013; Nießner et al. 2013] along with robust camera tracking [Dai et al. 2017; Whelan et al. 2015; Zhang et al. 2014].

The real-time capabilities of these approaches allow human operators to reconstruct environments in an interactive fashion; e.g., by moving a hand-held sensor through a scene guided by the immediate visual feedback. Unfortunately, these frameworks are hardly user-friendly and involve significant experience. For instance, it is challenging for novice users to keep the camera trajectory smooth so not to break the pose estimation of camera tracking approaches. In addition, the live visualization is often insufficient to communicate scanning goals; for instance, where the operator should go next in order to obtain data from the regions that have not been scanned. As a result, the obtained reconstructions are often unsatisfying, and suffer from incomplete regions and misaligned geometry.

An alternative to human-operated scanning is autonomous scene scanning and reconstruction. Here, the key advantage is that a robot can leverage the existing reconstruction for trajectory planning. This way, a robot can provide high-quality reconstructions, with high levels of coverage and pose alignment quality. Although the joint planning of robot movement paths and camera trajectories, with the overall control of 3D reconstruction quality is a highly challenging problem, there have been significant progress in the recent years in the graphics and robotics communities [Charrow et al. 2015; Liu et al. 2018; Song et al. 2015; Xu et al. 2017].

Despite the promising results of these works, the major drawback is that they are inherently designed to operate only a single robot at a time. This greatly limits the scanning efficiency for large indoor environments. In this work, we address this shortcoming by introducing a *collaborative scanning approach* that allows multiple robots to jointly reconstruct a scene via combining and sharing their scanning resources. While multi-robot scanning has incredible potential for fast scanning, its problem statement is significantly different from the one in a single robot scenario. In this setting, we need a joint path planning for multiple robots to allow them to efficiently coordinate their scanning effort in the unknown scene, such that the collective scanning coverage and reconstruction quality is maximized, while the overall scanning effort is minimized.

Our key idea is to reduce multi-robot collaborative scanning to a *dynamic task assignment* problem. Given the partial scene scanned so far, we extract a set of task views looking into the regions which are either unknown or uncertain thereby needing more scan. These task views are assigned to the multiple robots so that the traveling cost and scanning effort of all robots is minimized. We formulate this objective based on Optimal Mass Transport (OMT), which is a classic approach to resource distribution problems [Rachev and Rüschorndorf 1998]. We then compute for each robot an optimal traverse path over its assigned tasks by solving a traveling salesman problem (TSP), where each task view is visited exactly once. Finally, we transform the TSP path into a smooth robot movement path, and compute a camera trajectory along the movement path. Our method alternates between path planning and scene geometry update, until the entire scene is scanned and reconstructed.

We develop a multi-robot scanning system, where the robots can efficiently coordinate with each other, evenly distribute their scanning efforts, and quickly achieve a complete and quality 3D reconstruction of an environment. Our system is tested in real-world scanning of six indoor scenes. For quantitative evaluation, we run our method on synthetic scenes with known ground-truth.

We propose a few evaluation metrics and carefully analyze our algorithm on a benchmark dataset. In our results, we show that our method achieves significantly higher reconstruction quality and scanning efficiency than the state-of-the-arts. In addition, we show that the algorithm is extremely robust with respect to the initial robot positions and the number of robots, which facilitate explorative scanning with no prior knowledge. To sum up, the contributions of this work are:

- *Formulation*: Optimal Mass Transport formulation tailored for multi-robot scanning of unknown indoor environments.
- *Optimization*: Efficient solution to multi-robot scan planning based on a divide-and-conquer scheme that interleaves task assignment and path optimization.
- *Metrics and benchmark*: Metrics for quantitatively evaluating both reconstruction quality and scanning efficiency, as well as a benchmark for this task, which will be made publicly available.
- *System*: End-to-end scanning system for collaborative autonomous scanning, which will be released on top of the ROS [2014].

2 RELATED WORK

RGB-D reconstruction. With the introduction of commodity depth cameras, we have seen significant advances in offline and online RGB-D reconstruction. KinectFusion [Izadi et al. 2011; Newcombe et al. 2011] was one of the first to realize a real-time volumetric fusion framework of Curless and Levoy [1996]. In order to handle larger environments, spatial hierarchies [Chen et al. 2013], and hashing schemes [Kahler et al. 2015; Nießner et al. 2013] have been proposed. At scale, these methods also required robust, global pose optimizations which are common in offline approaches [Choi et al. 2015]; however, fast GPU optimization techniques [Dai et al. 2017] or online re-localization methods [Whelan et al. 2015] allow for real-time global pose alignment. Our work extends this line of research to achieve real-time autonomous scanning by multiple robots.

Autonomous reconstruction with robots. Recent years have witnessed fast development on autonomous scanning for scene reconstruction with robots. Different from the traditional research on robot mapping, the main focus of these works is on the final *dense* surface reconstruction quality, rather than only creating a *sparse* map. The early methods started to look at single objects [Krajinin et al. 2011; Kriegel et al. 2012; Wu et al. 2014], which was then subsequently expanded to environments [Charrow et al. 2015; Xu et al. 2017]. In this work, we go one step further by introducing a real-time approach to *collaborative* reconstruction of unknown indoor scenes with multiple robots.

Next best view (NBV) planning. View selection and camera trajectory optimization are the core problems for robot-operated autonomous scanning [Chen et al. 2011]. Due to the explorative nature of auto-scanning, NBV selection is typically solved in a greedy manner. Many algorithms have been developed for active scanning of single objects [Krajinin et al. 2011; Wu et al. 2014] and scenes [Fan et al. 2016; Low and Lastra 2006]. Recently, an object-centric approach has been proposed for view planning in active scene scanning [Liu

et al. 2018]. Reinforcement learning has recently shown promising results on robot navigation (e.g., [Gupta et al. 2017]), but is still intractable for online, explorative scanning of large environments.

Multi-robot coordination and collaborative mapping. There is a large body of literature on multi-robot coordination and collaborative mapping [Atanasov et al. 2015; Thrun et al. 2005; Yan et al. 2013]. Of particular relevance to our work is multi-robot simultaneous localization and mapping (SLAM) [Forster et al. 2013; Mohanarajah et al. 2015; Schmuck and Chli 2017], and more closely multi-robot scene exploration and coverage [Bhattacharya et al. 2014; Faigl et al. 2012; Visser et al. 2013]. In this problem, a team of robots is driven to build joint maps of an environment, where the core problem is inter-robot co-localization and map merging. These works, however, mostly produce sparse map of the scene, rather than dense, high-quality reconstruction. The latter problem poses special challenges such as quality-driven view planning for multiple robots, which goes beyond the frontier-based exploration studied in those previous works.

Optimal Mass Transport. The optimal task assignment problem with capacity constraints has been extensively studied in engineering and economics for resource distribution [Haker et al. 2004; Rachev and Rüschendorf 1998]. It has also been employed to solve optimal coverage problems in sensor networks and robotics [Arslan and Koditschek 2016; Cortés 2010; Patel et al. 2014]. In our work, we adapt the optimal mass transport formulation for autonomous scanning with quality requirements, which leads to a new and interesting problem statement. For example, the *view-oriented distribution* of scanning effort is different from the location-based coverage setting in the traditional formulation. Moreover, our work considers a *dynamic scanning task assignment* problem with dynamically changing boundary conditions due to the progressively acquired scene geometry, which also leads to a novel problem setting.

3 METHOD

3.1 Problem statement and method overview

Given an indoor scene whose map is unknown *a priori*, we drive R robots, $\{\mathcal{R}_r\}_{r=1,\dots,R}$, to collaboratively explore and map the scene such that the scanning coverage and reconstruction quality is maximized while the scanning effort is minimized.

Since the map of the scene is unknown *a priori*, it is impossible to plan for the entire scanning process. A natural approach is “scan-and-plan” where the system alternates between robot scanning and path planning. The planning is based on the information gained by the scanning completed so far. Therefore, we need a temporal discretization and conduct planning for consecutive time intervals, which we refer to as *planning intervals*. The duration of each planning interval is also known as the *planning horizon* [Thrun et al. 2005] in the robotics field.

In what follows, we elaborate our problem setting by introducing the spatial/temporal discretization and the task definition of robot scanning and then provide an overview of our solution.

Spatial discretization. Our method works with indoor scenes with walls and furniture but not staircases (assuming planar ground).

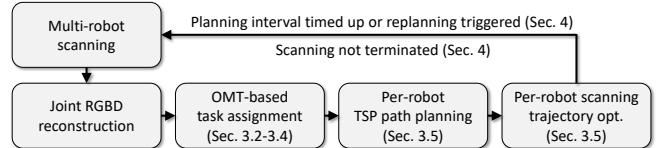


Fig. 2. An overview of our planning pipeline. In each planning interval, our method performs task view extraction (Sec. 3.2), OMT-based task assignment (Sec. 3.3 and 3.4), and per-robot path planning and camera trajectory optimization (Sec. 3.5). The cause of replanning and the termination criteria are elaborated in Sec. 4.

During explorative scanning, the scene is represented as a 3D volume of occupancy grid, based on OctoMap [Hornung et al. 2013]. To simplify the planning, the occupancy grid is projected to the floor plane forming a flat layout of obstacles. Under the assumptions of a planar world, each robot \mathcal{R}_r is parameterized by a state $(x_r, y_r, \theta_r) \in \mathbb{SE}(2)$ (see Figure 3(a)), where $(x_r, y_r) \in \mathbb{R}^2$ is the robot position in the 2D plane while $\theta \in [0, 2\pi]$ is camera orientation, which is independent from the robot’s direction of motion because the camera is mounted on the robot with a yaw degree of freedom.

Temporal discretization. The robot planning is conducted at consecutive planning intervals, $\{T_i\}_{i=0,\dots,+\infty}$. Specifically, at the beginning of each planning interval, we compute the scanning tasks and plan the motions for all robots for the time interval. During this time interval, the robots move and scan to accomplish the current tasks before entering the next planning interval. This process repeats until no more scanning task exists. Note that in this scheme the planning for each interval is based only on the scanned scene at the beginning of the interval and no update is performed for the planning during the execution of a plan. The duration of each planning interval is determined dynamically (see Sec. 4).

Task definition and planning goal. For each planning interval, we compute the scanning tasks based on scene geometry scanned so far. Intuitively, given the current scene geometry, scanning task can be defined as the scene regions that require more scans due to incomplete or low-quality coverage. To facilitate scan coverage, however, we convert region-based scanning tasks to view-oriented tasks. In particular, we sample a set of scanning views pointing to the task regions, which we name *task views*. In a planar world, a task view can be represented as $\mathcal{T}_k = (x_k, y_k, \theta_k) \in \mathbb{SE}(2)$. For each scanning interval, the set of all scanning tasks is represented by $\{\mathcal{T}_k\}_{k=1,\dots,K}$. Thus, our goal for the current planning interval is to drive the multiple robots to visit and take all task views in the most efficient way possible.

Multi-robot planning. To plan the scanning for multiple robots, we first construct a weighted task graph, by setting the task views as nodes and their pair-wise shortest path as edges. The path length is used as the edge weight. To minimize the scanning effort of all robots, a natural choice is to formulate the planning as a *multiple traveling salesman problem* (mTSP) which determines for each robot a tour such that the total traveling cost is minimized and that each task view is visited exactly once by one robot. Since mTSP is NP-hard [Bektas 2006] and there is no efficient exact solution, we propose a divide-and-conquer scheme which consists of two steps.

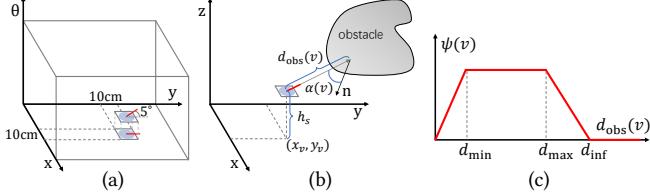


Fig. 3. (a): Discretization of 3D parametric view space (x, y, θ), with a spatial resolution of 10cm and angular resolution of 5°. (b): An illustration of scanning distance $d_{\text{obs}}(v)$ and scanning angle $\alpha(v)$ of a view v . The view is taken at the 3D position of (x_v, y_v, h_s) with h_s being the sensor height. n is the normal at the hitting point on the obstacle surface. (c): Function $\psi(v)$ defined over scanning distance $d_{\text{obs}}(v)$.

First, we solve a discrete optimal mass transport (OMT) for an optimal assignment of all task views to all robots. Then for each robot, we solve a vanilla *traveling salesman problem* (TSP) to determine an optimal visiting path for all tasks assigned to the robot. Although the per-robot TSP is still NP-hard, due to the reduced problem size, the problem can be solved much more efficiently to meet the requirement of online planning. We show through experiment that this OMT-based optimization outperforms the approximate solutions to mTSP in terms of scanning efficiency.

Figure 2 gives the pipeline of our method. Our method interleaves scanning task assignment and scanning path planning for consecutive planning intervals. In the following sections, we first introduce the extraction of task views (Sec. 3.2). We then present the formulation of our planning objective based on optimal mass transport (Sec. 3.3). Following that, we provide a discretized version of the objective as well as its solution (Sec. 3.4). Finally, we describe an algorithm to optimize the per-robot scanning trajectories (Sec. 3.5).

3.2 Scanning Task Extraction

Given the currently scanned scene geometry, we extract a set of scanning task views to be covered by the multiple robots. The task extraction take both coverage completeness and scanning quality into account. The former drives the robots to scan the unknown or uncertain regions. The latter is related to the limited resolution of a 3D scanner. For a 3D scanner, good scanning quality is obtained within a fixed range of scanning distance and scanning angle (between view direction and surface anti-normal). Regions scanned beyond these ranges will not receive a quality scanning coverage. For example, the valid range of scanning distance for a Kinect sensor is between 50cm and 3m. The best scanning angle is usually 0° and the quality decreases as the angle increases.

To facilitate the extraction of scanning task views, we compute two auxiliary maps in accordance to the two requirements above, an *uncertainty map* defined in the 3D occupancy volume and a *validness map* defined on the parametric space of camera view, i.e., $\mathbb{SE}(2)$.

Uncertainty map. Given the current occupancy volume, we measure the scanning uncertainty of each voxel as follows. First, an unknown voxel is assigned with a very high uncertainty. For a known voxel, there are two cases. If the known voxel is empty (unoccupied), its uncertainty is zero. Otherwise, the voxel represents the surface of an obstacle or object and we compute its uncertainty based on its mapping uncertainty in depth fusion. In particular, it is defined as the weights in the truncated signed distance field

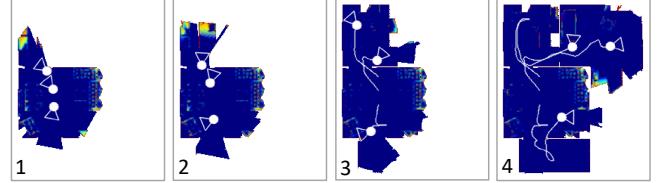


Fig. 4. Visualization of 2D uncertainty maps (red is high uncertainty and blue is low) with increasing scans.

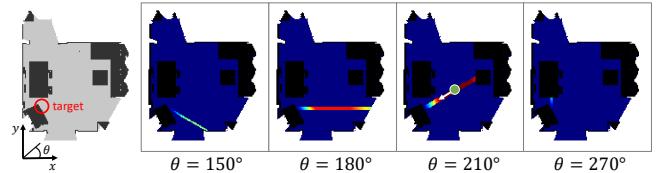


Fig. 5. Visualization of 2D validness maps (red is high validness and blue is low) for different turning angles ($\theta = 150^\circ, 180^\circ, 210^\circ$ and 270°) against a target obstacle (the red circle in the map to the left). The best scanning view against the target is obtained at the green dot with a turning angle of 210°.

(TSDF) [Curless and Levoy 1996]. TSDF weight is a simple and effective measure of scanning quality accounting for both scanning distance and angle. Figure 4 visualizes the uncertainty maps as the three-robot scanning proceeds.

Validness map. To ensure the extracted task views are within the valid scanning range against the obstacles in the scene, we define a validness map in the space of camera views. Since the camera view point v is defined in the 3D parametric space, i.e., $(x_v, y_v, \theta_v) \in \mathbb{SE}(2)$, the validness map is a 3D volumetric field (Figure 3(a)). The validness of a view point is 1 (the maximum value) if the view is within the valid distance range and has a 0° scanning angle against the nearest obstacle that is visible to the view point. The value decreases as the scanning distance deviates from the valid range or the angle from 0°. If a view points to an unknown voxel, we set its validness to 1, to encourage the robot to explore the unknown.

In practice, we discretize the parametric view space where the 2D position simply takes the 2D (floor) projection of the 3D occupancy grid and the turning angle is discretized at 5° intervals. Therefore, the validness map is defined in a $M \times N \times 72$ volume, with M and N being the horizontal resolution of the 3D occupancy grid. For each cell v in this volume, (x_v, y_v, θ_v) , we first find its closest obstacle visible along the view direction. This is achieved by casting a ray from the 3D point (x_v, y_v, h_s) along the direction of θ_v , where h_s is the height of the scanner mounted on the robot, and find the first intersection with an obstacle voxel in the 3D occupancy grid (Figure 3(b)). This can be performed efficiently in the 3D volume using the fast 3D DDA algorithm. If the ray does not intersect an obstacle voxel, we set the validness $\rho(v) = 1$. Otherwise, its validness is computed as $\rho(v) = \varphi(v)\psi(v)$, where $\varphi(v) = e^{-\alpha(v)^2/\sigma^2}$ measures the deviation of the scanning angle $\alpha(v)$ from 0°, with $\sigma = 0.6$. The function $\psi(v)$ measures the scanning distance deviation with respect to the valid scanning range of the sensor [d_{\min}, d_{\max}]; see

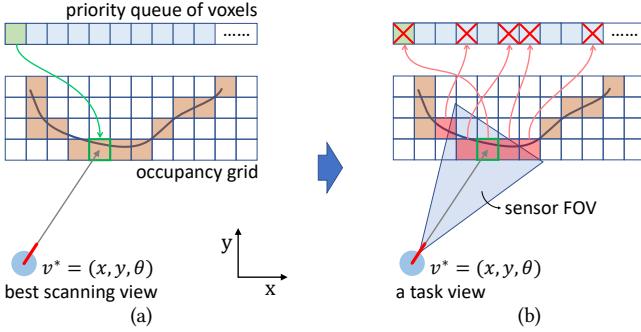


Fig. 6. Illustration of task view selection. (a): For the voxel in the head of priority queue, we select the best view, v^* , from which the voxel is visible and whose validness value is the highest. (b): Once v^* is selected, it is marked as a task view. The entries for all voxels covered by the view cone (FOV) of v^* , as well as the head, are removed from the priority queue.

Figure 3(c) for its definition. In the figure, $d_{\text{obs}}(v)$ is the distance from the position of view v to the closest visible obstacle. d_{inf} is set to 6m. To speed up runtime, we compute the validness value only for the view points located within known, empty regions. In Figure 5, we visualize the validness maps for a given position of target obstacle in four different turning angles.

Task view extraction. To extract task views, we first sort all voxels in the uncertainty map into a priority queue based on their uncertainty values. For efficiency reasons, we consider only those *frontier voxels* that lie in the interface between empty and unknown voxels. Starting from the head voxel in the priority queue, we select the best scanning view for it. To do this, we scan the validness map and extract all cells whose view ray passes through the voxel, meaning that the voxel is visible to that view, and whose location is not too close to an obstacle (distance to any obstacle voxel is larger than 1.5 times the radius of robot), to avoid collision. If such views exist, we select the one with the highest validness value as the best view for the voxel. This view is stored as a task view. Once a view is selected, all voxels which are within the field of view of the sensor placed at that view are removed from the priority queue. The head voxel is also removed from the priority queue. See Figure 6 for an illustration. The above process repeats until a prescribed maximum number κR task views are selected. κ is a scaling factor whose impact and choice will be elaborated in Section 5.4. Finally, we score the importance of each task view based on the reduction of mapping uncertainty using the method in [Krainin et al. 2011]. Please refer to Appendix (in the supplemental material) for details.

3.3 Planning Objective Formulation

Having computed a set of task views, our next step is to assign the robots to take the tasks so that the moving effort of all robots is minimized. We formulate this task assignment problem as an Optimal Mass Transport (OMT) problem. Given two distributions, an optimal transport plan transforms a source distribution into a target distribution, where optimality is defined according to a given metric. In our setting, the source is the *spatial distribution* of robots

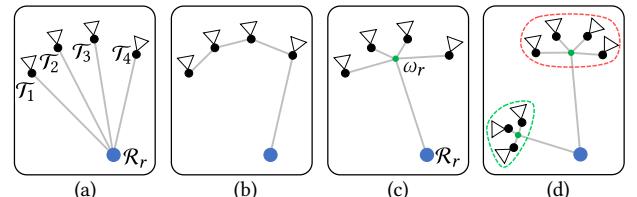


Fig. 7. Definition of OMT cost. The blue dot represents a robot and the black dots are task views assigned to the robot. The lines indicate distance cost. Since an efficient scanning path is a one-pass traverse of all tasks, TSP-based distance cost (b) is more natural than the distance summation in (a), while the former is costly to compute. We instead measure the distance from the robot to the centroid (the green dot in c) of all task points as the *distance cost*. The *compactness cost* is defined as the sum of task-to-centroid distances. In (d), the green cluster is more compact than the red one.

within the environments:

$$\mu_{\text{source}} = \sum_{r=1}^R \delta(\mathcal{R}_r) \quad (1)$$

where $\delta(\cdot)$ is the Dirac function. The target distribution describes the spatial distribution of the scanning tasks within the scene:

$$\mu_{\text{target}} = \sum_{k=1}^K \delta(\mathcal{T}_k) \quad (2)$$

Multi-robot task assignment objective. The optimal scan planning problem that assigns robots to scanning tasks can then be formulated as finding a mapping T that minimizes the following objective:

$$\arg \min_T \int_{x \in \mathbb{SE}(2)} \gamma(x, T(x)) \, d\mu_{\text{source}} \quad (3)$$

where $\gamma(a, b)$ measures the cost if a is mapped to b , and $T(x) \sim \mu_{\text{target}}$ is the image of $x \sim \mu_{\text{source}}$. The optimal mass transport is the minimum of the cost of T . The discretization of the objective with the cost definition will be discussed in the next section.

3.4 Discretization of Planning Objective

In this section, we propose a discretized planning objective through defining the cost of the OMT map T and provide a solution to the optimization. We mainly consider the following two aspects.

Movement cost. Optimally, the movement cost should be defined as a one-pass traversing distance visiting all the tasks. This requires solving a TSP which would make the optimization computationally intractable. Therefore, we opt for an approximate cost which is the shortest distance from the robot to the centroid of the set of assigned tasks. See Figure 7(c) for illustration.

Robot capacity. Due to the fixed moving speed of a robot, each robot has a limited scanning capacity defined as the maximum number of tasks it can accomplish during a planning interval. The time required for finishing one task depends not only on robot moving speed but also on the distance of the task; the latter is related to both room size and robot count. Therefore, an accurate estimation of robot capacity is difficult. In our work, we set the capacity of a robot R_r as $C_r = \frac{K^{\text{total}}}{R} \cdot \frac{v_r}{v_{\max}}$, where K^{total} is the total number of tasks, v_r the moving speed of the robot and v_{\max} the

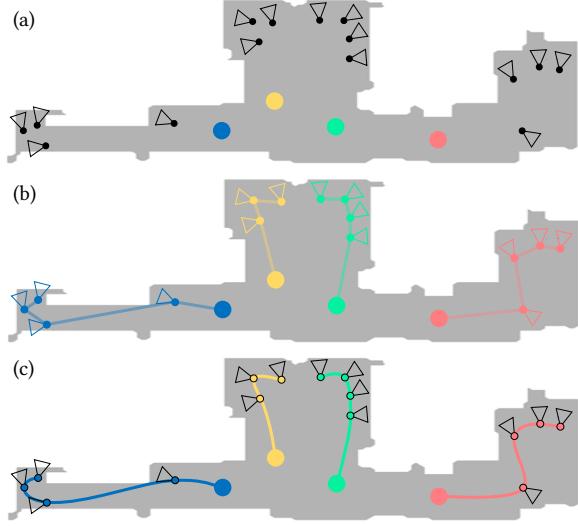


Fig. 8. (a): Given the currently scanned scene (shape in gray), a set of task views (black camera icon) are extracted. (b): By solving OMT, the tasks are assigned to the four robots (see color correspondence). For each robot, the visiting order of the tasks is determined by solving a local TSP. (c): The traversing paths are smoothed to obtain the final movement paths.

maximum speed of all robots. In the case where all robots move equally fast, it leads to a load-balancing assignment of tasks.

Discretized objective. To meet the requirements above, we discretize Eq. (3) in the following objective function:

$$\min_T \sum_{r=1}^R \underbrace{\sum_{\mathcal{T}_k \in \Omega_r} \gamma(\mathcal{T}_k, \omega_r)}_{\text{compactness}} + \underbrace{\sum_{r=1}^R \gamma(\mathcal{R}_r, \omega_r)}_{\text{distance}} + \underbrace{\sum_{r=1}^R (|\Omega_r| - C_r)^2}_{\text{capacity}} \quad (4)$$

where Ω_r is the set of assigned tasks for \mathcal{R}_r . ω_r is the centroid of Ω_r . $\gamma(\cdot, \cdot)$ is the shortest distance constrained on the currently scanned scene geometry. The first term measures the *compactness* of the tasks assigned to one robot by the sum of inter-task distances (Figure 7(c)). This is similar to the compactness term defined in power diagrams [Aurenhammer 1987]. Minimizing this term ensures that the tasks assigned to one robot are not spatially scattered. The *distance* term minimizes the moving cost of robot in finishing its tasks. The *capacity* term tries to match the number of task assigned to a robot to its capacity. The optimization can be solved very efficiently using the Lloyd's algorithm [Du et al. 2006] (see Section 4 for more details). Figure 8(b) shows an example of task assignment for four robots. In Figure 9, we demonstrate the progressive planning and scanning process on a real scene by three robots.

3.5 TSP-based Path Planning

The OMT-based planning partitions the scanning tasks into groups so that each robot can focus on its own set of tasks. In the next step, we compute an optimal traverse path for each robot based on its assigned tasks. This can be reduced to a TSP problem, where each task is visited by the robot exactly once, resulting in a connected TSP

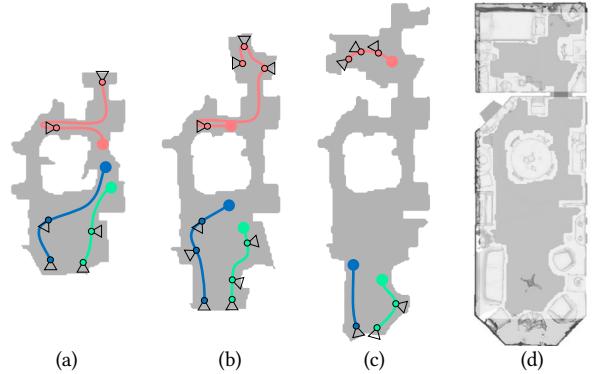


Fig. 9. Multi-robot paths planned for different planning intervals, during the progressive scanning process of a real-world scene from (a) to (c). The explored / known regions are shaded in grey color. (d) shows the final reconstruction in top view.

path. To maximize scanning quality, we optimize the TSP path into a smooth movement path (Figure 8(c)), and compute a continuous camera trajectory along the movement path.

TSP path. To compute a TSP path for a robot \mathcal{R}_r , we build a weighted graph where the nodes are the positions of a task \mathcal{T}_t or the robot \mathcal{R}_r while directed edges connect every two nodes with the shortest path between them. Our problem is slightly different from the original TSP, since the robot does not need to return to its starting point. To that end, we solve a modified version of TSP:

- 1 Set the robot position as the starting node and the farthest task to the robot as the ending one;
- 2 Remove all outgoing edges from the ending node except the one that directs to the starting node;
- 3 Solve TSP over the current graph;
- 4 Remove the path from the ending node to the starting node;

To solve the TSP at interactive rates, we adopt the approximate algorithm proposed in [Christofides 1976].

Path optimization. We optimize the TSP path to meet the following requirements: First, the movement path of any robot should be as smooth as possible. Smooth paths not only benefit robot control but facilitate frame-to-frame registration for both robot tracking and scene reconstruction. Second, since the camera orientation is independent on robot rotation in our setting, we also need to optimize the camera pose to achieve a smooth scan.

To optimize the movement path for robot \mathcal{R}_r , we first uniformly sample a sequence of points along the path. Let $P_r = (p_1, \dots, p_N)$ denote the ordered list of points encompassing both the sampled points and the set of task view points assigned to \mathcal{R}_r . T_r is set of indices of the task points in P_r . The following energy is minimized:

$$\arg \min_{P_r} \sum_{i=1}^{N-1} \frac{2}{\eta(p_i) + \eta(p_{i+1})} \|p_i - p_{i+1}\|^2 + \lambda \sum_{t \in T_r} \|p_t - p_t^0\|^2, \quad (5)$$

where η is the *outside* distance function computed for the 2D projection of the currently scanned scene. The function value is 0 for

points inside the scene surface. p_t^0 is the original position of p_t . The first term is devised to smooth the path curve while keeping the curve away from scene obstacles. The second term regularizes the optimization by fixing the task view points. We set $\lambda = 10$ in throughout all our experiments.

After obtaining a smooth motion path, we achieve a smooth camera view transition along the path through view interpolation based on cubic Hermite splines, similar to [Xu et al. 2017].

4 IMPLEMENTATION

We use Turtlebots as our robots, each of which is equipped with an Asus Xtion, as well as an RPLIDAR A2M8 360° LiDAR scanner (see Figure 1(a)). The Xtion captures RGBD data for online RGBD reconstruction. The LiDAR scanner is used for robot tracking, based on the Google’s Cartographer [Hess et al. 2016] which provides robust and real-time LiDAR-based SLAM. The two sensors are calibrated. Each robot carries an on-board laptop (Intel(R) Core(TM) i5-8250U CPU (1.60GHz×8), 8GB RAM) used for running LiDAR-based SLAM Cartographer. The RGBD data from the Xtion and the camera poses (robot locations) from Cartographer are both transmitted to a control machine (Intel(R) Core(TM) i7-6700K CPU (4.0GHz×4), 16GB RAM, GTX1080 GPU) that executes the online scene reconstruction and path planning. The GPU is used only for online scene reconstruction. The communication channel is a 54Mbps Wifi connection.

The robots are co-located at the initialization stage. To calibrate their relative poses before scanning, each robot takes a 360° scan and a joint localization is performed based on the AMCL [2013], an adaptive Monte Carlo localization algorithm implemented in ROS. No more communication or calibration between robots is needed during scanning. Based on the collected scans, the control machine runs online occupancy map updating with OctoMap [Hornung et al. 2013], online reconstruction using VoxelHashing [Nießner et al. 2013], as well as our online path planning algorithm. Based on the planned paths, it sends out the movement command and camera poses to the individual robots. Obstacle avoidance can be achieved inherently by our method when computing the shortest distances and optimizing the final paths.

Parameter setting. The moving speed of our robots is 0.3m/s. The spatial resolution of occupancy grid is 10cm in all dimensions and the angular resolution for the discretization of view space is 5°. The total number of task views in each planning interval is κR (R is the robot count), with $\kappa = 6$. The planning horizon for each planning interval is 15s. After planning, the robots execute the motions until 15s are elapsed, or any one of the robots has finished its tasks, whichever happens first. More planning details are discussed below.

Optimization and planning details. Our basic solution to the optimization in Equation (4) is to iteratively alternate between K-means clustering and a relaxation for constraint satisfaction [Balzer et al. 2009]. K-means requires a prescribed number (K) of clusters. To find the K automatically, we start with each robot position as a clustering seed, thereby setting $K = R$ initially. During the optimization, the K-means clustering optimizes the compactness and distance terms. The relaxation step further optimizes the compactness and satisfies

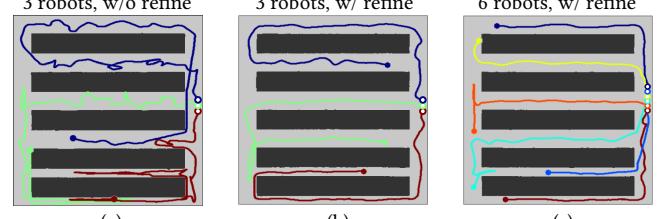


Fig. 10. Demonstrating the effect of compactness-based clustering refinement in task assignment via simulating multi-robot scanning in synthetic scenes with (b and c) and without (a) refinement. Three robots are used in (a) and (b) and six used in (c). The start positions for all robots are around the right side of the scene. Note how our method with clustering refinement dispatches the robots more reasonably with smoother planned paths.

the capacity constraint. Specifically, we check the compactness and capacity of each cluster. For any cluster, if there is a task which is too far away from the cluster center (the shortest distance is larger than a threshold $\delta = 1\text{m}$; see Section 5.4), we remove it from the cluster and set it as a new clustering seed for the next iteration, thus increasing the K . Similarly, if some cluster breaks the capacity constraint, we again isolate its most off-center task to form a new clustering seed. In the next iteration, a refined clustering will emerge, which better optimizes the terms in (4) with a more proper K . The optimization usually takes 3 ~ 8 iterations to converge. When the optimization converges, those clusters which were seeded by a robot location are assigned to the corresponding robot, while the other clusters are left unassigned.

Figure 10 demonstrates the effect of compactness-based clustering refinement with simulated multi-robot scanning in a synthetic scene. Without clustering refinement, the robots tend to move back and forth due to the frequent switching between spatially scattered tasks. With clustering refinement, the robots are dispatched more reasonably and the scanning paths look much smoother.

Energy consumption vs. time cost. In our main implementation, after the optimization converges, those clusters formed by non-robot seeds are *not* assigned to any robot. When there are very few tasks left, some robot-seeded clusters may not contain any task. This means that not every robot has to be assigned with tasks. Such planning strategy tends to minimize the total energy consumption. See the example demonstrated in Figure 13(top) where some robots are out of work due to the limited amount of tasks available. Another option is to make sure every robot is assigned with tasks. To achieve that, one can simply perform a K-means with $K = R$ at all times, thus removing the need of clustering refinement. This strategy leads to a more balanced workload and less scanning time, but incurs much higher overall energy consumption. In the supplemental material, we provide a quantitative comparison of the two strategies.

Termination criteria. Our system terminates when there is no more task view with an importance score (information gain or the reduction of mapping uncertainty [Krainin et al. 2011]) higher than 1.5 that are accessible to the robots.

Complexity. For each planning interval, the extraction of task views and the computation of OMT scale as $O(\kappa R)$, which requires roughly 0.5s and 0.6s, respectively, for $R = 10$ and $\kappa = 6$. The time is averaged across all scenes. The update of the occupancy map in

OctoMap takes 2s. The estimation of approximate TSP paths scales as $O(K_r^3)$ per robot (K_r is per-robot task count), which adds an additional 50 ~ 200ms (again averaged across all scenes). The computations of TSP paths for all robots can be parallelized. In summary, the duration of each planning is about 3 seconds. Such latency can be easily hidden in the switching between two consecutive planning intervals, through setting some overlap between them.

5 RESULTS AND EVALUATION

We evaluate our method by simulating our scanning procedure on a collection of synthetic 3D scene models (Sec. 5.1), and quantitatively measuring and comparing the scanning performance (Sec. 5.3–5.6) based on a series of evaluation metrics (Sec. 5.2). This *evaluation benchmark (dataset+metrics)* will be released to enable the evaluation of autonomous dense reconstruction in future works. Finally, we demonstrate abundant visual results on both synthetic and real-world scanning as a qualitative evaluation (Sec. 5.7).

5.1 Synthetic scanning for quantitative evaluation

Benchmark dataset. We construct a benchmark of synthetic scanning for quantitative evaluation of multi-robot collaborative scanning. The benchmark dataset contains 80 scenes selected from the SunCG [Song et al. 2017] and Matterport3D [Chang et al. 2017] datasets; please see Figure 18 for a few representative benchmark scenes and refer to the supplemental material for an overview of the benchmark dataset. Some preprocessing on the original scene models, such as making the ground perfectly planar and adjacent rooms strictly connected, was conducted to make them usable for simulation. Two representative benchmark scenes, SunCG#1 and Matterport3D#1 (see Figure 17), will frequently be used as the test scenes in most of the quantitative plots in this section. We design a series of evaluation metrics for quantifying both reconstruction quality and scanning efficiency in Section 5.2. Next, we describe our synthetic scanning framework built on top of ROS.

Synthetic scanning framework. Given an input 3D scene model represented by a 3D triangular mesh, we employ the Gazebo [Koenig and Howard 2004] framework running on ROS to produce at frame t a set of depth maps \mathcal{D}_t^r , one for each of the $r = 1 \dots R$ robots, as well as the camera view matrices \mathcal{T}_t^r . The framework allows us to control the trajectory of the robot by specifying the instantaneous (linear, and angular) velocity of the robot, and simulates any delay due to motion, change of trajectory, or scanning. While Gazebo typically runs at 20Hz, OctoMap is only able to fuse at 1Hz. Hence, we trigger a fusion operation when the robot has moved more than a given distance (20cm in our experiments), or rotated more than a given angle (60°). Once a mesh is loaded in the system, the robots are placed by specifying their six DoFs. To emulate a real system, the robots positions are initialized close to each other, so that there is sufficient overlap in their reconstructed areas – this enables the fusion of reconstructed geometry across different robots. In synthetic scanning, we assume the locations of robots are known throughout the simulation, therefore neither initial joint localization nor online SLAM is needed.

5.2 Evaluation metrics

We design a series of evaluation metrics to quantitatively evaluate both reconstruction quality and scanning efficiency.

Reconstruction quality. Given our reconstruction \mathcal{S} , and the corresponding ground truth \mathcal{G} , both of which are represented by triangular mesh, the two can be considered geometrically identical if an only if their (symmetric) *Hausdorff* distance $\varphi_{\mathcal{S} \leftrightarrow \mathcal{G}}$ is zero [Cignoni et al. 1998]:

$$\varphi_{\mathcal{S} \leftrightarrow \mathcal{G}} = \max(\varphi_{\mathcal{S} \rightarrow \mathcal{G}}, \varphi_{\mathcal{G} \rightarrow \mathcal{S}}), \quad (6)$$

$$\varphi_{\mathcal{S} \rightarrow \mathcal{G}} = \max_{\mathbf{s} \in \mathcal{S}} \left[\min_{\mathbf{g} \in \mathcal{G}} \phi(\mathbf{s}, \mathbf{g}) \right], \quad (7)$$

$$\varphi_{\mathcal{G} \rightarrow \mathcal{S}} = \max_{\mathbf{g} \in \mathcal{G}} \left[\min_{\mathbf{s} \in \mathcal{S}} \phi(\mathbf{s}, \mathbf{g}) \right]. \quad (8)$$

We can perform a few simple manipulations to the expression above to obtain two different metrics, one accounting for reconstruction *accuracy*, the other for reconstruction *completeness*. First of all, following [Tkach et al. 2016], we replace the max in (7) and (8) with integrals over the corresponding surfaces, and in turn, with a slight abuse of notation, replace these integrals with (area weighted) sums over the vertices $\mathbf{s} \in \mathcal{S}$ and $\mathbf{g} \in \mathcal{G}$. Then, for $\varphi_{\mathcal{S} \rightarrow \mathcal{G}}$ we select ϕ to be the ℓ_2 Euclidean norm (i.e. a-la Robust ICP), while for $\varphi_{\mathcal{G} \rightarrow \mathcal{S}}$ we select ϕ to be the ℓ_2^0 norm (i.e. a-la RANSAC), with ℓ_2^0 being $\|\mathbf{x}\|_2^0$:

$$\varphi_{\mathcal{S} \rightarrow \mathcal{G}} = \frac{1}{\sum A(\mathbf{s})} \sum_{\mathbf{s} \in \mathcal{S}} A(\mathbf{s}) \min_{\mathbf{g} \in \mathcal{G}} \|\mathbf{s} - \mathbf{g}\|_2, \quad (\text{RMS error}) \quad (9)$$

$$\varphi_{\mathcal{G} \rightarrow \mathcal{S}} = \frac{100}{\sum A(\mathbf{g})} \sum_{\mathbf{g} \in \mathcal{G}} A(\mathbf{g}) \min_{\mathbf{s} \in \mathcal{S}} \|\mathbf{s} - \mathbf{g}\|_2^0, \quad (\text{Completeness \%}) \quad (10)$$

where $A(\cdot)$ measures the area of a vertex. Note that ℓ_2^0 counts the number (L0 norm) of correspondences with small residual, analogous to counting inliers in RANSAC. Specifically, $\|\mathbf{x}\|_2^0$ differentiates between inliers and outliers via the parameter ε :

$$\|\mathbf{x}\|_2^0 = \begin{cases} 0 & \text{if } \|\mathbf{x}\|_2 < \varepsilon \\ 1 & \text{otherwise} \end{cases} \quad (11)$$

Our accuracy metric (9) is similar to the MSE (mean square error) metric with SE(3) alignment. The difference lies in the use of a robust L1 kernel to down-weight potential outliers (common in raw acquired data) via the use of M-estimators [Bouaziz et al. 2013]. We employ the robust L1 norm since the effect of outliers can be mitigated by the fusion-based reconstruction [Niessner et al. 2013].

As the synthetic depth maps \mathcal{D} are *noise free*. To properly evaluate accuracy (9), we first add synthetic noise to the depth maps based on the noise model proposed in [Handa et al. 2014]. We then discard depth values in $\{\mathcal{D}_t^r\}$ outside the 0.2m ~ 3m range, as we know the error in this range is at acceptable levels [Fanello et al. 2016, Fig. 4]. This way, the metrics in (9) and (10) become sufficient to evaluate the overall reconstruction quality. In our experiments, we select $\varepsilon = 15\text{cm}$, which is three times the standard deviation of depth noise for objects scanned in the range of 0.2m ~ 3m.

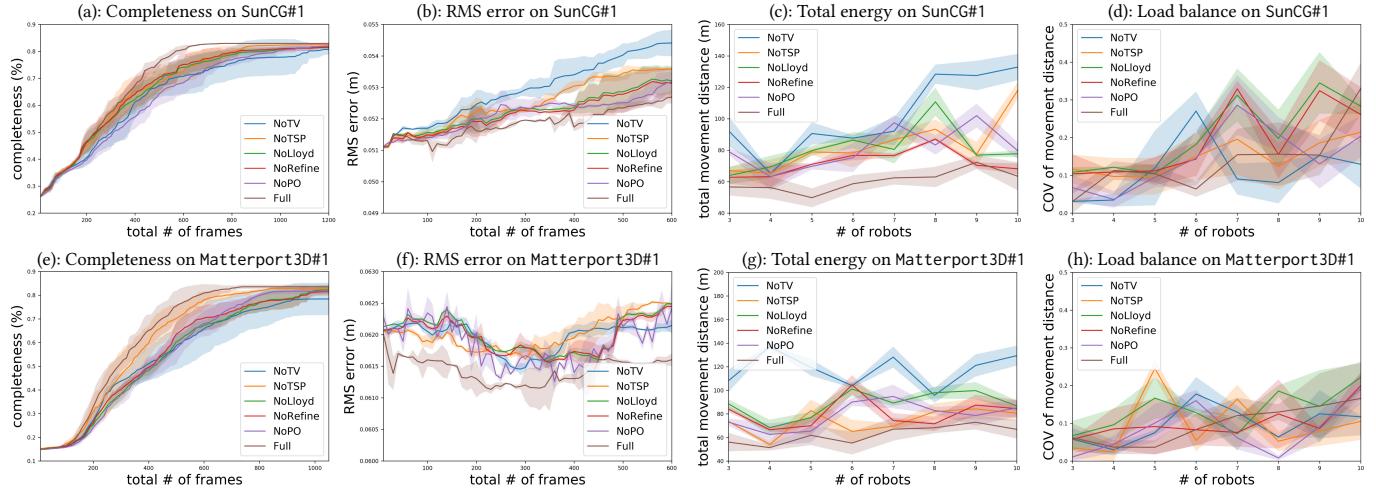


Fig. 11. Ablation studies on two synthetic test scenes SunCG#1 (top row) and Matterport3D#1 (bottom row).

Table 1. Comparing reconstruction quality (completeness and RMS error) and scanning efficiency (total energy is normalized with scene area) between our full method and four baselines over the 80 benchmark scenes.

	Reconstruction quality		Scanning efficiency	
	Completeness	RMS error	Total energy	Load balance
NoTV	81.83%	0.0576	0.934	0.105
NoTSP	82.69%	0.0574	0.673	0.121
NoLloyd	82.12%	0.0573	0.721	0.171
NoRefine	81.61%	0.0573	0.649	0.146
NoPO	81.90%	0.0572	0.686	0.115
Full	83.62%	0.0560	0.525	0.105

Scanning efficiency. To measure scanning efficiency, we design three metrics covering different aspects:

Scanning redundancy measured by the total number of frames scanned by all robots. Redundancy leads to increased total number of frames, which can be caused either by too few or too many robots (see Figure 13). When there are too few robots, a robot could retrace its steps frequently (e.g. entering and then exiting a tunnel with a dead end). We refer to this kind of redundancy as *intra-robot redundancy*. On the other hand, too many robots would, inevitably, cause *inter-robot redundancy*.

Total energy consumption measured by the total movement distance of all robots.

Load balance measured by the coefficient of variation (COV) of movement distance over all robots. COV, defined as the ratio of the standard deviation to the mean, is a standard metric for quantifying load balance [Huang et al. 2005]. The lower the COV is, the more balanced the task assignment is.

Running time of the system for finishing the scanning.

5.3 Ablation studies

To justify the design choices of the various components in our algorithm, we compare our full method (Full) against the following baseline variants, over the evaluation metrics proposed above:

No task view (NoTV) Robot tasks are defined as sample points of frontiers, as is done in previous works [Faigl et al. 2012; Visser et al. 2013], instead of task views used in our method.

No TSP (NoTSP) After OMT-based task assignment, instead of following a TSP path, each robot finishes its own tasks in a greedy manner, i.e., always choosing the closest task.

No Lloyd (NoLloyd) Instead of optimizing (4) based on Lloyd’s algorithm, we employ the Centroidal Power Diagrams (CPD) [Xin et al. 2016], a standard solution to OMT problems.

No clustering refinement (NoRefine) During the optimization, no clustering refinement is performed.

No path optimization (NoPO) The optimization of both movement paths and camera trajectories (Sec. 3.5) is disabled.

Aggregated performance statistics. To avoid bias in our evaluations, in our quantitative plots we report aggregated statistics over *random* initialization of robot placements in the scene (see Figure 17 for scanning paths corresponding to five random initializations of our OMT planning algorithm on SunCG#1 and Matterport3D#1). We execute each algorithm 10 times, and at each time step (counted by number of scanning frames) and for each number of robots, we compute *mean* and *standard deviation* of the metrics over the observations. We visualize the mean as a solid line, and the corresponding standard deviation as a transparent overlay of the same color.

Results on reconstruction quality. The left two columns of Figure 11 show the plots of reconstruction quality (completeness and accuracy) over increasing number of scanning frames, when 10 robots are used for synthetic scanning of SunCG#1 (top row) and Matterport3D#1 (bottom row). The plots in (a) and (e) show that our full method achieves consistently the fastest increase of scan completeness compared to all baselines. The final completeness of all methods are nearly the same, except NoTV, which verifies the importance of task view extraction. Notice that none of the methods gets 100% completeness because the fixed height of scanners (with only the yaw DoF) limits the range of scan coverage. The plots in

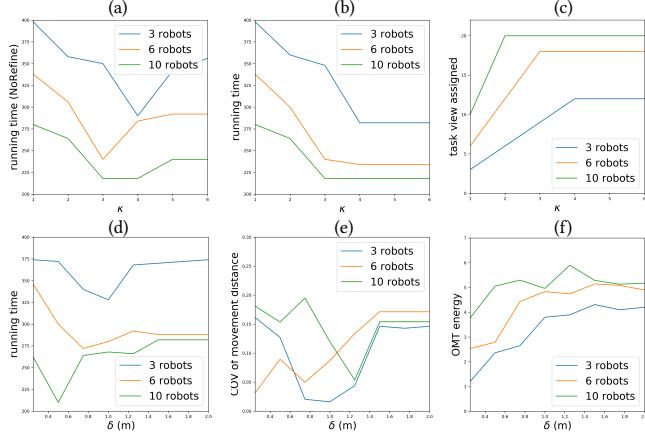


Fig. 12. Studies on two key parameters κ (top) and δ (bottom), through synthetic scanning of SunCG#1. For κ , running time (in second) is plotted for both the baseline NoRefine (a) and our full method (b). For δ , COV of movement distance and OMT energy are plotted.

(b) and (f) demonstrate that our full method obtains the lowest reconstruction error due to the reasonably planned movement paths and carefully optimized camera trajectories for all robots. Note that the increase-decrease patterns of the accuracy curves depend on the scene layout and the initialization of robots.

Results on scanning efficiency. The right two columns of Figure 11 are the plots of scanning efficiency (total energy consumption and load balance) over different numbers of robots. The results again justify the design choices of our algorithm. Note that all the methods attain similar load balance since they all utilize OMT-based task assignment. The contrast of load balance is more significant in the comparison against non-OMT-based approaches in Section 5.6.

Table 1 reports the results of reconstruction quality and scanning efficiency averaged over all the 80 scenes from our benchmark dataset. For all methods, reconstruction quality is measured at the end of the scanning, while scanning efficiency is reported for 10 running robots. To make the total energy consumption metric comparable across different scenes, we normalize it through dividing by scene area.

5.4 Studies on key parameters

We studied the effect of two key parameters through experiments, which helped us find the best parameter settings for our algorithm. Once the settings are found, they are fixed throughout the experiments, including both synthetic and real-world tests.

Scaling factor κ . This parameter controls the maximum number of task views (κR) extracted in Section 3.2. To study the effect of this parameter, we run synthetic scanning on SunCG#1 with fixed initialization. We first show in Figure 12(a) the system running time over varying values of κ , under the optimization setting of no clustering refinement (NoRefine). The plots show that too small κ leads to frequent and local planning. In particular, when $\kappa = 1$, our algorithm degenerates to a greedy planning. In general, larger κ results in more global planning thus more efficient scanning. When κ grows further, however, the scanning efficiency degrades since a large number of tasks, which are likely spatially scattered, would

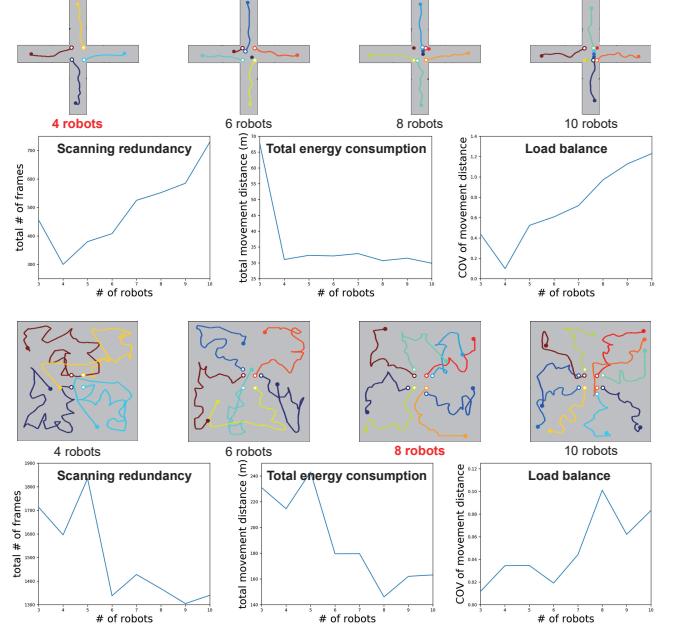


Fig. 13. Study on the number of robots with two hand-crafted scenes. For each scene, we show the robot scanning paths (starting points are depicted with circles and terminal points with solid dots) for different number of robots, all obtained by our algorithm. We also plot scanning redundancy, total energy consumption and load balance over increasing number of robots. The optimal numbers of robots for both cases are highlighted in red color.

make the OMT optimization highly non-convex thus hard to reach a good local optimal. When clustering refinement is added in our algorithm (Figure 12(b)), increasing κ does not result in efficiency degradation because the refinement ensures that the tasks assigned to each robot always form a compact cluster, thus greatly benefiting the OMT optimization. Meanwhile, satisfying the compactness constraint would lead to many spatially scattered tasks left unassigned. This can be observed in Figure 12(c), where the number of tasks assigned levels out as κ increases.

In conclusion, by adopting clustering refinement in the OMT optimization, κ can be set to an arbitrarily large number, gaining a more global planning without sacrificing optimality. Note, however, large κ causes many task views, which incurs higher computational cost in the optimization. We choose $\kappa = 6$ in all our experiments.

Distance threshold δ . This parameter is used to assess the compactness of a task cluster during clustering refinement (Section 4). The plots in Figure 12(d) and (e) show that the best scanning efficiency and load balance are both achieved when $\delta \approx 1\text{m}$. A too small δ leads to overly fine-grained clusters, which makes the planning less global, while a too large δ virtually invalidates the cluster refinement. The latter is also reflected in the plots of objective function (4) in Figure 12(f), where the energy values become stable as δ growing. The choice of δ is insensitive to the number of robots.

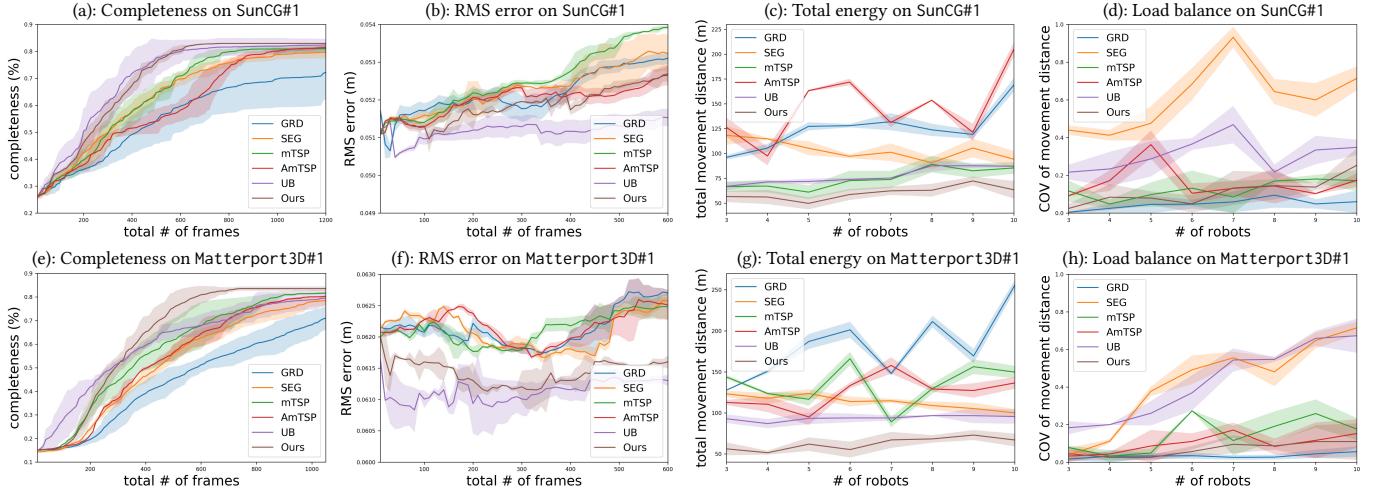


Fig. 14. Comparisons on two synthetic test scenes SunCG#1 (top row) and Matterport3D#1 (bottom row).

Table 2. Comparing reconstruction quality and scanning efficiency between our method and five alternatives over the 80 benchmark scenes.

	Reconstruction quality		Scanning efficiency	
	Completeness	RMS error	Total energy	Load balance
GRD	82.48%	0.0572	1.293	0.040
SEG	80.23%	0.0571	0.924	0.519
mTSP	81.66%	0.0573	0.876	0.136
AmTSP	81.65%	0.0571	1.165	0.130
Ours	83.62%	0.0566	0.525	0.105

5.5 Study on number of robots

To have a clear understanding on how the number of robots relates to scanning redundancy, we run our method for synthetic scanning of two simple, hand-crafted scenes including a narrow cross and an empty square (see Figure 13). In each scene, a group of robots starts around the center of the scene and performs collaborated scanning guided by our OMT-based planning. We show the robot scanning paths and plot scanning redundancy, total energy consumption and the load balance over varying number of robots.

For the narrow cross scene, four robot exhibits the lowest redundancy and the best load balance (0.1). This is because each of the four robots takes charge of a branch of the cross, with a balanced task assignment and without any return trip. With more robots deployed, the extra robots do not receive any task but stay still, according to our planning algorithm. This incurs imbalanced task load (up to 1.2) but at the same time saves the total energy consumption.

In the empty square, the lowest redundancy and energy cost are achieved with around 8 ~ 9 robots, where the robots are distributed roughly towards the four sides and the four corners. The load balance is consistently good (0.01 ~ 0.1) for varying numbers of robots. This is because all robots can be dispatched to scan into different directions in such a spacious scene as the empty square.

In general, however, the choice of a proper number of robots depends on the specific scene layout, which is practically intractable when the scene is unknown *a priori*. Our algorithm, however, can reduce the scanning redundancy and save energy consumption for

the whole system as much as possible with a reasonable, OMT-based task assignment.

5.6 Comparisons

We compare our method to five alternatives, sorted roughly in increasing algorithmic sophistication. The second (SEG) and the fourth (AmTSP) methods represent the state-of-the-arts for multi-robot collaborative mapping. In order to gauge the upper limit of the performance of our method, we design an “upper-bound” variant of our method, where the scene is assumed to be known *a priori*.

Greedy (GRD) We first assign tasks to robots in a greedy manner alike [Visser et al. 2013]; then for each robot, we choose the closest among all tasks assigned to it as the active target.

Segmentation-based (SEG) In each planning interval, a Voronoi segmentation is performed over the current frontiers; each robot then takes the closest segmented region and scans the frontiers within it in a greedy manner [Bhattacharya et al. 2014].

Multiple TSP (mTSP) In each planning interval, we first estimate task views, then build a weighted task graph with their pair-wise shortest distances as edge weights, and solve an mTSP using the genetic algorithm to obtain the movement path for each robot. All other steps are similar to our method.

Approximate mTSP (AmTSP) In each planning interval, this method constructs a weighted graph of frontier samples, with the shortest distance between samples as edge weights, and solves an mTSP based on interleaving Kmeans clustering and TSP path estimation [Faigl et al. 2012]. The optimization process is similar to ours but without clustering refinement. Besides, their frontier-based objective formulation is completely different from our view-based one.

Upper bound (UB) An “upper bound” implementation of our method where the scene is assumed known *a priori*, so that all task views can be extracted at the beginning.

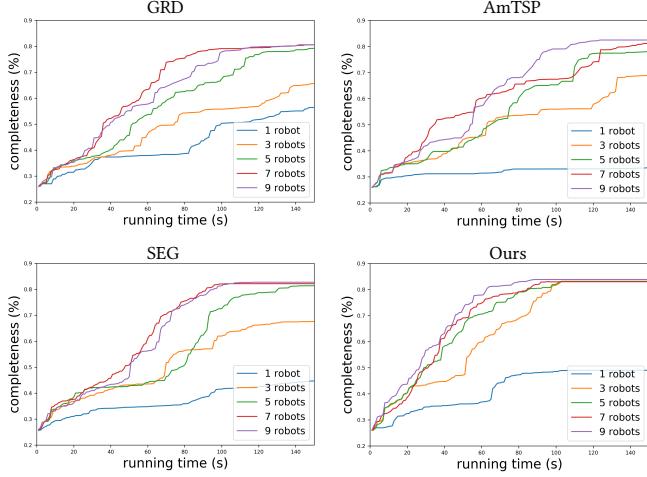


Fig. 15. We compare our method with three alternatives (GRD, AmTSP and SEG) in terms of their diminishing return as the number of robots is increased on the SunCG#1 scene. Notice how for our OMT algorithm the curves are approaching the top-left part of the graphs the fastest.

Similar to the ablation studies in Figure 11, the plots in Figure 14 are also obtained by running all algorithms for 10 random initializations of robot placement to avoid bias. Both mean and standard deviation of the metrics are visualized in the plots. Table 2 compares reconstruction quality and scanning efficiency between our method and five alternatives over the 80 benchmark scenes, similar to Table 1.

Results on reconstruction quality. The left two columns of Figure 14 evaluate the completeness and accuracy over increasing number of scanning frames, through running again 10 robots on the two test scenes. Our algorithm achieves the fastest increasing rate and highest final completeness, as well as the lowest reconstruction error, among all the alternatives except the upper-bound one. The increasing rate of AmTSP is relatively low because the method assigns *all* frontiers to the robots in each planning. This causes the tasks assigned to some robots to be *incompact* for which a good TSP path is difficult to find. The reconstruction error of the SEG and AmTSP methods is relatively high since these methods are not view-based planning and thus the scanning trajectories are unaware of valid scanning distance and angle of the sensor.

Results on scanning efficiency. The right two columns of Figure 14 focus on scanning efficiency, demonstrating that our method is the most efficient among all methods except for the upper-bound (UB). Our method achieves lower total energy consumption (c and g) than UB since the latter involves more task views when the scene is known *a priori* (UB achieves higher completeness at the same time). The incompact task assignment of AmTSP again leads to high total energy consumption, due to the suboptimal TSP solution. About the load balance (d and h), the greedy planning (GRD) is the most balanced of all since all robots are simultaneously dispatched to the best view tasks causing their movement distances to be almost the same. The total movement distance, however, is very high. The planning by SEG is the most unbalanced since the Voronoi-based segmentation is essentially a distance-based assignment, which may

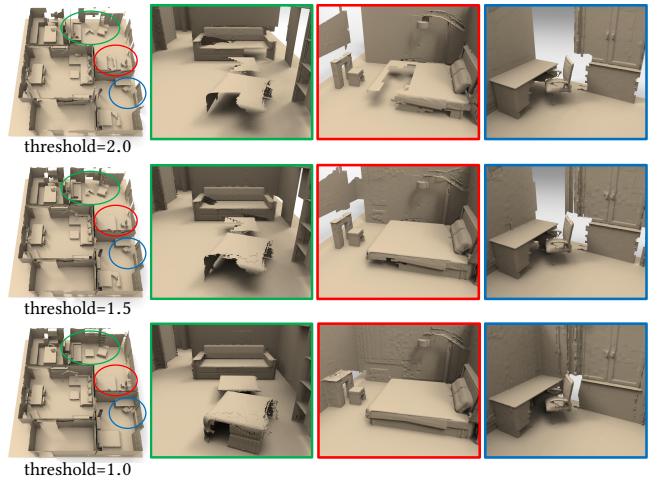


Fig. 16. Different scanning completeness due to different thresholds of terminating information gain (see termination criteria in Section 4). Lower thresholds lead to more complete reconstruction while needing more scanning passes. When threshold is 1.0 (bottom row), for example, the scene was scanned for 2.5 passes averagely.

lead to greater imbalance than task-clustering-based approaches. See Figure 18 for visual comparisons of path planning.

Congestion and diminishing returns. We also compare the scanning performance of four algorithms (GRD, AmTSP, SEG and ours) as we increase the number of robots in Figure 15, to see how well they are able to dispatch many robots. The test is performed on SunCG#1, for which 5 robot is the saturation point of scanning efficiency. Our method achieves consistently faster scanning coverage than the alternatives for every number of robots. More importantly, the performance of our method keeps improving as the number of robots increasing, even when other methods have already reached saturation (diminishing return). In particular, the alternative methods saturate at 7 robot, since the curves for 9 robot become lower than those for 7. Our method, on the other hand, is able to dispatch 9 robots better and achieves a faster scanning and a higher completeness. By inspecting the sequences we also noticed that, when no high-level coordinated strategy for exploration is provided (e.g. GRD), the interference between different robots can actually be detrimental to scanning effectiveness.

5.7 Qualitative results

Figure 16 demonstrates an example that higher scanning completeness can be obtained by decreasing the threshold of terminating information gain (see termination criteria in Section 4). When a relatively small threshold (1.0 for example) is set, most regions of the scene were scanned for more than twice by the robots.

Figure 17 shows the scanning paths of our OMT algorithm over SunCG#1 and Matterport3D#1, with different initializations (starting positions of robots). The scanning paths for different robots are shown with distinct colors. Note how our method achieves visually consistent exploration behavior with different initializations, due to the minimization of scanning redundancy. In contrast, we also show

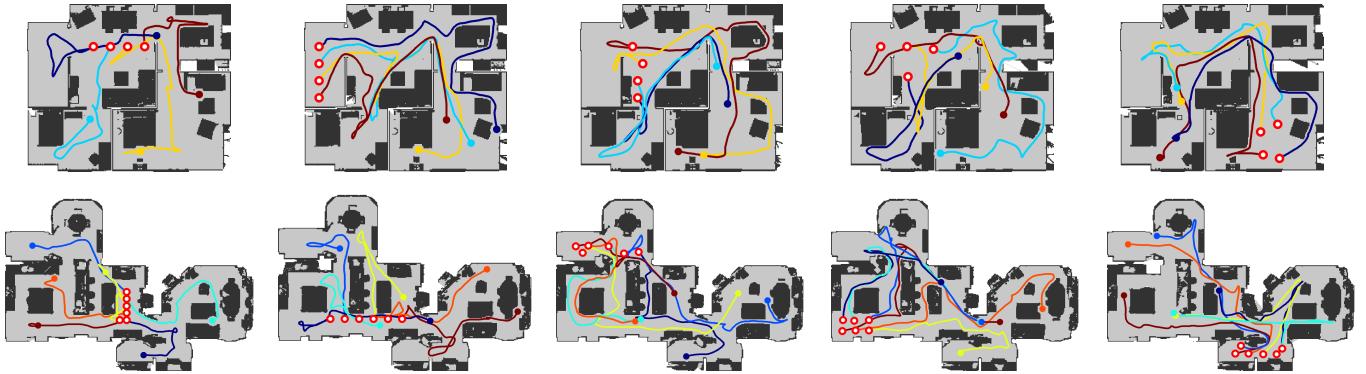


Fig. 17. Scanning paths planned by our OMT-based algorithm on the SunCG#1 (top) and Matterport3D#1 (bottom) scene, for four and six robots, respectively, with different initializations. The starting points of robots are depicted with red circles. Notice the exploration behavior of our solution is visually consistent.

in the supplemental material the scanning paths obtained by SEG and AmTSP, with the corresponding configurations (robot count and initializations). For these alternative methods, the exploration patterns are less consistent for different initializations.

Figure 18 shows a gallery of planned paths for synthetic scanning of five indoor scenes from our benchmark dataset. This figure also serves as a qualitative comparison of scanning paths planned by our method and four alternatives (GRD, SEG, mTSP and AmTSP). Each row of the figure corresponds to one scene and each column to a specific algorithm. For each scene, all algorithms run with the same number of robots deployed with the same initialization. The uniform distribution and smooth spanning of our paths explain how our method attains higher reconstruction quality and scanning efficiency. Moreover, our method achieves the same level of completeness with much smaller total traverse distance. Please refer to the supplemental material for more visual results.

Figure 1 and Figure 19 shows real-world scanning performed with our multi-robot scanning system, over six indoor scenes including an office (Office), a sitting room (Sitting_room), a classroom (Classroom), a meeting room (Meeting_room), a dormitory (Dorm), and a computer lab (Lab). For each scene in Figure 19, we show a bird-eye view picture of the real scene, the scanning paths planned and the final reconstruction result by our method. As a comparison, we also show the reconstruction results obtained by manual scanning with a hand-held depth camera. In general, it is difficult for a human scanner to plan the camera trajectories which are both globally efficient and locally smooth. In addition, the live visualization of online reconstruction (e.g. VoxelHashing) is too local, and is insufficient to communicate scanning progress on which regions are scanned and which are not. With our multi-robot scanning system, more complete and higher quality reconstruction can be achieved with much less scanning time, due to the reasonably planned smooth movement paths and camera trajectories. Please watch the accompanying video for live demonstration of the scanning processes.

Timings and statistics. Table 3 reports the timings and statistics for scanning the two synthetic scenes (SunCG#1 and Matterport3D#1) and the six real-world scenes (Office, Sitting_room, Classroom,

Table 3. Timing and statistics of both synthetic (row 2 and 3) and real scenes (row 4 ~ 9). For each scene, we report scene area, number of robots (#R), number of planning intervals (#I), planning time of each planning interval (PT), duration of each planning interval (IT), as well as total time to finish the scanning (TT), and total movement distance of all robots (TD).

Scene	Area	#R	#I	PT	IT	TT	TD
SunCG#1	110 m ²	3	16	0.9 sec	22 sec	6 min	48 m
Matterport3D#1	125 m ²	6	17	2.2 sec	26 sec	8 min	55 m
Office	60 m ²	3	10	0.8 sec	17 sec	3 min	32 m
Sitting_room	85 m ²	4	9	1.1 sec	25 sec	4 min	21 m
Classroom	120 m ²	5	5	1.2 sec	40 sec	5 min	41 m
Meeting_room	80 m ²	3	4	0.8 sec	46 sec	4 min	18 m
Dorm	35 m ²	2	4	0.5 sec	40 sec	3 min	17 m
Lab	300 m ²	6	40	2.3 sec	6 sec	5 min	156 m

Meeting_room, Dorm and Lab). All time was measured on the control machine described in Section 4. The computational time for each planning (planning time) is roughly proportional to scene complexity (layout of walls and furniture), while the duration of planning interval is inversely proportional to scene complexity. The latter is because when the scene has a complex layout, the tasks tend to be spatially scattered so the task clusters are highly fine-grained which each requires a shorter time to finish.

6 DISCUSSION AND CONCLUSIONS

We have presented a simple and robust algorithm for multi-robot collaborative scanning of unknown indoor scenes. The method attains the following key features. (i) It adopts view-based planning rather than (frontier) location-based, which is better tailored for scanning quality control. (ii) It achieves fast planning with a divide-and-conquer scheme which interleaves task assignment and per-robot path optimization. (iii) It realizes a flexible scanning setting where robot motion and camera orientation are independent.

Limitations and failure cases. Our method / system has the following limitations:

- **Task view extraction.** Our task view extraction is essentially a greedy solution, which may result in locally optimal views. One possible mitigation is to consider voxel groups instead

- of single voxels during the priority-based selection, which may lead to a more global solution possibly with raised computational cost.
- *Approximate OMT cost.* Our distance cost in OMT objective is defined as the distance from the robot to the centroid of its tasks. When the tasks are spatially scattered, such a cost will not be a good measure of robot movement effort. This has been alleviated by the compactness term in the objective as well as the clustering refinement strategy.
 - *View orientation smoothness.* The orientation smoothness of task views are not considered in OMT-based task assignment and TSP-based path optimization, but only in the last step for camera trajectory optimization.
 - *Smooth transition between planning intervals.* This could have been realized simply by joining the paths of two consecutive intervals with a B-spline curve. When many robots are deployed, however, the duration of planning intervals is short. A better solution is to stabilize the robot motion when the replanning is triggered frequently, using techniques from control theory.
 - *Non-distributed solution.* Our current method does not support a distributed system so a centralized control is required.
 - *Congestion control.* There is no explicit congestion control in our method. In the supplemental material, we show a failure case caused by movement congestion.

Future work. First, although the divide-and-conquer scheme leads to an efficient planning, a more interesting solution would be solving the OMT task assignment directly in the parametric view space, thereby skipping the view selection step. Second, we plan to investigate the incorporation of orientation-based constraints in the OMT formulation. Third, it would be especially interesting to study a decentralized solution of OMT-based task assignment. Lastly, an interesting future direction would be incorporating prior knowledge about scene layout learned offline or online into our OMT-based planning. Integrating geometry-based and learning-based planning could exploit their complementary advantages, leading to a more globally optimal path planning.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported in part by 973 Program (2015CB352501) and NSFC (61572507, 61532003, 61622212). The authors are grateful to Dorobot Inc., and to Honghua Li and Hongda Jiang, for their help in various stages of this project.

REFERENCES

- AMCL. 2013. The ROS Package of AMCL. <http://wiki.ros.org/amcl>.
- Omur Arslan and Daniel E Koditschek. 2016. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *Proc. ICRA*. IEEE, 4259–4266.
- Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. 2015. Decentralized active information acquisition: Theory and application to multi-robot SLAM. In *Proc. ICRA*. IEEE, 4775–4782.
- Aurenhammer. 1987. Power diagrams: properties, algorithms and applications. *Siam Journal on Computing* (1987).
- Michael Balzer, Thomas Schlämer, and Oliver Deussen. 2009. *Capacity-constrained point distributions: a variant of Lloyd's method*. Vol. 28. ACM.
- T. Bektas. 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* 34, 209–219 (2006).
- Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. 2014. Multi-robot coverage and exploration on Riemannian manifolds with boundaries. *The International Journal of Robotics Research* 33, 1 (2014), 113–137.
- Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2013. Sparse iterative closest point. *Computer Graphics Forum (SGP)* (2013), 113–123.
- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)* (2017).
- Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. 2015. Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Proceedings of Robotics: Science and Systems*.
- Jiawen Chen, Dennis Bautembach, and Shahram Izadi. 2013. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 113.
- Shengyong Chen, Youfu Li, and Ngai Ming Kwok. 2011. Active vision in robotic systems: A survey of recent developments. *International Journal of Robotics Research* (2011).
- Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. 2015. Robust reconstruction of indoor scenes. In *Proc. CVPR* 5556–5565.
- Nicos Christofides. 1976. Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem. (1976).
- Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. 1998. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* (1998).
- Jorge Cortés. 2010. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Trans. Automat. Control* 55, 3 (2010), 749–754.
- Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proc. of SIGGRAPH*. 303–312.
- Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Reintegration. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 24.
- Qiang Du, Maria Emelianenko, and Lili Ju. 2006. Convergence of the Lloyd Algorithm for Computing Centroidal Voronoi Tessellations. *SIAM J. Numer. Anal.* 44, 1 (2006), 102–119.
- Jan Faigl, Miroslav Kulich, and Libor Přeučil. 2012. Goal assignment using distance cost in multi-robot exploration. In *Proc. IROS*. IEEE, 3741–3746.
- Xinyi Fan, Lingguang Zhang, Benedict Brown, and Szymon Rusinkiewicz. 2016. Automated View and Path Planning for Scalable Multi-Object 3D Scanning. *ACM Trans. on Graph. (SIGGRAPH Asia)* 35, 6 (2016), 239.
- Sean Ryan Fanello, Christoph Rhemann, Vladimir Tankovich, Adarsh Kowdle, Sergio Orts Escolano, David Kim, and Shahram Izadi. 2016. Hyperdepth: Learning depth from structured light without matching. In *Proc. CVPR*.
- Christian Forster, Simon Lynen, Laurent Kneip, and Davide Scaramuzza. 2013. Collaborative monocular slam with multiple micro aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3962–3970.
- Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. 2017. Cognitive mapping and planning for visual navigation. *arXiv preprint arXiv:1702.03920* 3 (2017).
- Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. 2004. Optimal mass transport for registration and warping. *International Journal of computer vision* 60, 3 (2004), 225–240.
- Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. 2014. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *Proc. ICRA*. IEEE, 1524–1531.
- Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. 2016. Real-time loop closure in 2D LIDAR SLAM. In *Proc. ICRA*. IEEE, 1271–1278.
- Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots* 34, 3 (2013), 189–206.
- Chengdu Huang, Gang Zhou, Tarek F Abdelzaher, Sang Hyuk Son, and John A Stankovic. 2005. Load balancing in bounded-latency content distribution. In *Proc. of RTSS*. IEEE, 12–pp.
- Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *UIST*. 559–568.
- O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S Torr, and D. W. Murray. 2015. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Trans. Vis. & Computer Graphics (ISMAR)* 22, 11 (2015).
- N Koenig and A Howard. 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *International Conference on Intelligent Robots and Systems*. 2149–2154 vol.3.
- M. Krainin, B. Curless, and D. Fox. 2011. Autonomous Generation of Complete 3D Object Models Using Next Best View Manipulation Planning. In *Proc. ICRA*.

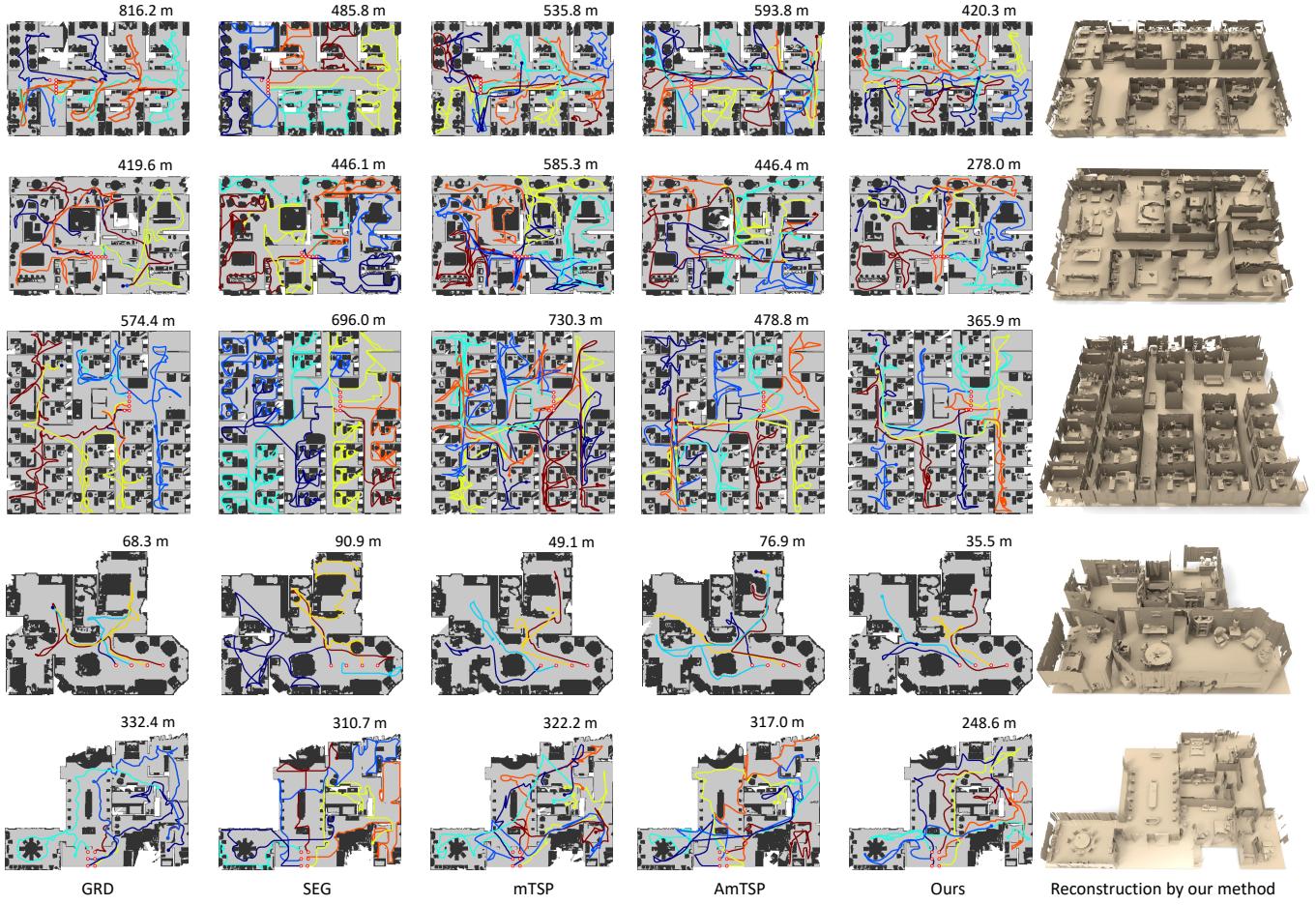


Fig. 18. A visual comparison of path planning for synthetic scene scanning over 5 benchmark scenes (3 SUNCG scenes and 2 Matterport3D scenes). For each scene (row), our result is compared to those by four alternatives including GRD, SEG, mTSP and AmTSP. The total traverse distance of all robots is given to the top-right corner of each planning result, for achieving a completeness level of 83%. The final reconstruction results by our method are shown in the last column.

- Simon Kriegel, Christian Rink, Tim Bodenmüller, Alexander Narr, Michael Suppa, and Gerd Hirzinger. 2012. Next-best-scan planning for autonomous 3D modeling. In *Proc. IROS*. 2850–2856.
- Ligang Liu, Xi Xia, Han Sun, Qi Shen, Juzhan Xu, Bin Chen, Hui Huang, and Kai Xu. 2018. Object-Aware Guidance for Autonomous Scene Reconstruction. *ACM Trans. on Graph. (SIGGRAPH)* 37, 4 (2018).
- Kok-Lim Low and Anselmo Lastra. 2006. An adaptive hierarchical next-best-view algorithm for 3d reconstruction of indoor scenes. In *Proceedings of 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*.
- Gajamohan Mohanarajah, Vladyslav Usenko, Mayank Singh, Raffaello D’Andrea, and Markus Waibel. 2015. Cloud-based collaborative 3D mapping in real-time with low-cost robots. *IEEE Transactions on Automation Science and Engineering* 12, 2 (2015), 423–431.
- Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. IEEE Int. Symp. on Mixed and Augmented Reality*. 127–136.
- M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. 2013. Real-time 3D Reconstruction at Scale using Voxel Hashing. *ACM Trans. on Graph. (SIGGRAPH Asia)* 32, 6 (2013), 169.
- Rushabh Patel, Paolo Frasca, and Francesco Bullo. 2014. Centroidal area-constrained partitioning for robotic networks. *Journal of Dynamic Systems, Measurement, and Control* 136, 3 (2014), 031024.
- Svetlozar T Rachev and Ludger Rüschendorf. 1998. *Mass Transportation Problems: Volume I: Theory*. Vol. 1. Springer Science & Business Media.
- ROS. 2014. ROS Wiki. <http://wiki.ros.org/>.
- Patrik Schmuck and Margarita Chli. 2017. Multi-UAV collaborative monocular SLAM. In *Proc. ICRA*. IEEE, 3863–3870.
- Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic Scene Completion from a Single Depth Image. (2017).
- S. Song, L. Zhang, and J. Xiao. 2015. Robot In a Room: Toward Perfect Object Recognition in Closed Environments. *arXiv preprint arXiv:1507.02703* (2015).
- Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. *Probabilistic robotics*. MIT press.
- Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-Meshes for Real-Time Hand Modeling and Tracking. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)* (2016).
- A Visser, J De Hoog, A Jimenezgonzalez, and J. R Martinezde Dios. 2013. Discussion of Multi-Robot Exploration in Communication-Limited Environments. In *2013 ICRA Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration*.
- Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. 2015. ElasticFusion: Dense SLAM without a pose graph. In *Proc. Robotics: Science and Systems*.
- Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoguan Chen. 2014. Quality-driven poisson-guided autoscanning. *ACM Trans. on Graph. (SIGGRAPH Asia)* 33, 6 (2014), 203.
- Shi-Qing Xin, Bruno Lévy, Zhonggui Chen, Lei Chu, Yaohui Yu, Changhe Tu, and Wenping Wang. 2016. Centroidal Power Diagrams with Capacity Constraints: Computation, Applications, and Extension. *ACM Trans. on Graph.* 35, 6 (2016), 244:1–244:12.

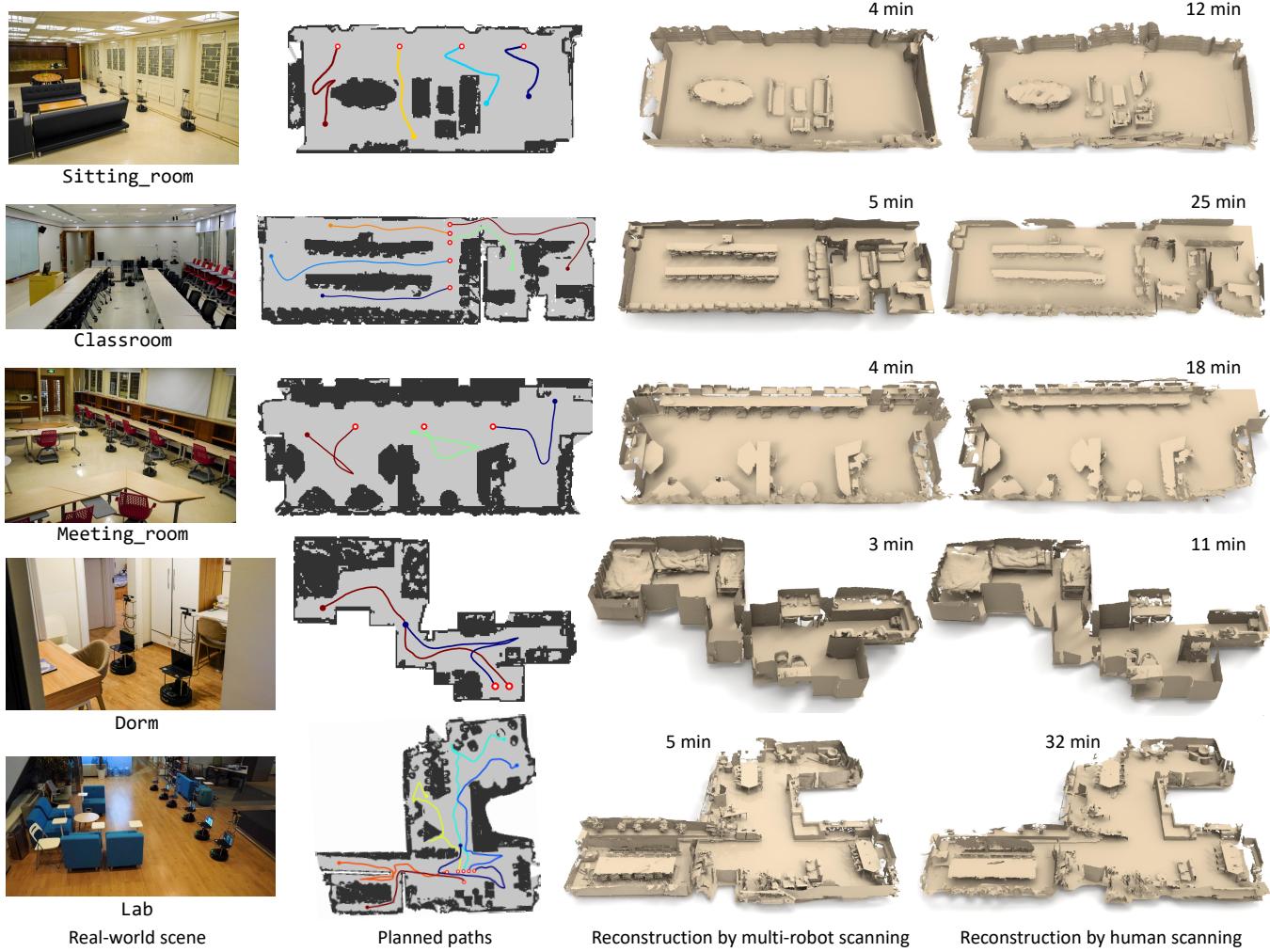


Fig. 19. A gallery of real-world scene scanning examples. In each row, we show from left to right a bird-eye view picture of the indoor scene, the online planned multi-robot scanning paths, the reconstruction result obtained by our multi-robot system, and the reconstruction by human scanning. Scanning time is provided for both autonomous and manual scanning. Our system achieves more complete reconstruction with much less scanning time.

Kai Xu, Lintao Zheng, Zihao Yan, Guohang Yan, Eugene Zhang, Matthias Nießner, Oliver Deussen, Daniel Cohen-Or, and Hui Huang. 2017. Autonomous Reconstruction of Unknown Indoor Scenes Guided by Time-varying Tensor Fields. *ACM Transactions on Graphics* 2017 (TOG) (2017).

Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. 2013. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems* 10, 12 (2013), 399.

Yizhong Zhang, Weiwei Xu, Yiyi Tong, and Kun Zhou. 2014. Online Structure Analysis for Real-time Indoor Scene Reconstruction. *ACM Trans. on Graph.* 34, 5 (2014), 159.