Abstract—Adding fiducial markers to a scene is a well-known strategy for making visual localization algorithms more robust. Traditionally, these markers are placed by humans who are familiar with visual localization techniques. This letter 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 [2], 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 [3], [4]; 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) 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 [6], AprilTag [5], ChromaTag [7]) and emerging learning-based marker designs [8]. Fiducial markers are widely recognized as an effective approach for improving localization and mapping accuracy. DeGol et al. [4] 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 [3] integrates marker detection with a bag-of-words approach and presents more robust tracking and relocalization than SLAM techniques with no marker detection [9], [10]. However, marker placements in these SfM or SLAM systems are manually determined and not planned by algorithms.

Existing work about marker deployment focuses on robotic localization without considering scene features [11], [12], [13]. Beinhofer et al. [14] 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. [15] 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. [16] investigate landmark deployment for poses on 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.

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 $M$ (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. We derive a set of feasible camera poses $C$ from the feasible camera locations. Each of the camera locations yields $n$ camera poses whose optical axes are parallel to the ground plane.

2) A sensitivity study about the influence of position and size deviations of markers on localization performance is available in the supplement [1].
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 $c$ 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 $z$ of the points to create a camera localization problem using scene features. The problem is represented by factor graph 1 and distribution $p(C|z)$ 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 $m$ to the marker, we create another localization problem which is represented by factor graph 2 and distribution $p(C|z, m)$. The new problem leads to a new entropy and a new localizability score.

We use an approach that is similar to the one proposed by Stachniss et al. [19] to define the information gain of a marker placement. The information gain is defined as the change of entropy that the marker placement $m$ yields at the camera pose $c$ as seen in

$$I(m, c) = H(p(C|z)) - H(p(C|z, m)).$$

Fig. 5(a) 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. 5(b) 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. 5(b) and the information gain in Fig. 5(c).

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

$$I(m, c) = H(p(C|z)) - H(p(C|z, m)).$$

Algorithm 1: Optimized Marker Placement (OMP).

**Input:** The number of markers $k$, the list of feasible marker poses $M$, the ground plane space $S$.

**Output:** $k$ marker poses

1. Initialize an empty list for storing selected marker poses $O$.
2. **repeat** $k$ times
3. Initialize the best marker pose $T^* = \emptyset$.
4. Initialize the highest localizability gain $g^* = -\infty$.
5. Evaluate localizability scores $L^*$ of camera poses in space $S$.
6. For Pose $T$ in $M$ do
   7. Place a marker at pose $T$ in space $S$.
   8. Evaluate localizability scores $L$ of camera poses.
   9. Compute information gains $L = L - L^*$.
   10. Evaluate localizability gain $g$ of the marker by (6).
   11. if $g > g^*$ then
       12. $T^* = T$
       13. $g^* = g$.
   14. Remove the marker from space $S$.
15. Push $T^*$ to $O$.
16. Place a marker at pose $T^*$ in space $S$.
17. Remove $T^*$ from $M$.
18. **return** List of marker poses $O$.

We use an approach that is similar to the one proposed by Stachniss et al. [19] to define the information gain of a marker placement. The information gain is defined as the change of entropy that the marker placement $m$ yields at the camera pose $c$, as seen in

$$I(m, c) = H(p(C|z)) - H(p(C|z, m)).$$

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

\[
p(\tilde{p}|p) = \mathcal{N}(\tilde{p} - p; 0, \Sigma_p)
\]

where \( \Sigma_p \) is the covariance we set for modeling noise. For \( n \geq 6 \) for 6DOF poses. \( \Sigma_p \) is defined as the known feasible camera pose so the covariance \( \Sigma_p \) 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 localization gains for all markers since camera poses seeing no markers receive zero information gains and constitute a great portion of the information gain distribution \( \mathcal{I} \).

**Estimation of camera pose distributions:** We use the Laplace approximation [20, 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 \( \Sigma \) 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 [21, Sec. 2] for the estimation of the covariance). Thus the entropy encountered in the synthetic localization problem can be approximated by

\[
H(p(C|\cdot)) \approx \frac{1}{2} \ln |\Sigma| + \frac{1}{2}(1 + \ln(2\pi))
\]

where the dimensionality \( d \) is 6 for 6DOF poses.

3) **The Greedy algorithm**: 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 localization 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 \( m \) is defined as the \( q \)th percentile of information gains that marker \( m \) yields at all feasible camera poses \( C \), as seen in

\[
g(m) = \inf \left\{ i \in \mathbb{R} : F_I(i) \geq \frac{q}{100} \right\},
\]

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

\[
\mathcal{I} = \{ I(m, c) : c \in C \}.
\]

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 localization 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 localization 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 marker poses. In the ground plane space, for any marker \( m \), we can find a set of affected camera poses \( C_m \) 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

\[
\mathcal{P} = \left\{ \frac{|C_m|}{|C|} \times 100 : m \in \mathcal{M} \right\}.
\]

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

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

(9) indicates \( q \) is a non-decreasing function of \( v \). When \( v \) approaches 100, \( q \) approaches 100 as well so only markers that
in Python with assistance of a few open source software packages. We used the Unreal Engine 4.27 [22] and the AirSim library (v1.8.1) [23] to simulate and collect images from 3D models. We used the Open3D library [24] to downsample scan points to get candidate marker locations. We used the GTSAM library [25] to create factor graphs and estimate covariances in Gaussian approximations of camera pose distributions. The SIFT feature [18] 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 [15] 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 [17], while the third model is an Unreal Engine simulation environment that resembles typical realworld offices. The first three are for simulated experiments. The last one is a motion capture room at MIT for the real experiment [1]. The textured mesh of the room was created by fusing RGB-D images from groundtruth poses, in these rooms already enjoyed good localizability scores (see Fig. 5(a)) 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 Section III-B in Python with assistance of a few open source software packages. We used the Unreal Engine 4.27 [22] and the AirSim library (v1.8.1) [23] to simulate and collect images from 3D models. We used the Open3D library [24] to downsample scan points to get candidate marker locations. We used the GTSAM library [25] to create factor graphs and estimate covariances in Gaussian approximations of camera pose distributions. The SIFT feature [18] 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 [15] 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 [17], while the third model is an Unreal Engine simulation environment that resembles typical realworld offices. The first three are for simulated experiments. The last one is a motion capture room at MIT for the real experiment [1]. The textured mesh of the room was created by fusing RGB-D images from groundtruth poses, in these rooms already enjoyed good localizability scores (see Fig. 5(a)) 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 Section III-B in Python with assistance of a few open source software packages. We used the Unreal Engine 4.27 [22] and the AirSim library (v1.8.1) [23] to simulate and collect images from 3D models. We used the Open3D library [24] to downsample scan points to get candidate marker locations. We used the GTSAM library [25] to create factor graphs and estimate covariances in Gaussian approximations of camera pose distributions. The SIFT feature [18] 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 [15] 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 [17], while the third model is an Unreal Engine simulation environment that resembles typical realworld offices. The first three are for simulated experiments. The last one is a motion capture room at MIT for the real experiment [1]. The textured mesh of the room was created by fusing RGB-D images from groundtruth poses, in these rooms already enjoyed good localizability scores (see Fig. 5(a)) and (ii) a large hyperparameter $v$ does not emphasize area coverage.
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 I
SPECIFICS OF SCENES

| Model  | Area (m²) | # 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 |

using the volumetric fusion [26] and marching-cubes algorithms and the screened Poisson surface reconstruction [27]. 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 [28] but with an extra function of fiducial marker detection, provided by the ApritiTag library [5]. The tag detection and VLAD descriptors [29] were sequentially employed to find matched images in the map data. Camera poses were estimated using P3P [30] with RANSAC [31] followed by Levenberg-Marquardt optimization [32]. The rotation error $\delta_R$ is defined as the angular distance between the estimated rotation matrix $\hat{R}$ and the groundtruth rotation $R$ while the translation error $\delta_t$ is defined as the Euclidean distance between the estimated translation $\hat{t}$ and the groundtruth translation $t$, as seen in

$$\delta_R = \arccos\left(\frac{\text{tr}(\hat{R}^T R) - 1}{2}\right),$$

$$\delta_t = \|\hat{t} - t\|_2.$$

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. Let $\mathcal{L} = \{l(c) : c \in \mathcal{C}\}$ be the set of localizability scores of feasible camera poses in the ground plane space with no markers. The weights are defined as

$$\mathcal{W} = \{2l^* - \bar{l} - l(c) : c \in \mathcal{C}\},$$

where $l^*$ 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 [1].

V. RESULTS

We present two sections of results. In the first section, we present results comparing different marker placement methods.

6Results on test images uniformly sampled are available in [1].
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 real-world 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 Section 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. 10(b) 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. 10(a) 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. 9(c)). 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] Q. Huang, J. DeGol, V. Fragoso, S. N. Sinha, and J. J. Leonard, “Optimizing fiducial marker placement for improved visual localization,” 2023, arXiv:2211.01513.

[2] 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.

[3] R. Muñoz-Salinas and R. Medina-Carnicer, “UcoSLAM: Simultaneous localization and mapping by fusion of keypoint- and plane-based markers,” Pattern Recognit., vol. 101, May 2020, Art. no. 107193.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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., vol. 45, no. 3, pp. 2931–2944, Mar. 2023.

[9] 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.

[10] 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.

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

[12] M. P. Vitus and C. J. Tomlin, “Sensor placement for improved robotic navigation,” in Proc. Robot. Sci. Syst., 2011, pp. 217–224.

[13] D. B. Jourdan and N. Roy, “Optimal sensor placement for agent localization,” ACM Trans. Sensor Netw., vol. 4, no. 3, pp. 1–40, May 2008.

[14] M. Beinhefer, 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.

[15] D. Meyer-Delius, M. Beinhefer, 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.

[16] 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., vol. 70, no. 6, pp. 5994–6003, Jun. 2023.

[17] S. K. Ramakrishnan et al., “Habitat-matterport 3D dataset (HMD3): 1000 large-scale 3D environments for embodied AI,” in Proc. Conf. Neural Inform. Process. Syst. Dataset. Benchmark. Track (Round 2), 2021, pp. 1–21.

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

[19] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using Rao-blackwellized particle filters,” in Proc. Robot. Sci. Syst., 2005, vol. 2, pp. 65–72.

[20] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[21] 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.

[22] Epic Games, “Unreal engine,” Mar. 2023. [Online]. Available: https://www.unrealengine.com

[23] 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.

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

[25] F. Dellaert et al., “borglab/gtsam,” May 2022. [Online]. Available: https://doi.org/10.5281/zenodo.5794541

[26] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in Proc. Ann. Conf. Comput. Graph. Interact. Tech., 1996, pp. 303–312.

[27] M. Kaufman and H. Hoppe, “Screened poisson surface reconstruction,” ACM Trans. Graph., vol. 32, no. 3, pp. 1–13, 2013.

[28] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” in Proc. IEEE/ICCV Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12716–12725.

[29] H. Jegou, 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.

[30] 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.

[31] 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.

[32] G. Bradski, “The OpenCV library,” Dr Dobbs’ J. Softw. Tools, vol. 25, no. 11, pp. 120–123 2000.