Optimizing Fiducial Marker Placement for Improved Visual Localization

Citation: Q. Huang, J. DeGol, V. Fragoso, S. N. Sinha, and J. J. Leonard, "Optimizing Fiducial Marker Placement for Improved Visual Localization", IEEE Robotics and Automation Letters (RA-L), 2023.

As Published: 10.1109/lra.2023.3260700

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: https://hdl.handle.net/1721.1/153658

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

This is an extended technical report for our publication in the *IEEE Robotics and Automation Letters*.
Please cite the paper as: Q. Huang, J. DeGol, V. Fragoso, S. N. Sinha, and J. J. Leonard,
"Optimizing Fiducial Marker Placement for Improved Visual Localization",
IEEE Robotics and Automation Letters (RA-L), 2023.

Optimizing Fiducial Marker Placement for Improved Visual Localization

Qiangqiang Huang[1], Joseph DeGol[2], Victor Fragoso[2], Sudipta N. Sinha[2], and John J. Leonard[1]

***Abstract*—Adding fiducial markers to a scene is a well-known strategy for making visual localization algorithms more robust. Traditionally, these marker locations are selected by humans who are familiar with visual localization techniques. This paper explores the problem of automatic marker placement within a scene. Specifically, given a predetermined set of markers and a scene model, we compute optimized marker positions within the scene that can improve accuracy in visual localization. Our main contribution is a novel framework for modeling camera localizability that incorporates both natural scene features and artificial fiducial markers added to the scene. We present optimized marker placement (OMP), a greedy algorithm that is based on the camera localizability framework. We have also designed a simulation framework for testing marker placement algorithms on 3D models and images generated from synthetic scenes. We have evaluated OMP within this testbed and demonstrate an improvement in the localization rate by up to 20 percent on four different scenes.**

***Index Terms*—Localization, Computer Vision for Automation, Landmark Deployment, Fiducial Markers.**

I. Introduction

VISUAL localization is a foundational technique for AR/VR, autonomous driving, and robotic navigation and manipulation. A typical problem in visual localization is to estimate the camera pose of a query image, provided a pre-built map. While the problem has long been investigated in many fields [1], visual localization still suffers due to challenging scenes such as textureless walls and repetitive structures (e.g., Rooms A and B in Fig. 1). One common solution to these challenges is to place fiducial markers as additional texture and identifiers in the scene [2], [3]; however, placing fiducial markers in larger environments is a time consuming process and the resulting performance improvement depends on marker positions. Thus, optimizing marker placement is valuable for robust visual localization.

This work proposes an automatic approach to optimizing marker placement such that 1) the resulting marker positions yield improved accuracy in visual localization and 2) a human user will be able to place markers at positions planned by the approach (e.g., no markers on the ceiling). Specifically, the approach computes optimized marker positions, given a predetermined set of markers and a scene model. The key contributions of this work include:

[1]Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA{hqq, jleonard}@mit.edu.
[2]Microsoft, Redmond, WA 98052, USA{jodegol, victor.fragoso, sudipta.sinha}@microsoft.com.

The code is available at https://github.com/doublestrong/OMP. This work was started during Qiangqiang's internship at Microsoft and extended at MIT. Qiangqiang and John were partially supported by ONR grant N00014-18-1-2832 and ONR Neuroautonomy MURI grant N00014-19-1-2571.

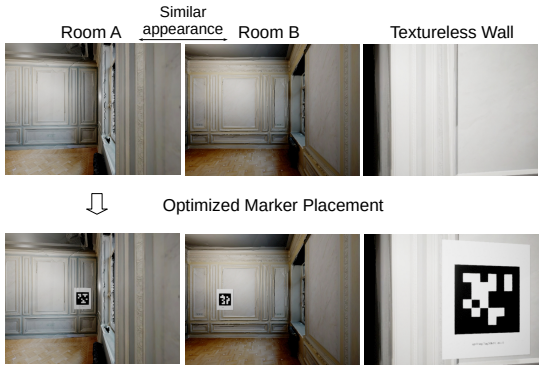

Fig. 1. Three challenging examples for visual localization. The images on the left and middle show two almost identical rooms in the scene, whereas the image on the right depicts a very weakly textured surface. Marker placements[1] in this scene guided by our optimized marker placement approach led to improved visual localization on these examples.

1) This is the first work that optimizes marker placement for visual localization based on scene features and fiducial markers.
2) We propose a novel framework that models localizability of camera poses in a scene and computes localizability scores.
3) We develop a greedy algorithm that optimizes marker positions with the goal of increased localizability scores.
4) We design a simulation framework for testing marker placement algorithms on 3D scene models that enables others to reproduce and build on our work.
5) We demonstrate that optimized marker placement by our approach can improve the localization rate by up to 20 percent on four different scenes.

II. Related Work

We briefly review some recent work related to mapping and localization with fiducial markers and marker/landmark placement optimization. Examples of fiducial markers include tag families with explicit IDs (e.g., ArUco markers [5], AprilTag [4], ChromaTag [6]) and emerging learning-based marker designs [7]. Fiducial markers are widely recognized as an effective approach for improving localization and mapping accuracy. DeGol et al. [3] demonstrate that marker IDs are useful in image matching and resectioning for structure from motion (SfM), leading to improvements in reconstruction results. The UcoSLAM system [2] integrates marker detection with a bag-of-words approach and presents more robust tracking and relocalization than SLAM techniques with no marker detection [8], [9]. However, marker placements in these SfM

[1]Fiducial markers in the examples are AprilTags [4] but our algorithm is general and can be used with any existing family of fiducial markers.

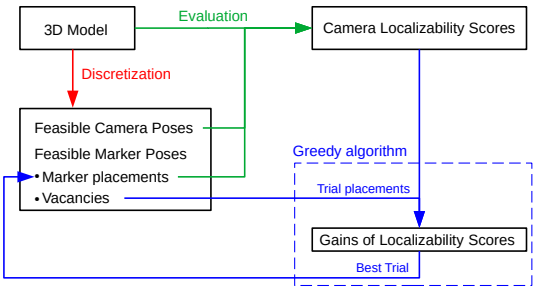

Fig. 2. An overview of our approach. We first create a set of feasible camera poses and marker poses by discretizing space in the 3D model. Then we evaluate localizability scores of the feasible camera poses and update the scores once a feasible marker pose is selected to place a marker. The marker placement is selected by a greedy algorithm as the best trial out of trial placements in the vacancies (unselected marker poses). These trial placements are ranked by gains of localizability scores.

or SLAM systems are manually determined and not planned by algorithms.

Existing work about marker deployment focuses on robotic localization without considering scene features [10]–[12]. Beinhofer et al. [13] explore optimal placement of artificial landmarks such that a robot equipped with range and/or bearing sensors repeatedly follows predetermined trajectories in planar environments with improved accuracy. Meyer-Delius et al. [14] introduce a measure that defines the uniqueness of robot poses in the context of Monte Carlo localization using laser scanners and then propose a greedy algorithm to incrementally select landmark locations for maximizing the measure. While we find the greedy algorithm is similar to ours, it is not straightforward to apply the measure to visual localization using images and scene features. Lei et al. [15] investigate landmark deployment for poses on $\mathrm{SE}(3)$ and demonstrate placing fiducial markers in a cubic environment; however, features in the scene are not involved in optimizing the marker placement.

III. Methods

We aim to compute k 3D locations in the scene for placing k fiducial markers such that after marker placement, the camera localization performance improves for query images from anywhere within the scene. In summary, we solve the global search of optimal k locations by a greedy algorithm that seeks one marker placement each time.

A. Assumptions

This work makes two assumptions: 1) A textured 3D model of the scene is available, and 2) markers and cameras are located on a 3D plane parallel to the ground plane at roughly the eye level of a person with average height. Note that the textured model can be a 3D simulation environment or a dense reconstruction of scenes. We will collect images (e.g., RGB, depth, and surface normal) and corresponding camera poses from the model and take them as input to our approach for optimizing marker placement. The second assumption ensures that our marker placement will be reachable to a human user and constrains the number of feasible camera and marker locations for computational efficiency.

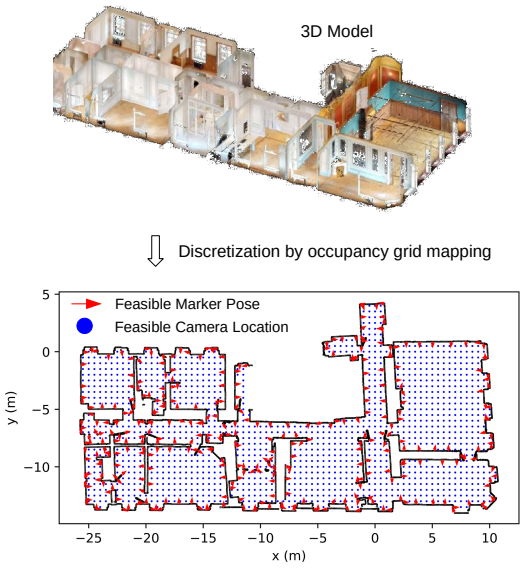

Fig. 3. Discretization of a model from the Habitat-Matterport 3D dataset [16]. We select a ground plane in the 3D model at roughly eye level of a human user. The discretized space of the ground plane consists of feasible marker poses (red arrows), which are sampled from scan points on the ground plane perimeter, and feasible camera locations (blue dots), which are centers of unoccupied cells in the 2D discrete grid.

B. Key Elements of Proposed Approach

Fig. 2 shows an overview of our approach, which is composed of three key elements: 1) discretization, 2) evaluation of camera localizability, and 3) a greedy algorithm for selecting marker placements.

1) Discretization: We first convert the ground plane in the 3D model to a discretized space of camera and marker poses, as shown in Fig. 3. The conversion is implemented by occupancy grid mapping. Centers of unoccupied grid cells are designated as feasible camera locations (dots in Fig. 3) while scan points form the perimeter of the free space (lines in Fig. 3). We uniformly downsample the scan points to generate a set of feasible marker poses $\mathcal{M}$ (arrows in Fig. 3) whose orientations are determined by surface normals in the 3D model. Note that one can choose other ways to select feasible marker poses and then still apply our marker placement algorithm. For example, the feasible marker poses can be further refined by incorporating semantics and physical constraints. It is possible that the algorithm could produce a marker placement in an infeasible location, although we found this was rare. Even so, we have done a sensitivity study showing that we can place the marker nearby the exact location and still get most of the gain[2]. We derive a set of feasible camera poses $\mathcal{C}$ from the feasible camera locations. Each of the camera locations yields n camera poses whose optical axes are parallel to the ground plane and evenly spaced in $[0, 2\pi]$ (e.g., the default $n = 8$).

2) Camera localizability score: We compute camera localizability scores by evaluating uncertainty in localizing feasible camera poses. Specifically, for any feasible camera pose $\mathbf{c} \in \mathcal{C}$ (the corresponding random variable is C), we synthesize measurements $\mathbf{z}$ to create a camera localization problem, estimate

[2]A sensitivity study about the influence of position and size deviations of markers on localization performance is available in Sec.X.

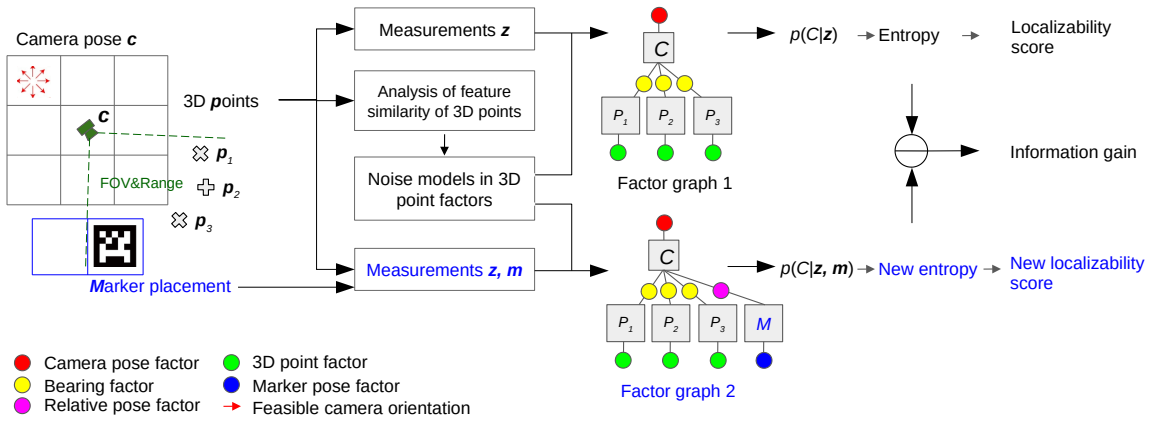

Fig. 4. Evaluation of localizability scores and the information gain brought by a marker placement. On the left we show a grid of feasible camera poses. Feasible camera poses are positioned at cell centers with orientations shown as the red arrows. The field of view of camera pose $\mathbf{c}$ covers points p_1, p_2, and p_3 in the 3D model and a marker placement on the discretized perimeter of the level set of the ground plane. We synthesize measurements $\mathbf{z}$ of the points to create a camera localization problem using scene features. The problem is represented by factor graph 1 and distribution $p(C|\mathbf{z})$ by which we can compute the entropy as well as the localizability score of the camera pose seeing no markers. We penalize contributions of repetitive structures on the localizability score via the analysis of feature similarity. With additional measurements $\mathbf{m}$ to the marker, we create another localization problem which is represented by factor graph 2 and distribution $p(C|\mathbf{z}, \mathbf{m})$. The new problem leads to a new entropy and a new localizability score.

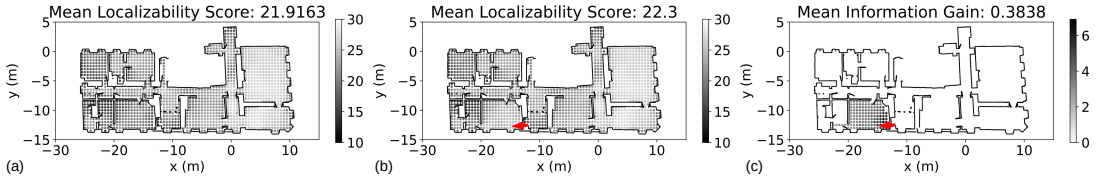

Fig. 5. Results of localizability scores: (a) no markers, (b) a trial marker placement (red arrow), and (c) the information gain. The score (or gain) at a dot is the mean score (or gain) of camera poses at the dot with all feasible orientations. Darker dots stress low localizability scores in (a) and (b) and high information gains in (c). This trial turns out to be the first marker in the optimized placement (see the Apartment in Fig. 9).

the distribution of the camera pose $p(C|\mathbf{z})$, and define the localizability score of the camera pose $l(\mathbf{c})$ as the negation of the entropy of the distribution, as shown in

$$l(\mathbf{c}) = -H(p(C|\mathbf{z})) = \mathbb{E}[\ln p(C|\mathbf{z})]. \tag{1}$$

If a new fiducial marker is added in the field of view (FOV) and range of the camera pose, the new synthetic measurement regarding the marker will change the entropy of the camera pose distribution, resulting in an information gain that quantifies the impact of the marker placement. Fig. 4 summarizes steps for evaluating the localizability score and the information gain. These steps are explained in detail in following paragraphs.

Synthesized data for computing the localizability score: The leftmost part of Fig. 4 illustrates 3D points and a feasible marker pose (i.e., trial marker placement) that are in the FOV/range of a feasible camera pose[3]. We collect RGB and depth images at the camera pose in the 3D model. These images will be used to compute 3D points and descriptors of features (e.g., SIFT [17]). We use these known poses and points to synthesize measurements and estimate probability density functions (PDFs) of the camera pose variable. Measurements $\mathbf{z}$ in Fig. 4 contain the camera pose, the 3D points, and bearings between them. Thus the PDF $p(C|\mathbf{z})$, which is represented by factor graph 1, expresses the distribution of the camera pose constrained by the 3D points. Placing a marker in the FOV/range of the camera leads to new synthetic measurements $\mathbf{m}$ of the marker pose and the relative pose between the marker and the camera. As a result, the camera pose is further constrained by measurements $\mathbf{m}$ thus is described by a new PDF $p(C|\mathbf{z}, \mathbf{m})$ represented by factor graph 2 in Fig. 4. We use an approach that is similar to the one proposed by Stachniss et al. [18] to define the information gain of a marker placement. The information gain is defined as the change of entropy that the marker placement $\mathbf{m}$ yields at the camera pose $\mathbf{c}$, as seen in

$$I(\mathbf{m}, \mathbf{c}) = H(p(C|\mathbf{z})) - H(p(C|\mathbf{z}, \mathbf{m})). \tag{2}$$

Fig. 5a shows localizability scores of camera poses in the original ground plane with no marker placement. Note that the score at a dot in the figure is the mean score of camera poses with all feasible orientations. Fig. 5b shows localizability scores after adding a marker (the arrow) to the ground plane perimeter. The scores increase in the region around the marker, indicated by the brighter dots in the region in Fig. 5b and the information gain in Fig. 5c.

Analysis of feature similarity of 3D points: Repetitive structures in scenes cause similar features across RGB images and can result in localizing to a wrong location. To reduce the contribution of repetitive structures to localizability scores, we penalize the localizability score if similar features appear in the FOV of the camera. Specifically, when modeling 3D points with similar features in factor graphs, we set greater uncertainty in noise models of 3D point factors to encode the

[3]In practice, one can further refine marker poses in the FOV by considering marker sizes and rejecting corner cases that may fail the detection of markers. The cases include marker poses that are too close to the boundary of the view frustum of the camera.

Algorithm 1: Optimized Marker Placement (OMP)

Input: The number of markers k, the list of feasible marker poses $\mathcal{M}$, the ground plane space $\mathcal{S}$
Output: k marker poses

Initialize an empty list for storing selected marker poses $\mathcal{O}$
repeat k **times**
  Initialize the best marker pose $T^\star = \emptyset$
  Initialize the highest localizability gain $g^\star = -\inf$
  Evaluate localizability scores $\mathcal{L}^\star$ of camera poses in space $\mathcal{S}$
  for Pose T in $\mathcal{M}$ **do**
    Place a marker at pose T in space $\mathcal{S}$
    Evaluate localizability scores $\mathcal{L}$ of camera poses
    Compute information gains $\mathcal{I} = \mathcal{L} - \mathcal{L}^\star$
    Evaluate localizability gain g of the marker by (6)
    if $g > g^\star$ **then**
      $T^\star = T$
      $g^\star = g$
    Remove the marker from space $\mathcal{S}$
  Push $T^\star$ to $\mathcal{O}$
  Place a marker at pose $T^\star$ in space $\mathcal{S}$
  Remove $T^\star$ from $\mathcal{M}$
return List of marker poses $\mathcal{O}$

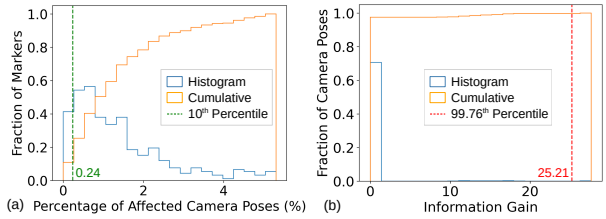

Fig. 6. Histograms for the HM3D apartment model: (a) percentage of affected camera poses and (b) information gains at camera poses yielded by a marker. The most visible 90% markers (i.e., $v = 90$) means 10^{th} percentile in (a), determining the percentile $q = 99.76$ by (9). The 99.76^{th} percentile in (b) indicates a localizability gain 25.21 of the marker by (6).

fact that similar 3D points are ambiguous and less informative. (3) shows the 3D point factor that formulates the difference between the noisy 3D location $\tilde{\mathbf{p}}$ and true 3D location $\mathbf{p}$ using a Gaussian distribution

$$p(\tilde{\mathbf{p}}|\mathbf{p}) = \mathcal{N}(\tilde{\mathbf{p}} - \mathbf{p}; \mathbf{0}, \Sigma_{\mathbf{p}}) \tag{3}$$

where $\Sigma_{\mathbf{p}}$ is the covariance we set for modeling noise. For example, in the leftmost part of Fig. 4, points $\mathbf{p}_1$ and $\mathbf{p}_3$ are visually similar, so we set big covariances in 3D point factors of $\mathbf{p}_1$ and $\mathbf{p}_3$. Informally, factors with big covariances impose loose constraints on the camera pose distribution, leading to lower contributions on the localizability score.

We perform an analysis of feature similarity of 3D points to determine noise models in 3D point factors (i.e., $\Sigma_{\mathbf{p}}$ in (3)), as shown in the flow chart in Fig. 4. The analysis is to count the number of similar 3D points to any 3D point. The resulting covariance $\Sigma_{\mathbf{p}}$ is formulated as

$$\Sigma_{\mathbf{p}} = (1 + n_{\mathbf{p}})\Sigma_0 \tag{4}$$

where Σ_0 is a base covariance (e.g., $diag(2.5, 2.5, 2.5) \times 10^{-3}\ m^2$ in our experiments) and $n_{\mathbf{p}}$ denotes the number of similar 3D points to the query point $\mathbf{p}$. 3D points observed by all feasible camera poses are filtered to select similar ones of the query point. The selection is determined by two criteria: 1) the selected points have similar descriptors to the query point and 2) the selected points are not too close to the 3D location of the query point. The intuition is that, if two areas in the scene look similar but they are far away from each other, a wrong place recognition would incur a huge localization error.

Estimation of camera pose distributions: We use the Laplace approximation [19, Ch. 4.4] to estimate a Gaussian distribution that approximates the camera pose distribution encountered in the synthetic localization problem. The mean of the Gaussian is the known feasible camera pose so the covariance Σ is the only unknown. The covariance can be approximated by an estimated Hessian of the negative logarithm of the camera pose distribution at the mean (see [20, Sec. 2] for the estimation of the covariance). Thus the entropy encountered in the synthetic localization problem can be approximated by

$$H(p(C|\cdot)) \approx \frac{1}{2}\ln|\Sigma| + \frac{d}{2}(1 + \ln(2\pi)) \tag{5}$$

where the dimensionality d is 6 for 6DOF poses.

3) The greedy algorithm[4]*:* The algorithm sequentially selects k poses from feasible marker poses $\mathcal{M}$ (see Algorithm 1). The algorithm executes k loops to search the best k poses. In each loop, we update localizability scores, tentatively place a marker at any feasible marker pose, and compute localizability gains of trial marker placements. The best pose that earns the highest localizability gain will be removed from feasible marker poses and be permanently occupied by a marker. The marker will influence future updates of localizability scores.

We summarize information gains at all feasible camera poses in the scene, using a single scalar quantity that we refer to as localizability gain. Informally, one could think of the localizability gain as the reward for placing an additional marker at a specific position. The localizability gain of any marker placement $\mathbf{m}$ is defined as the q^{th} percentile of information gains that marker $\mathbf{m}$ yields at all feasible camera poses $\mathcal{C}$, as seen in

$$g(\mathbf{m}) = \inf\{i \in \mathbb{R} : F_I(i) \geq \frac{q}{100}\}, \tag{6}$$

where $F_I(\cdot)$ is the cumulative distribution function (CDF) after sorting the information gains at all camera poses

$$\mathcal{I} = \{I(\mathbf{m}, \mathbf{c}) : \mathbf{c} \in \mathcal{C}\}. \tag{7}$$

The choice of percentile $q \in [0, 100]$ is crucial and dependent on environments (i.e., the ground plane). For example, in a large environment where any marker is only visible to a small fraction of feasible camera poses, a low percentile q would likely incur zero localizability gains for all markers since camera poses seeing no markers receive zero information gains and constitute a great portion of the information gain distribution $\mathcal{I}$.

We use an adaptive approach to determine the percentile q before computing the localizability gain. The approach introduces a hyperparameter $v \in [0, 100]$ and ensures that the most visible v percent of markers earn nonzero localizability gains. A high v allows more markers, even the ones stuck in corners, to effectively join in the selection of best marker while a low v favors the most visible ones among feasible

[4]Discussion about the complexity of the algorithm and the possibility of generalizing the greedy algorithm is available in Sec. VII.

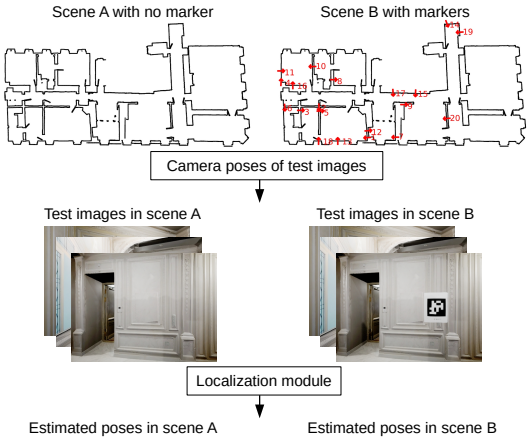

Fig. 7. The flowchart of our system for performing camera localization experiments. Scenes with different marker placements share the same set of camera poses for acquiring test images and the same localization module.

marker poses. In the ground plane space, for any marker $\mathbf{m}$, we can find a set of affected camera poses $\mathcal{C}_\mathbf{m}$ that are supposed to see the marker (i.e., nonzero info. gain). We can derive a CDF $F_P(p)$ using percentages of affected camera poses for all markers

$$\mathcal{P} = \left\{ \frac{|\mathcal{C}_\mathbf{m}|}{|\mathcal{C}|} \times 100 : \mathbf{m} \in \mathcal{M} \right\}. \tag{8}$$

To ensure only the most visible v percent of markers earn nonzero localizability gains, the percentile q is determined by the $(100-v)^{th}$ percentile in percentages of affected camera poses, as seen in

$$q = 100 - \inf \left\{ p \in [0, 100] : F_P(p) \geq \frac{100-v}{100} \right\}. \tag{9}$$

(9) indicates q is a non-decreasing function of v. When v approaches 100, q approaches 100 as well so only markers that earn a greater maximum in information gains will be considered in the best marker selection (see (6)); when v approaches 0, q approaches 0 as well so the best marker will only be selected from markers that influence large areas. Thus the choice of hyperparameter v can reflect the trade-off between helping the worst single camera pose and influencing the most camera poses.

Fig. 6 shows an example for computing the percentile q and the localizability gain for the marker placement in Fig. 5. We set $v = 90$ as the default setting so the most visible 90% markers receive nonzero localizability gains and are effective best marker candidates. This setting results in a marker placement strategy that tends to support worst camera poses instead of area coverage, as shown in the optimized marker placement for the apartment model in Fig. 9. No markers are placed in the two big rooms on the right of the apartment since (i) camera poses in these rooms already enjoyed good localizability scores (see Fig. 5a) and (ii) a large hyperparameter v does not emphasize area coverage.

IV. Experimental Setup

A. Implementation

We implemented all three key elements and Algorithm 1 in Sec. III-B in Python with assistance of a few open source software packages. We used the Unreal Engine 4.27 [21] and the AirSim library (v1.8.1) [22] to simulate and collect images from 3D models. We used the Open3D library [23] to downsample scan points to get candidate marker locations. We used the GTSAM library [24] to create factor graphs and estimate covariances in Gaussian approximations of camera pose distributions. The SIFT feature [17] was used throughout our experiments.

Additionally, we implemented a simulation system for testing marker placement algorithms and a camera localization module for estimating camera poses of test images. Fig. 7 presents a flowchart of the system. The system adds markers to a scene model at positions planned by marker placement algorithms and then generates test images from the same set of camera poses for different marker placements for the fairness in comparison. We stress three advantages of the simulation system over real world pipelines for performing camera localization experiments: 1) reproducible data collection by other researchers for future development of marker placement algorithms, 2) a large number of test images that cover the scene, 3) consistent camera poses for generating test images in scenes with different marker placements.

B. Evaluation

1) Methods for comparison: We compare our algorithm OMP with 1) no marker placement, 2) random marker placements, 3) uniform marker placements, and 4) markers placed by a human. Random marker placements refer to uniformly weighted samples from feasible marker poses. Uniform placements distribute the markers roughly uniformly along the perimeter of the environment (see [14] for details). We generated 5 versions of random and uniform placements for each scene and all placements were manually inspected in scene models to ensure reasonable quality. The comparison with humans is only conducted in the real experiment. The human prioritizes centers in less textured areas.

2) Scenes: The method comparison is performed on four scenes: apartment, studio, office, and room, as seen in Fig. 9. The first two are pre-built dense maps of realworld spaces, provided by the Habitat-Matterport 3D (HM3D) Research Dataset [16], while the third model is an Unreal Engine simulation environment that resembles typical realworld offices[5]. The first three are for simulated experiments. The last one is a motion capture room at MIT for the real experiment (see Sec. IX). The textured mesh of the room was created by fusing RGB-D images from groundtruth poses, using the volumetric fusion [25] and marching-cubes algorithms and the screened Poisson surface reconstruction [26]. Table I lists specifics of these models.

3) The localization module: Fig. 8 presents the flowchart of our localization module. The localization module is similar to standard approaches [27] but with an extra function of fiducial

[5]The serial number of the apartment model is 00770-NBg5UqG3di3 in the HM3D dataset and that of the studio model is 00254-YMNvYDhK8mB. We inspected all scenes in the dataset and chose these two as representatives of medium and large scenes with textureless areas and potential perceptual aliasing. The office model is the ThreeDee Office project in the Unreal Engine Marketplace.

TABLE I. Specifics of scenes

Model	Area (m^2)	# of map images	# of test images
Apartment	339.3	10856	10000
Studio	149.6	2832	3000
Office	108.3	1768	2000
Room	21.0	250	200

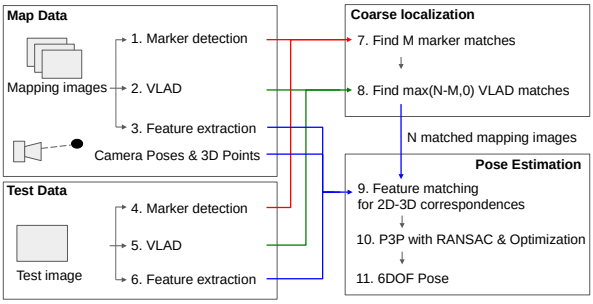

Fig. 8. The localization module using fiducial marker detection. The numbers indicate the order of different operations.

marker detection, provided by the AprilTag library [4]. The tag detection and VLAD descriptors [28] were sequentially employed to find matched images in the map data. Camera poses were estimated using P3P [29] with RANSAC [30] followed by Levenberg-Marquardt optimization [31]. The rotation error δ_R is defined as the angular distance between the estimated rotation matrix $\hat{R}$ and the groundtruth rotation R while the translation error δ_t is defined as the Euclidean distance between the estimated translation $\hat{t}$ and the groundtruth translation t, as seen in

$$\delta_R = \left|\arccos\left(\frac{\mathrm{tr}(\hat{R}^\mathsf{T} R) - 1}{2}\right)\right|, \tag{10}$$

$$\delta_t = \left\|\hat{t} - t\right\|_2. \tag{11}$$

4) The map and test data: In simulated experiments, the camera was set to a FOV of 90 degrees and a range of 10 meters (RGB res. 600×450, depth res. 300×225). The camera poses for collecting the map data are the same as the feasible camera poses in the ground plane space. The camera poses for collecting test images are sampled from the feasible camera poses with weights and then perturbed by translation and rotation noises that are subject to a uniform distribution in $[-0.5, 0.5]$. We intend to sample more densely from the difficult areas, which are of our interest, so the weights in the sampling correlate with localizability scores for generating more test images around low-scoring camera poses[6]. Let $\mathcal{L} = \{l(\mathbf{c}) : \mathbf{c} \in \mathcal{C}\}$ be the set of localizability scores of feasible camera poses in the ground plane space with no markers. The weights are defined as

$$\mathcal{W} = \{2l^\star - \bar{l} - l(\mathbf{c}) : \mathbf{c} \in \mathcal{C}\}, \tag{12}$$

where $l^\star$ is the maximal score in $\mathcal{L}$ and $\bar{l}$ is the mean of all scores. Thus all weights will be non-negative and a lower score incurs a greater weight. In the real experiment, we used the Realsense L515 camera for RGB-D data (image res. 1280×720) and the OptiTrack system for groundtruth poses. The map and test data were sampled along two lawn-mower paths around feasible camera poses (see Sec. IX).

[6]Results on test images uniformly sampled are available in Sec. VIII.

V. Results

We present two sections of results. In the first section, we present results comparing different marker placement methods. Next, we show a parameter study about factors that can affect our algorithm and the localization performance. The main metric we analyze is the recall, which is defined as the percent of test images localized within certain thresholds of errors: (5 cm, 5 deg) for simulated experiments and (30 cm, 10 deg) for the real experiment considering errors in the dense map, sensor noise, and large textureless areas. The default hyperparameter v is 90.

A. Comparison of Marker Placement Methods

As optimized marker placements in Fig. 9 show, our algorithm focuses on placing markers around low-scoring areas and improves mean localizability scores by a large margin. For example, the largest room in the studio model only receives a single marker (marker 9 on the top right of the studio) since the room already possesses good localizability scores even with no markers.

Optimized marker placements consistently outperform no marker placement, random placements, and uniform placements on the recall. After placing 20 markers, our algorithm improves the recall by over 1.5 percentages for the apartment model, 3.0 percentages for the studio model, 20.0 percentages for the office model, and over 20.0 percentages for the room scene. Note that the area of the apartment model is very big and the model has attained a high recall 85% with no assistance of markers so the increment of recall for the apartment model was expected to be lower than that for the other models. The real experiment in the room scene shows that our algorithm is on par with markers placed by a human. Although our experiment demonstrates the efficacy of optimizing marker placements in 3D models for realworld applications, we emphasize that the efficacy relies on the similarity between rendered and real images. Vision features in rendered images can be affected by many factors including mesh quality and lighting. For example, we covered the glass door in the room by a well-textured poster to reduce the difficulty in 3D reconstruction. In addition, if one has quality real RGB-D data at feasible camera poses, the textured mesh is not needed for using our marker placement algorithm.

B. Parameter Study

We design four experiment groups and change one of the default parameters in each experiment group. The experiment groups are 1) different values of v in the greedy algorithm, 2) enabling/disabling marker detection in the localization module, 3) low-scoring/uniform test data and 4) enabling/disabling the analysis of feature similarity, as seen in Table II. The default setting is with $v = 90$, marker detection enabled, the low-scoring test data that has more test images in low-scoring areas in the ground plane, and the similarity analysis where similar 3D points are downweighted in the localizability score. For the parameter study, we use the office model.

Too large or small values of hyperparameter v incur lower improvements of the recall. As explained in

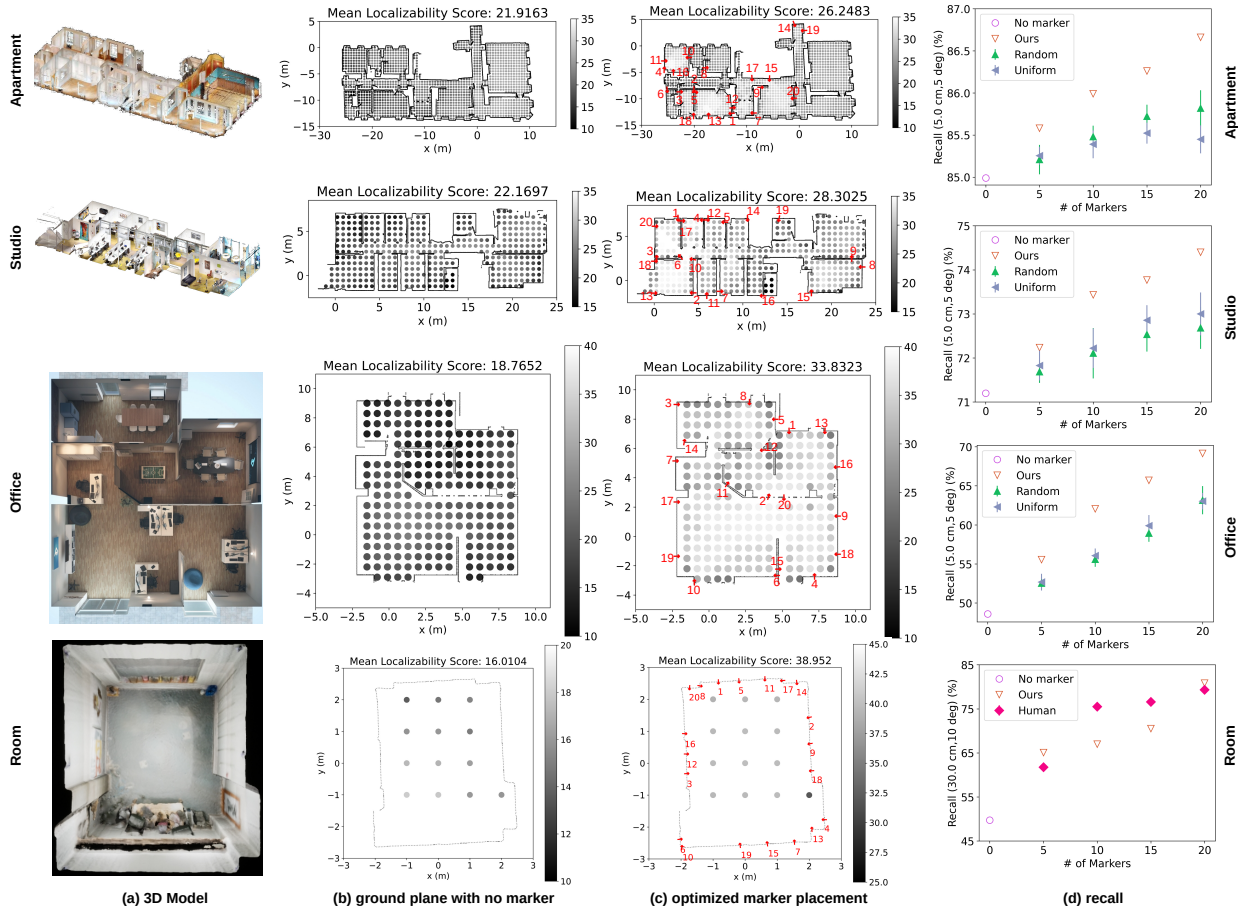

Fig. 9. Results for all scenes: (a) 3D models, (b) ground plane space with no markers where darker dots indicate lower localizability scores, (c) optimized marker placements where the red arrows represent optimized marker placements and the numbers beside the arrows indicate the order of marker placements, and (d) the recall in camera localization experiments. We exclude camera poses near the bottom of the room where a table occupies.

TABLE II. Parameter study about the hyperparameter v, the test data, enabling/disabling marker detection, and enabling/disabling the similarity analysis.

Experiment group	Recall of test images with k markers (%)				
	$k=0$	5	10	15	20
$v = 90$ (df.)	**48.6**	**55.5**	**62.1**	**65.7**	**69.2**
$v = 99$	48.6	55.5	60.4	64.5	67.4
$v = 70$	48.6	54.8	61.1	63.2	66.6
$v = 50$	48.6	54.3	57.6	63.9	66.8
Marker detect. on (df.)	**48.6**	**55.5**	**62.1**	**65.7**	**69.2**
Marker detect. off	48.6	55.2	60.7	64.2	67.5
Low-scoring data (df.)	48.6	55.5	62.1	65.7	69.2
Unif. test data	**57.4**	**63.7**	**68.4**	**72.1**	**74.8**
Similarity analysis (df.)	**48.6**	**55.5**	**62.1**	**65.7**	**69.2**
Sim. analysis disabled	48.6	55.4	61.8	65.0	67.8

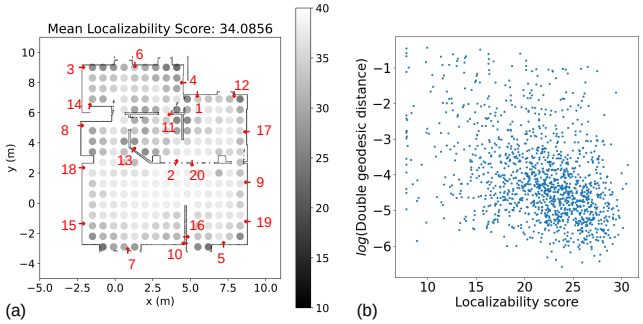

Fig. 10. Parameter study: (a) the optimized marker placement after disabling the similarity analysis and (b) scatter plot of the localizability score and the log of estimation error of test images. The error is computed as the double Geodesic distance, $\sqrt{\delta_R^2 + \delta_t^2}$. To avoid outliers, samples are admitted to the plot only if the translation and rotation errors are within $(50cm, 50deg)$. The Pearson correlation coefficient and p-value for testing non-correlation is $(-0.41, 2.4 \times 10^{-55})$.

Sec. III-B3, lower v favors markers that cover larger areas while greater v tends to stress the worst single camera pose. Table II shows that the default value ($v = 90$) consistently outperforms small value 50 and large value 99, indicating that the default attains a good balance between area coverage and helping the worst cases.

The localizability score can be a good indicator of localization errors. Table II shows that uniform test samples enjoy greater recall than test samples that stress low-scoring areas by at least 5 percentages. Fig. 10b indicates a statistically significant, negative correlation between the localizability score and the localization error.

Both the visual appearance and decoded label of markers are helpful for localization. We disable marker detection in the localization module (Fig. 8) to investigate its impact on the recall. Table II shows that markers still improve the recall even though the detector is turned off. The reason is that the visual appearance of markers is still helpful for coarse localization and pose estimation in the localization module.

Deactivating the analysis of feature similarity decreases the recall. Fig. 10a presents the marker placement after disabling the similarity analysis (i.e., no scaling in (4)). The first five markers remain in the same positions as those guided by the similarity analysis (Fig. 9c). Thus the recall does not change significantly until placing 10 markers, as shown in the last group in Table II. The decrease in the recall with no similarity analysis justifies the efficacy of downweighting similar features in the computation of the localizability score.

VI. Conclusion and Future Work

This work provides a promising foundation for optimizing and evaluating marker placement for improved visual localization. Our OMP algorithm defines localizability scores for different areas in the scene and uses a greedy algorithm to find the best marker placements in the sense of increased localizability scores. We applied the OMP algorithm to four scenes and demonstrated that OMP consistently improves camera localization recall compared to random and uniform marker placements. We believe that our marker placement approach is also useful for SLAM. However, our approach could be further extended to compute optimal marker placement for specific tasks in SLAM. One potential idea involves extending the localizability score to a trackability score that incorporates uncertainty propagation along a robot path while restricting feasible camera poses to the operating area of the robot.

The OMP algorithm only considers placing markers in a scene model (i.e., mapped areas in the scene), however, regions in the scene which are challenging for mapping are also likely to be good locations for placing markers. Thus, it would be worth exploring ways to extend the algorithm to prioritize marker placements in regions that are either partially or inadequately mapped. Further research is also needed to compute more accurate localizability scores and explore more efficient optimization methods beyond the greedy algorithm, including: (1) joint optimization of marker poses and sizes, (2) extending the single-layer ground plane to multi-layer planes for deploying markers in multi-storey structures, (3) using non-Gaussian distribution estimation techniques to compute localizability scores, and (4) applying submodular optimization to jointly select multiple best markers together with fewer iterations.

References

[1] Z. Zhang, T. Sattler, and D. Scaramuzza, "Reference pose generation for long-term visual localization via learned features and view synthesis," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 821–844, 2021.

[2] R. Munoz-Salinas and R. Medina-Carnicer, "UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers," *Pattern Recognit.*, vol. 101, p. 107193, May 2020.

[3] J. DeGol, T. Bretl, and D. Hoiem, "Improved structure from motion using fiducial marker matching," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 281–296.

[4] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3400–3407.

[5] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[6] J. DeGol, T. Bretl, and D. Hoiem, "Chromatag: A colored marker and fast detection algorithm," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1481 – 1490.

[7] Z. Zhang, Y. Hu, G. Yu, and J. Dai, "DeepTag: A general framework for fiducial marker design and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[8] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[9] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2198–2204.

[10] Y. Chen, J.-A. Francisco, W. Trappe, and R. P. Martin, "A practical approach to landmark deployment for indoor localization," in *Proc. IEEE Commun. Soc. Sens. Ad Hoc Commun. Netw.*, vol. 1, 2006, pp. 365–373.

[11] M. P. Vitus and C. J. Tomlin, "Sensor placement for improved robotic navigation," *Proc. Robot.: Sci. Syst.*, pp. 217–224, 2011.

[12] D. B. Jourdan and N. Roy, "Optimal sensor placement for agent localization," *ACM Trans. Sens. Netw.*, vol. 4, no. 3, pp. 1–40, May 2008.

[13] M. Beinhofer, J. Müller, and W. Burgard, "Effective landmark placement for accurate and reliable mobile robot navigation," *Robot. Auton. Syst.*, vol. 61, no. 10, pp. 1060–1069, Oct. 2013.

[14] D. Meyer-Delius, M. Beinhofer, A. Kleiner, and W. Burgard, "Using artificial landmarks to reduce the ambiguity in the environment of a mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 5173–5178.

[15] Z. Lei, X. Chen, Y. Tan, X. Chen, and L. Chai, "Optimization of directional landmark deployment for visual observer on SE(3)," *IEEE Trans. Ind. Electron.*, pp. 1–10, 2022.

[16] S. K. Ramakrishnan *et al.*, "Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI," in *Proc. Conf. Neural Inform. Process. Syst. Dataset. Benchmark. Track (Round 2)*, 2021.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[18] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters." in *Proc. Robot.: Sci. Syst.*, vol. 2, 2005, pp. 65–72.

[19] C. M. Bishop, *Pattern recognition and machine learning*. New York, USA: Springer, 2006.

[20] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1198–1210, Dec. 2009.

[21] Epic Games, "Unreal engine." [Online]. Available: https://www.unrealengine.com

[22] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Serv. Robot.* Springer, 2018, pp. 621–635.

[23] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[24] F. Dellaert *et al.*, "borglab/gtsam," May 2022. [Online]. Available: https://doi.org/10.5281/zenodo.5794541

[25] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. Annu. Conf. Comput. Graph. Interact. Tech.*, 1996, pp. 303–312.

[26] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[27] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 716–12 725.

[28] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[29] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Supplementary Material

VII. Discussion about the OMP Algorithm

The crux of computation in Algorithm 1 is evaluating localizability scores. To avoid redundant computation, we update the localizability score of a camera pose only if the newly added marker is covisible to the camera pose. The complexity of Algorithm 1 is $O(|\mathcal{C}| + k|\mathcal{M}| \max_{\mathbf{m}}(|\mathcal{C}_{\mathbf{m}}|))$ where $O(1)$ is the complexity for evaluating the localizability score of a camera pose. $\max_{\mathbf{m}}(|\mathcal{C}_{\mathbf{m}}|)$ denotes the maximal number of covisible camera poses to a marker so it generally increases with the FOV and range of the camera. The first term $|\mathcal{C}|$ indicates the cost for initializing localizability scores over all feasible camera poses while $\max_{\mathbf{m}}(|\mathcal{C}_{\mathbf{m}}|)$ in the second term bounds the cost for evaluating the localizability gain brought by a marker. For each of the k loops, we evaluate localizability gains of all $|\mathcal{M}|$ markers. Note that the time complexity can be further reduced because it is not necessary to re-evaluate localizability gains for all markers in each loop (lines 8-10 in Algorithm 1). For example, if the covisible camera poses of an unselected marker have not been affected by all selected markers, we do not need to re-compute the localizability gain of that unselected marker.

We discuss the possibility of generalizing the greedy algorithm by re-defining the localizability score. For example, one can use the pose estimation error from a visual localization system (e.g., Fig. 8) to replace the localization score and keep the rest of the algorithm the same. The new marker placement based on the error may enjoy advantages in localization experiments using the same localization system since the marker placement is directly optimized for the system. However, updating the error along with trial marker placements is computationally much more expensive than evaluating the localization score since we need to add markers to the 3D scene model, generate new map and test images, update the map in the localization system, estimate camera poses of test images using the system, and compute the pose estimation error. In contrast, updating the localizability score just needs to re-estimate camera pose distributions, as shown in Fig. 4.

VIII. Uniformly Sampled Test Images

We show recall of uniformly sampled test images in Table III.

TABLE III. Recall (%) of test images that are uniformly sampled. Rand. refers to random marker placements and Unif. refers to uniform marker placements.

Scene (NoMarker)	Method	Mean±STD with k markers			
		$k=5$	$k=10$	$k=15$	$k=20$
Apt. (88.2)	Ours	**88.6**	**88.8**	**89.2**	**89.5**
	Rand.	88.5±0.1	88.8±0.1	89.0±0.1	89.1±0.2
	Unif.	88.4±0.1	88.6±0.1	88.7±0.1	88.9±0.1
Studio (80.3)	Ours	**81.4**	**82.7**	**83.0**	**83.1**
	Rand.	80.8±0.3	81.4±0.4	81.2±0.4	81.4±0.7
	Unif.	80.9±0.2	81.2±0.3	81.6±0.5	81.8±0.2
Office (57.4)	Ours	**63.7**	**68.4**	**72.1**	**74.8**
	Rand.	60.7±0.7	63.0±1.0	66.7±1.5	69.1±1.8
	Unif.	60.0±0.6	63.0±0.7	67.6±0.9	69.9±1.3

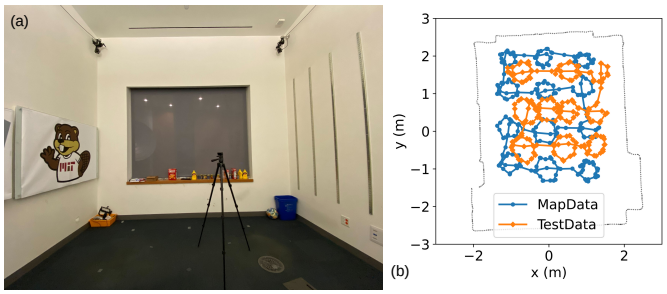

Fig. 11. Real experiment: (a) the motion capture room and (b) paths for collecting data.

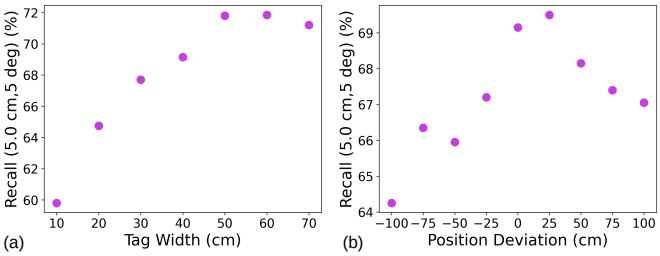

Fig. 12. Sensitivity study: (a) re-sizing tags and (b) applying different position deviations to marker poses planned by the OMP algorithm.

IX. Setups in the Real Experiment

The motion capture room for the real world experiment is shown in Fig. 11. Fig. 11b shows the lawn mower paths for collecting the map and test data.

X. Sensitivity Study of Marker Sizes and Positions

It is quite likely that a user will not be able to place fiducial markers exactly at the positions computed by the OMP algorithm; meanwhile, different users may print fiducial markers with different sizes. Thus we investigate the impact of position deviations and marker sizes on the recall. For the sensitivity study, we used the office model.

Enlarging markers up to a certain size keeps increasing the recall. Fig. 12a shows that, under 50 cm, larger tag widths lead to greater recall (note that the threshold 50 cm should correlate with environments). Excessively large sizes can degrade the recall because the markers become too big to be detected from nearby views.

Mild position deviations slightly degrade the performance of the optimized marker placement. All 20 markers planned by the OMP algorithm were moved left or right by certain distances to implement position deviations. Fig. 12b shows the recall can decrease by 2% in the presence of ± 0.25 meters position deviations and by 5% in the presence of ± 1 meter position deviations, compared with zero position deviation. However, marker placements with the position deviations still outperform no marker placement by a large margin ($\sim$ 15 percentages in the recall).