Improved Structure from Motion Using Fiducial Marker Matching

Joseph DeGol[1(✉)], Timothy Bretl[1], and Derek Hoiem[1,2]

[1] University of Illinois, Urbana-Champaign, USA
{degol2,tbretl,dhoiem}@illinois.edu
[2] Reconstruct Inc., Champaign, USA
derek.hoiem@reconstructinc.com

Abstract. In this paper, we present an incremental structure from motion (SfM) algorithm that significantly outperforms existing algorithms when fiducial markers are present in the scene, and that matches the performance of existing algorithms when no markers are present. Our algorithm uses markers to limit potential incorrect image matches, change the order in which images are added to the reconstruction, and enforce new bundle adjustment constraints. To validate our algorithm, we introduce a new dataset with 16 image collections of large indoor scenes with challenging characteristics (e.g., blank hallways, glass facades, brick walls) and with markers placed throughout. We show that our algorithm produces complete, accurate reconstructions on all 16 image collections, most of which cause other algorithms to fail. Further, by selectively masking fiducial markers, we show that the presence of even a small number of markers can improve the results of our algorithm.

Keywords: Structure from motion · SFM · Fiducial markers
3D reconstruction · Simultaneous localization and mapping · SLAM

1 Introduction

Fiducial markers are often claimed to be useful for 3D reconstruction [1–7]. Markers provide highly detectable and identifiable features that 3D reconstruction can use to overcome challenging scene characteristics such as low-texture surfaces (e.g., blank walls), reflective surfaces (e.g., windows), and repetitive patterns (e.g., columns and door frames). Figure 1 shows an example of a dataset with exactly these challenging characteristics. Figure 1 also shows that approaches that treat markers as texture, only use them as additional tracks, or rely on them exclusively perform no better or even worse than if markers were ignored.

In this paper, we present an incremental structure from motion (SfM) algorithm that significantly outperforms these other approaches when markers are
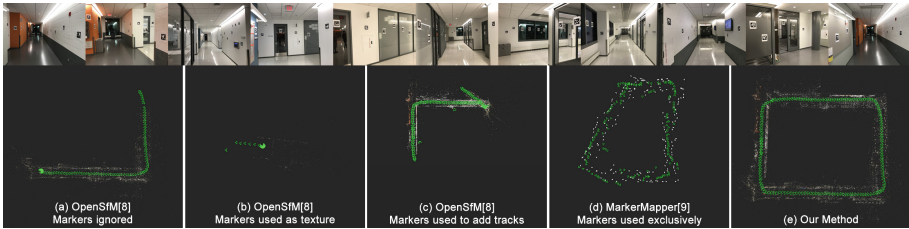
Fig. 1. We introduce a new dataset of unordered image collections of challenging indoor scenes with markers placed throughout (example images along top row). We process the data using OpenSfM [8] with (a) markers ignored, (b) markers used as texture, and (c) markers used as additional tracks; with (d) MarkerMapper [9], which uses markers exclusively; and with (e) our approach, which uses markers to limit image matches, dictate resectioning order, and constrain bundle adjustment. Clearly, our method (e) outperforms the others. Moreover, the other approaches often perform worse than ignoring the markers, highlighting the importance of our method.

present in the scene. We exploit that markers can be identified with very low false positive rates (e.g. AprilTag2 with 36h11 markers has a false positive rate of 0.000044% [2]) to create a reliable marker match graph that guides image matching and resectioning. We encode constraints on marker size, shape, and planarity in bundle adjustment to further improve results. Importantly, our approach benefits from any detected markers without sacrificing performance when markers are not detected, and can benefit from even a small number of markers.

To evaluate our method, we introduce a new dataset with 16 image collections of indoor scenes. The scenes present challenging circumstances for SfM (e.g. blank hallways, reflective glass facades, and repetitive brick walls). Each indoor scene has tens to hundreds (depending on scene size) of markers placed approximately uniformly throughout. We test our system and several cutting edge benchmarks on this data and show that our system performs favorably. We also selectively mask markers and show that performance gracefully degrades towards markerless SfM as the number of markers in the scene decreases.

In summary, the **contributions** of this paper are: (1) an SfM algorithm that uses both fiducial markers (when available) and interest point features for improved results; (2) a large, challenging dataset of indoor scenes with markers placed throughout; and (3) experiments showing the effectiveness of our approach, even when only a small number of markers are visible.

2   Related Work

Incremental SfM: Early works by Schaffalitzky and Zisserman [10] and Snavely et al. [11] establish the pipeline for feature extraction, matching, and incremental SfM for unordered image collections. Focus then turns to large image collections with work by Agarwal et al. [12] and Frahm et al. [13] who use appearance based clustering to limit potential image matches; enabling reconstructions of

Fig. 2. Example images from the Neunert et al. [16] dataset: desk (top left), dataset1 (top right), cube (bottom left), and pavilion (bottom right). Experiments in Sect. 5 show that our method and current SfM methods perform well on this data, motivating our new dataset that offers new challenges and better distinguishes between approaches.

Rome from thousands of internet photos. Work by Wu [14] shows that preemptive feature matching and well timed global bundle adjustments can maintain high accuracy while reducing the runtime of SfM to roughly $\mathcal{O}(n)$. Recently, several new SfM algorithms are available including COLMAP [15] by Schönberger and Frahm and OpenSfM [8] by Mapillary. These impressive works provide the baseline for the work in this paper.

3D Reconstruction Using Fiducial Markers: Early works using markers for 3D reconstruction focus on tracking the markers in simultaneous localization and mapping (SLAM) systems. Work by Klopschitz and Schmalstieg [17] tracks both feature points and marker matches in video frames to estimate the camera pose and triangulate the marker positions in 3D. Lim and Lee [18] and Yamada et al. [19] add an extended kalman filter (EKF) for estimating robot camera pose and marker positions in 3D. Neunert et al. [16] integrates IMU measurements into the EKF-SLAM system to improve pose estimates during marker tracking. Feng et al. [20] proposes an incremental SfM approach to marker based 3D reconstruction. They use markers to create an initial reconstruction, add new images using marker matches, and add constraints to bundle adjustment to enforce the square shape and planarity of markers. The work of Muñoz-Salinas et al. [9] introduces MarkerMapper. MarkerMapper overcomes the pose ambiguity problem [21] with planar marker pose estimation to create an initial proposal of 3D camera and marker locations and refines the proposal using global bundle adjustment. Only MarkerMapper [9] and Feng et al. [20] pursue 3D reconstruction from unordered image collections. However, neither method uses both image features and marker detections for 3D reconstruction. Experiments in Sect. 5 show that both image features and marker detections can be used together to achieve the best results, and, when few or no markers are available, our system performs no worse than non-marker based SfM.

Datasets: Datasets for testing marker based 3D reconstruction are limited. Only the dataset of Neunert et al. [16] is publicly available. Figure 2 provides snapshots from the four video sequences of this dataset. With only four sequences (two of which are of very small environments with only 1–3 markers), this dataset is no longer challenging for the current state of the art (e.g. in Sect. 5, we process this data with our method and other current SfM approaches, and all perform well).
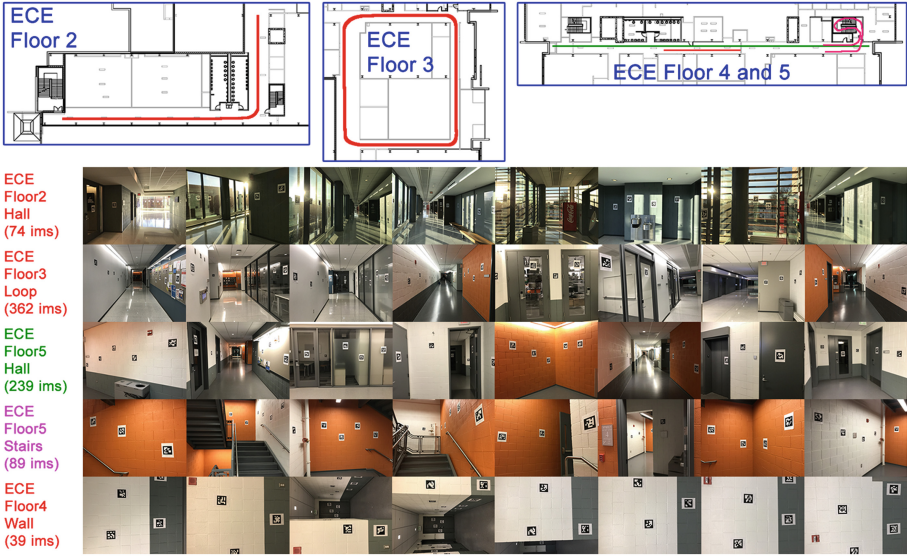
Fig. 3. The top diagrams are floor plans of ECE. The paths for image collection are superimposed in red, green, and magenta. These colors correspond to the image set name and example images. For example, *ECE Floor5 Stairs* is shown in the *ECE Floor4 and 5* floor plan as a magenta line and the name with example images is also magenta. (Color figure online)

Our new dataset (Sect. 3) consists of 16 new image collections in environments with challenging characteristics for SfM (e.g. many low-texture walls and reflective glass). We hope our dataset will offer new challenges for future work on SfM both with and without marker assistance.

3   Indoor Image Collections with Fiducial Markers

We introduce 16 new unordered image sets for evaluating structure from motion for scenes containing fiducial markers. Each set is from one of three buildings: ECE, CEE, or MUF. Figures 3 and 4 provide floor plans for the sections of these buildings that are used to collect this data. Paths are drawn on each floor plan and the colors of the paths match the respective image sets in the figures (e.g. the green path on Floor 4 and 5 of ECE matches the ECE Floor5 Hall image set). For each set, fiducial markers are placed around the scene with enough density to see at least one in every image (and images are captured to satisfy this also). All images are captured with an iPhone7 camera and have a resolution of 4032 × 3024 pixels.

There are seven image sets not shown in the figures. That is because they are either combinations or subsets of the shown sets. Specifically, *ECE Floor5* includes all the images of *ECE Floor5 Hall* and *ECE Floor5 Stairs*. *ECE Floor3*
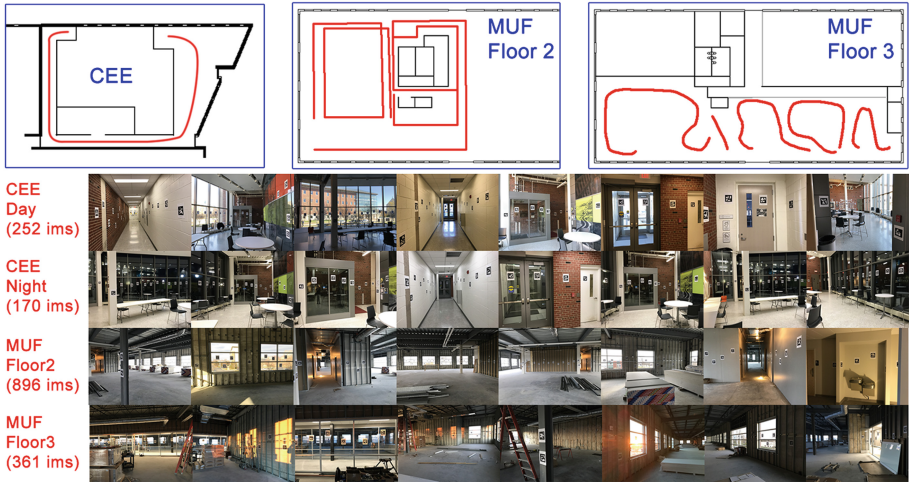
Fig. 4. The top diagrams are floor plans for CEE and MUF. The paths for image collection are superimposed in red. Image set names and example images are shown. (Color figure online)

Loop includes all the images of *ECE Floor3 Loop CW* and *ECE Floor 3 CCW*. *CEE Day* includes all the images of *CEE Day CW* and *CEE Day CCW* (plus some extra images). The nice thing about collecting data in this way is that we can test progressively larger datasets that present different circumstances that may make the image set easier or more difficult. For example, the results in Sect. 5 show that *ECE Floor3 Loop CW* and *ECE Floor3 Loop CCW* are typically more difficult than putting them together into *ECE Floor 3 Loop*. This is most likely because of the additional overlap between images since all locations are now seen more often from more viewing directions.

We use ECE, CEE, and MUF because they are large indoor scenes with characteristics that are challenging for SfM (as shown in Sect. 5). Specifically, ECE has long plain hallways, large glass walls separating conference rooms, large exterior windows, and the hallways form a loop. CEE has a two-floor glass facade and repetitive brick walls. MUF is currently under construction and has large open spaces and limited texture. See supplementary material for more examples.

4   Improving SfM with Markers

Figure 5 diagrams our marker assisted incremental SfM algorithm. The blue boxes represent the components of our algorithm that are different from typical state of the art incremental SfM approaches: detecting markers, filtering image pairs, resectioning images, and marker constraints for bundle adjustment.
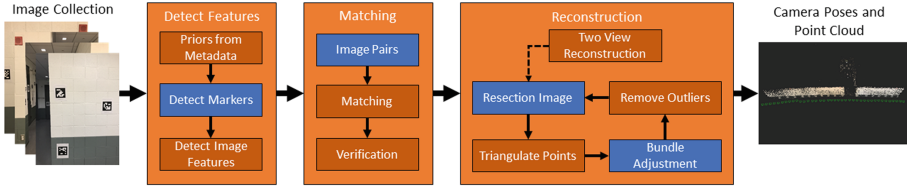
Fig. 5. This diagram depicts the typical incremental SfM approach: extracting priors from metadata (e.g. focal length), detecting features, matching features, and reconstruction. The blue boxes are the areas we added or changed in our method. (Color figure online)

4.1 Incremental SfM Overview

Incremental SfM takes a collection of images as input. For each image, focal length (and other priors) is estimated from metadata (or using heuristics when metadata is unavailable). Next, image features (e.g. SIFT features [22]) are extracted from each image. These image features are matched across image pairs. Matching is attempted between the set of all images pairs or a subset of the image pairs selected based on filtering criteria (e.g. GPS locations [13], Vocab Tree [12]). A fundamental matrix is estimated from the feature matches to filter bad matches and verify that each image pair is a good match.

After matching, reconstruction begins. Feature matches in two images are used to create an initial 3D reconstruction (pose of the two images with triangulated 3D points). Then, one at a time, a new image is added to the reconstruction (resectioning). This image is typically chosen based on the number of feature matches this image shares with the already reconstructed images. These shared feature matches are used to estimate the pose of this new camera and triangulate new 3D points. Bundle adjustment then optimizes all camera poses and 3D point positions to minimize reprojection error. Lastly, outlier points are removed. Resectioning is repeated to add all images to the reconstruction. The final output is a point cloud and set of camera poses, one for each image that is successfully resectioned.

4.2 Detect Markers

We run a square marker detection algorithm on each input image. The images are processed in parallel. Image name, marker id, corner locations, and corner pixel colors are saved for each detection.

4.3 Marker Informed Image Pairs

Prior to matching and verification, we create a set of image pairs that potentially match. We only attempt matching on the image pairs in this set. One approach is to add all possible image pairs; however, this greatly increases matching time and can lead to bad image matches that cause errors in the reconstruction. We
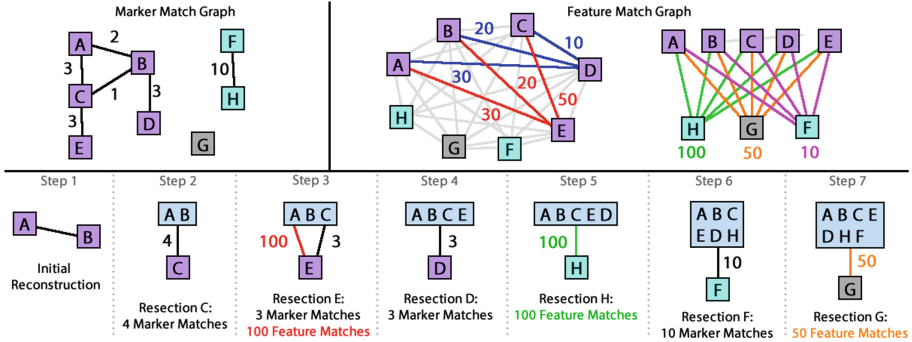
Fig. 6. The top left diagram depicts images as lettered boxes with edges representing the number of matched markers between image pairs. The top middle and top right diagrams depict the number of common feature matches between images. The bottom diagram depicts the resectioning order of images A to G based on two rules: (1) add the image that shares the most marker matches with the reconstruction; (2) break ties using most shared feature matches.

apply three rules to use marker detections to dictate which images are added. **Rule 1:** we add an image pair if the same marker (at least one) is detected in both images. **Rule 2:** if an image does not share a detected marker with any other image, we add all possible pairs that contain that image. **Rule 3:** if the set of all added pairs do not form one connected component, we connect separate components by adding pairs for each image in the separate component to each image not in the separate component.

As an example, consider the top left diagram in Fig. 6. Each lettered box represents an image, and each numbered edge represents the number of marker matches those images share. Applying rule 1, we add the following possible image pairs (A, B), (A, C), (B, C), (B, D), (C, E), and (F, H). No pair is added that includes G, so based on rule 2, we add (G, A), (G, B), ..., (G, H). Lastly, since (F, H) is a separate component (rule 3), we add (F, A), (F, B), ..., (F, E) and (H, A), (H, B), ..., (H, E). We show in the results that this strategy can greatly speed up processing and eliminate many bad image matches. Note that other filtering approaches (e.g. Vocab Tree [12]) can be used in conjunction with our approach to add or filter image pairs.

4.4 Marker Informed Resectioning

Resectioning is the process of adding a new image to the existing reconstruction. The order in which images are added is important because poorly registered images can propagate errors that result in failure. One approach is to choose the image to resection that shares the most feature matches with the images in the reconstruction. This approach works well when image features are distinct and plentiful; however, for the challenging scenes we are targeting, failure can occur. Instead, we apply two rules to use marker detections to dictate resectioning order.

Rule 1: the next image to resection shares the most marker matches with the current reconstruction. **Rule 2:** if multiple images share the same number of marker matches with the current reconstruction, choose the image that shares the most feature matches.

For example, consider the diagrams in Fig. 6. In the top left diagram, each edge represents the number of marker matches those images share. In the top middle and top right diagram, each numbered edge represents the number of image feature matches those images share. The bottom diagram depicts the resectioning procedure. First, images A and B are used for the initial reconstruction (step 1). The next image that is resectioned is C because it shares 4 (3 with A and 1 with B) marker matches with the current reconstruction (step 2). After that, image E is added because E and D both share 3 marker matches with the reconstruction, but E shares 100 feature matches and D only shares 60 (step 3). Image D is then added (step 4). No remaining images share marker matches with the current reconstruction, so image H is added based on shared image feature matches (step 5). F is added next (step 6) because it now shares marker matches with the reconstruction (because H was added). Lastly, G is added (step 7).

4.5   Marker Constraints for Bundle Adjustment

In bundle adjustment, we solve for camera poses $\boldsymbol{P}$ and 3D points $\boldsymbol{X}$ that optimize the following:

$$\min_{\boldsymbol{P},\boldsymbol{X}} \left[w_R E_R \left(\boldsymbol{P}, \boldsymbol{X} \right) + w_S E_S \left(\boldsymbol{V} \right) + w_O E_O \left(\boldsymbol{V} \right) \right] . \tag{1}$$

$\boldsymbol{V}$ is the set of vectors formed between neighboring 3D corners on each marker (i.e. there are four vectors for each marker detection). w_R, w_S, and w_O are weights. Reprojection error [23] is

$$E_R \left(\boldsymbol{P}, \boldsymbol{X} \right) = \sum_{i=1}^{C} \sum_{j=1}^{N} L \left(x_{ij}, \boldsymbol{P}_i \left(\boldsymbol{X}^j \right) \right) \tag{2}$$

where C is the number of cameras, N is the number of 3D points (both marker and feature points), L is a loss function, x_{ij} is the 2D location in image i of 3D point $\boldsymbol{X}^j$, and $\boldsymbol{P}_i$ is the projection function of camera i. Similar to [20], we also include error terms for marker scale (E_S, Eq. 3) and marker orthogonality (E_O, Eq. 4).

Marker Scale: The distance between marker corners in the reconstruction should match the known marker size. We define this error as $E_S \left(\boldsymbol{V} \right) =$

$$\sum_{i=1}^{T} \left(\left\| \boldsymbol{V}_{12}^i \right\|_2 - S \right)^2 + \left(\left\| \boldsymbol{V}_{23}^i \right\|_2 - S \right)^2 + \left(\left\| \boldsymbol{V}_{34}^i \right\|_2 - S \right)^2 + \left(\left\| \boldsymbol{V}_{41}^i \right\|_2 - S \right)^2 \tag{3}$$

where $\boldsymbol{V}_{NM}^i$ is the 3D vector from the 3D point of corner N to the 3D point of corner M on marker i, T is the number of markers, and S is the marker size.

Marker Orthogonality: Adjacent sides of the marker should be perpendicular. We define this error as $E_O(\boldsymbol{V}) =$

$$\sum_{i=1}^{T} \left(\boldsymbol{V}_{12}^i \cdot \boldsymbol{V}_{23}^i\right)^2 + \left(\boldsymbol{V}_{23}^i \cdot \boldsymbol{V}_{34}^i\right)^2 + \left(\boldsymbol{V}_{34}^i \cdot \boldsymbol{V}_{41}^i\right)^2 + \left(\boldsymbol{V}_{41}^i \cdot \boldsymbol{V}_{12}^i\right)^2 \qquad (4)$$

where $\boldsymbol{V}_{NM}^i$ is the 3D vector from the 3D point of corner N to the 3D point of corner M on marker i, and T is the number of markers.

4.6   Implementation Details

We implement our approach on top of OpenSfM v0.1.0 [8]. We use default parameters, which work well for unordered image collections. We use AprilTag2 [2] to detect markers. For all experiments, we use a soft L1 loss for L; cost weights of $w_R = 62500$, $w_S = 100$, and $w_O = 100$; and marker size $S = 0.21$ m. In principal, our approach works with any square marker detector and can be integrated with any incremental or global [24,25] (except resectioning) SfM method.

5   Results

We process our new dataset using: (1) OpenSfM [8], an open source state of the art SfM algorithm that is actively used and maintained by Mapillary [26]; (2) OpenSfM, but with all feature points on markers masked; (3) MarkerMapper [9], a state of the art algorithm for marker based SfM; (4) OpenSfM with the four marker corners used as tracks in reconstruction; and (5) our method. Table 1 provides quantitative results on the number of images localized, number of points, and reprojection errors. Failure reconstructions are denoted by a "-". Figures 7 and 8 provide qualitative results of the 3D reconstructions. The green pyramids are the camera locations. The floor plans in Figs. 3 and 4 provide guidelines for how each reconstruction should look (e.g. ECE Floor3 Loop should be a rectangle). Because of the challenging nature of these datasets, the algorithms often fail or have large, noticeable mistakes; therefore, we focus more on the qualitative results because they illustrate the improvements clearly.

We also process the Neunert et al. [16] dataset. Since it is video data, we subsample the frames by a factor of 5 to simulate an unordered image collection. All OpenSfM methods and our method successfully reconstruct all image sets. MarkerMapper has trouble with this dataset because there are few (often only one) markers in each image. Reconstruction and timing results are shown in Tables 1 and 2 respectively. Qualitative results are in the supplementary material.

We do an ablation study with marker informed matching (Sect. 4.3) and marker informed resectioning (Sect. 4.4). For each dataset and method we calculate the percent of images localized. The average percentages of localized images are 98% (our full method), 68% (no marker informed resectioning), 50% (no marker informed matching), and 42% (OpenSfM with markers masked — the next best method). These percentages show that both marker informed matching

Table 1. Reconstruction results for OpenSfM [8], OpenSfM with markers masked (denoted by [8]*), MarkerMapper [9], OpenSfM with marker tracks (denoted by MT), and our method. Failure reconstructions (Figs. 7 and 8) are blank because the numbers can be misleading (e.g. all cameras localized to one spot). Our method achieves similar or better results for number of registered images and points for all reconstructions.

	# Images	# Registered					# Points					Avg. Rep. Error [px]				
		[8]	[8]*	[9]	MT	Ours	[8]	[8]*	[9]	MT	Ours	[8]	[8]*	[9]	MT	Ours
ECE F2 Hall	74	-	70	-	-	71	-	15.9K	-	-	16.4K	-	3.1	-	-	2.8
ECE F3 Loop CCW	192	-	-	190	-	191	-	-	808	-	61K	-	-	200.8	-	2.8
ECE F3 Loop CW	170	-	-	166	-	170	-	-	736	-	58K	-	-	358.1	-	2.7
ECE F3 Loop	362	-	-	356	-	360	-	-	920	-	105K	-	-	324.0	-	2.8
ECE F5 Hall	239	230	230	213	223	231	50K	45K	736	47K	63K	2.8	2.7	141.0	2.7	2.7
ECE F5 Stairs	89	52	51	-	45	89	20K	20K	-	14K	43K	1.9	1.7	-	1.9	1.8
ECE F5	328	313	315	-	-	327	79K	73K	-	-	109K	2.3	2.3	-	-	2.3
ECE F4 Wall	39	21	18	39	18	39	13K	9K	204	9K	28K	1.1	1.1	25.8	1.2	1.2
CEE Day CW	63	55	52	-	52	62	24K	20K	-	28K	30K	1.6	1.6	-	1.6	1.6
CEE Day CCW	120	65	116	-	116	119	30K	52K	-	56K	61K	1.6	1.5	-	1.6	1.5
CEE Day	252	-	251	238	103	246	-	89K	768	398	104K	-	1.7	204.8	0.2	1.8
CEE Night CW	96	96	96	96	-	96	48K	44K	548	-	51K	1.7	1.6	164.0	-	1.7
CEE Night CCW	79	-	-	79	-	77	-	-	580	-	40K	-	-	116.6	-	1.5
CEE Night	170	-	166	170	-	170	-	61K	760	-	77K	-	1.6	181.4	-	1.6
MUF F2	896	883	514	-	885	882	224K	133K	-	151K	251K	2.5	2.5	-	2.1	2.9
MUF F3	361	343	-	-	324	358	84K	-	-	55K	89K	2.8	-	-	2.4	2.8
cube [16]	327	327	327	-	327	327	99K	101K	-	100K	99K	0.8	0.8	-	0.8	0.8
dataset1 [16]	91	91	91	3	91	91	31K	30K	8	31K	33K	0.9	0.9	0.6	0.9	0.8
pavilion [16]	585	585	585	-	585	583	178K	168K	-	186K	178K	0.8	0.7	-	0.7	0.7
table [16]	80	80	49	38	80	80	7K	5K	12	7K	6K	0.9	1.0	0.3	0.9	1.0

and resectioning are useful individually, but most effective when used together. We also test our method without marker scale (E_S, Eq. 3) and orthogonality (E_O, Eq. 4) constraints and find that they provide little to no gain, sometimes making the results worse. See supplementary material for more details about the ablation study.

All experiments use an Intel Xeon E5-2620 V4 2.1 GHz 16 cores (32 virtual cores) processor with 128 GB of RAM. No graphics card is used.

Using Markers as Texture Often Makes Reconstructions Worse. Masking the markers shows how OpenSfM performs if the scenes have no markers. Comparing column 1 (OpenSfM) and column 2 (OpenSfM with masked markers) in Figs. 7 and 8, shows that masking the markers often produces better results. For example, *ECE Floor2 Hall* should have an "L" shape, which OpenSfM with masked markers achieves, but OpenSfM does not. Other examples where masking markers is clearly better are *ECE Floor3 Loop CW*, *ECE Floor5 Stairs*, *CEE Day CCW*, *CEE Day*, and *CEE Night*.

Marker texture does not always produce bad results (e.g. *MUF Floor3*), but marker texture can cause bad feature matches because the appearance is similar between the markers (i.e. black and white squares). This reinforces the need for our approach which takes advantage of visible markers to improve results.

Using Marker Detections as Tracks Has Little Effect. Comparing column 4 (OpenSfM with marker tracks) to columns 1 and 2 (OpenSfM with and without markers masked) of Figs. 3 and 4 shows that the marker tracks rarely improve
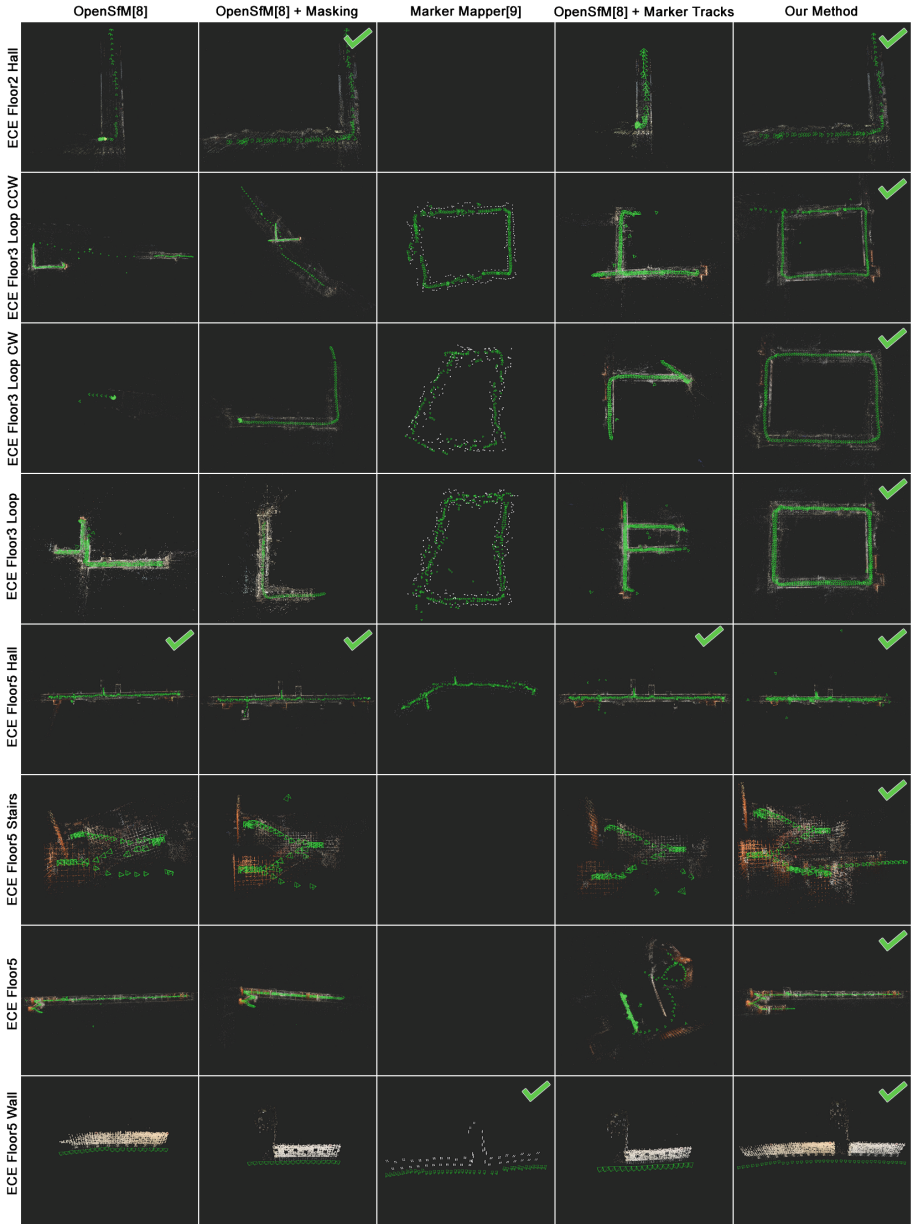
Fig. 7. Reconstructions for OpenSfM, OpenSfM with markers masked, MarkerMapper, OpenSfM with marker tracks, and our method on the ECE image collections. Using the markers as texture often produces worse results (e.g. *ECE Floor2 Hall*, *ECE Floor3 Loop CW*, *ECE Floor3 Loop*, and *ECE Floor5 Stairs*). Our method produces complete reconstructions that are as good or better than the other methods for all image collections. The best results are denoted by a green check mark. (Color figure online)
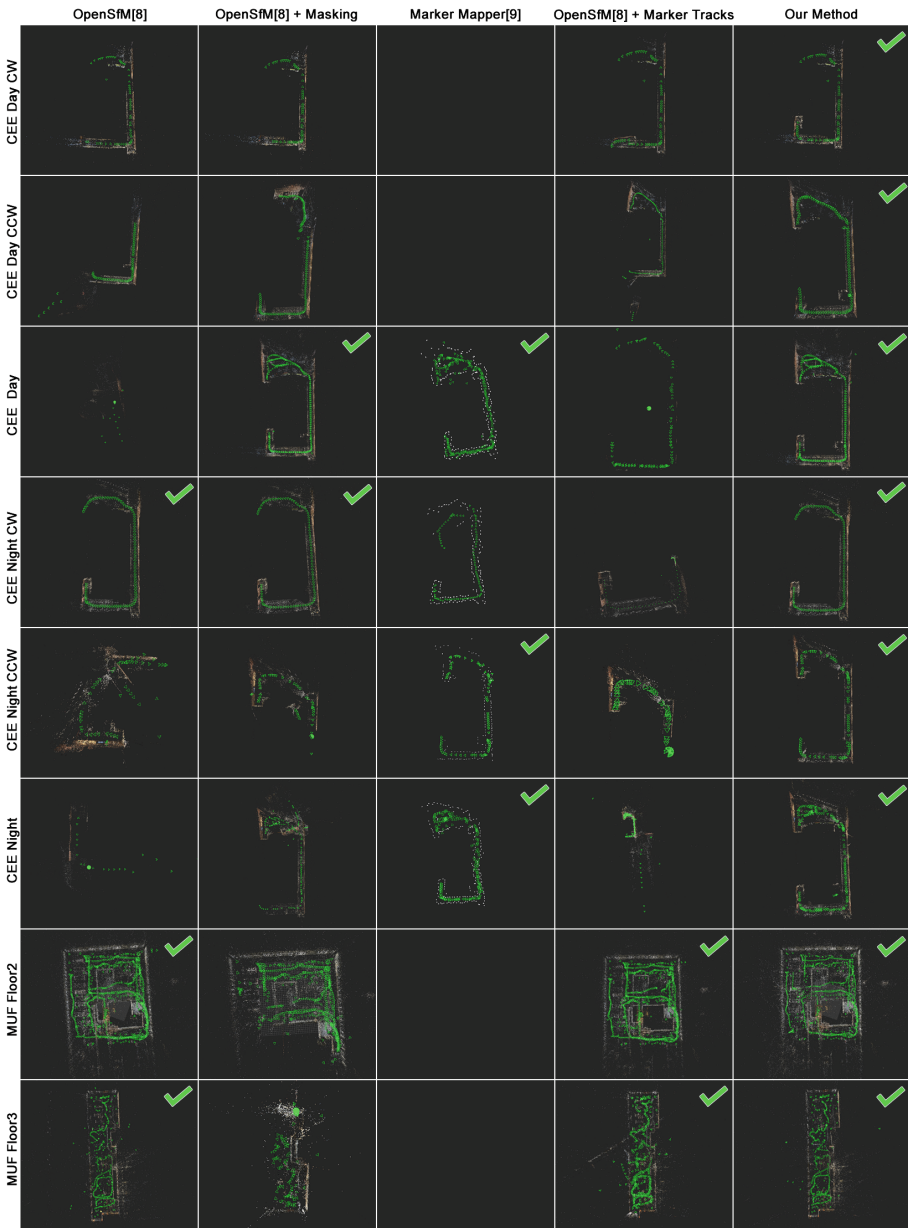
Fig. 8. Reconstructions for OpenSfM, OpenSfM with markers masked, MarkerMapper, OpenSfM with marker tracks, and our method on the CEE and MUF image collections. Again, using the markers as texture often produces worse results (e.g. *CEE Day CCW*, *CEE Day*, and *CEE Night*). Our method produces complete reconstructions that are as good or better than the other methods for all image collections. The best results are denoted by a green check mark. (Color figure online)

Table 2. Reconstruction timings for OpenSfM [8], OpenSfM with markers masked (denoted by [8]*), MarkerMapper [9], OpenSfM with marker tracks (denoted by MT), and our method. Using the markers to limit possible image pairs decreases the matching time significantly. Also, because more images are resectioned, the reconstruction time increases. Overall, our method produces better reconstructions in a shorter time.

	# Images	Marker Detection [s]			Matching [s]				Reconstruction [s]				
		[8]	[8]*	MT	Ours	[8]	[8]*	MT	Ours	[8]	[8]*	[9]	MT	Ours
ECE F2	74	0	14	14	14	215	186	223	73	331	222	-	363	277
ECE F3 Loop CCW	192	0	32	32	32	1356	1160	1398	293	3433	2766	85	2282	3097
ECE F3 Loop CW	170	0	30	30	30	1071	888	1152	273	2797	2084	83	2430	2367
ECE F3 Loop	362	0	59	59	59	4568	3820	4675	876	9944	5082	195	9239	9704
ECE F5 Hall	239	0	40	40	40	1955	1650	1974	296	2810	2363	80	2774	3061
ECE F5 Stairs	89	0	16	16	16	307	258	317	55	425	278	-	347	658
ECE F5	328	0	57	57	57	3787	3195	3945	372	6341	5083	-	7268	5513
ECE F4 Wall	39	0	9	9	9	61	46	47	8	133	46	22	63	263
CEE Day CW	63	0	11	11	11	160	126	171	49	336	216	-	489	382
CEE Day CCW	120	0	21	21	21	535	437	570	139	1011	437	-	2305	1809
CEE Day	252	0	41	41	41	2373	1919	2567	440	7137	5252	148	4102	4987
CEE Night CW	96	0	16	16	16	358	278	380	99	1083	793	25	1136	1010
CEE Night CCW	79	0	14	14	14	247	193	70	69	425	418	32	917	654
CEE Night	170	0	30	30	30	1093	873	1154	216	3232	2251	93	3287	2984
MUF F2	896	0	158	158	158	31180	25613	35844	5596	72055	40958	-	66095	60542
MUF F3	361	0	64	64	64	5094	4302	5205	758	8977	6903	-	5017	9090
cube [16]	327	0	6	6	6	3473	2724	2776	423	4066	5232	-	4244	4134
dataset1 [16]	91	0	1	1	1	351	305	304	207	847	593	1	619	595
pavilion [16]	585	0	7	7	7	9103	9239	9219	2100	22908	15931	-	21903	22594
table [16]	80	0	1	1	1	64	60	63	65	113	188	2	205	191

the reconstructions and sometimes make them worse (e.g. *ECE Floor5* and *CEE Night*). We suspect this is because the localization of the marker corners can be less accurate (e.g. off by 3–5 pixels [3]) than image features.

Our Approach Succeeds Where Others Fail. From Figs. 7 and 8, we see that our method produces a successful reconstruction for every image set. We also see that our method produces better results than the other methods on the challenging sets. Most notable are *ECE Floor3 Loop CW*, *ECE Floor3 Loop*, and *CEE Day CCW* because all other methods fail or have significant mistakes. For *ECE Floor5 stairs*, *ECE Floor5*, and *CEE Day CW*, other methods produce reasonable results, but our approach is more complete.

Our Approach Succeeds Where Others Succeed. There are several image sets where all (or most) of the methods produce successful reconstructions (e.g. *ECE Floor5 Hall* and *CEE Night CW*). In these cases, our method also produces nice reconstructions. This is important because our algorithm improves on the challenging image sets without sacrificing accuracy on the easier image sets.

Using Markers Improves Reconstruction Time. Table 2 provides the run times for marker detection, matching, and reconstruction for all image sets. Timings for other parts of SfM are not included since they do not change between methods. Also, only the total run time of MarkerMapper is reported because it does not follow the same pipeline as the others. One main thing to note is that using markers to limit pairs for image matching can decrease run times

significantly (e.g. for *MUF Floor2*, our method took 5596 seconds and the other OpenSfM approaches took 5–6 times longer). Time is added to detect markers in each image, but it is typically negligible compared to the time saved in matching. Another interesting point is that reconstruction time often increases. This is because more images are able to be registered with our method.

Few Marker Detections Still Improves Reconstructions. Figure 9 demonstrates how marker density effects the reconstructions. In particular, the left six images show how the reconstruction of *ECE Floor3 Loop CCW* improves as the marker density increases. Here AMD stands for average marker detections per image (e.g. AMD = 0.0 means there are no markers detected, and AMD = 6.0 means that there were an average of 6 markers detected per image).

The plot in Fig. 9 shows how the percent of localized images increases as the AMD increases for seven datasets. These datasets were chosen because our method achieves clear improvements over the other methods. The trend line is plotted in black. We see from this plot that markers help even when AMD is less than 1 (sometimes even 100% of the images are localized). As AMD increases, the number of localized images increases towards 100%. Placing enough markers for an AMD of 6 will likely produce accurate, complete reconstructions with 90+% of images localized. However, markers are most useful in areas with challenging conditions for SfM, so placing more markers in these challenging areas and fewer (or zero) markers in easier areas can help our method achieve accurate, complete reconstructions with drastically fewer total marker detections.
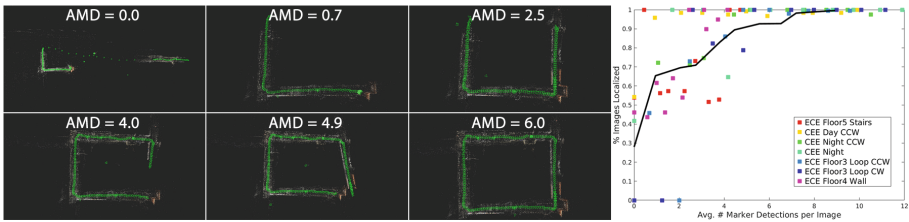

Fig. 9. The left six images show how the reconstruction of *ECE Floor3 Loop CCW* improves as marker density increases. Here AMD means average marker detections per image. The right image shows the percent of images localized as the AMD increases. Each color represents a different dataset. The trend line is shown in black. As the AMD increases, the percent of localized images increased to 100%. (Color figure online)

6   Conclusion

We present an incremental SfM method that significantly outperforms existing methods when fiducial markers are detected in the scene. We introduce a new dataset with 16 image collections of indoor scenes with square markers placed throughout. We use the unique marker IDs to improve image matching and resectioning order and show that these improvements greatly improve reconstruction

results when compared to other methods. Lastly, we show that even a small number of visible markers often improves reconstruction results.

References

1. Birdal, T., Dobryden, I., Ilic, S.: X-tag: a fiducial tag for flexible and accurate bundle adjustment. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 556–564, October 2016
2. Wang, J., Olson, E.: AprilTag 2: efficient and robust fiducial detection. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2016
3. DeGol, J., Bretl, T., Hoiem, D.: ChromaTag: a colored marker and fast detection algorithm. In: ICCV (2017)
4. Garrido-Jurado, S., noz Salinas, R.M., Madrid-Cuevas, F., Marín-Jiménez, M.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn. **47**(6), 2280–2292 (2014)
5. Fiala, M.: Designing highly reliable fiducial markers. IEEE Trans. Pattern Anal. Mach. Intell. **32**(7), 1317–1324 (2010)
6. Bergamasco, F., Albarelli, A., Cosmo, L., Rodola, E., Torsello, A.: An accurate and robust artificial marker based on cyclic codes. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2016)
7. Calvet, L., Gurdjos, P., Griwodz, C., Gasparini, S.: Detection and accurate localization of circular fiducials under highly challenging conditions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
8. OpenSfM. https://github.com/mapillary/opensfm
9. Muoz-Salinas, R., Marn-Jimenez, M.J., Yeguas-Bolivar, E., Medina-Carnicer, R.: Mapping and localization from planar markers. Pattern Recogn. **73**, 158–171 (2018)
10. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "How Do I Organize My Holiday Snaps?". In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 414–431. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47969-4_28
11. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: Proceedings of ACM SIGGRAPH (2006)
12. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a day. In: IEEE 12th International Conference on Computer Vision, pp. 72–79, September 2009
13. Frahm, J.-M., et al.: Building Rome on a cloudless day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_27
14. Wu, C.: Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision - 3DV 2013 (2013)
15. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

16. Neunert, M., Bloesch, M., Buchli, J.: An open source, fiducial based, visual-inertial motion capture system. In: 2016 19th International Conference on Information Fusion (FUSION) (2016)

17. Klopschitz, M., Schmalstieg, D.: Automatic reconstruction of wide-area fiducial marker models. In: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (2007)

18. Lim, H., Lee, Y.S.: Real-time single camera slam using fiducial markers. In: 2009 ICCAS-SICE (2009)

19. Yamada, T., Yairi, T., Bener, S.H., Machida, K.: A study on slam for indoor blimp with visual markers. In: ICCAS-SICE 2009, pp. 647–652 (2009)

20. Feng, C., Kamat, V., Menassa, C.C.: Marker-assisted structure from motion for 3D environment modeling and object pose estimation. In: Construction Research Congress (2016)

21. Schweighofer, G., Pinz, A.: Robust pose estimation from a planar target. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2024–2030 (2006)

22. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)

23. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN 0521540518

24. Moulon, P., Monasse, P., Marlet, R., et al.: OpenMVG. An open multiple view geometry library. https://github.com/openMVG/openMVG

25. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: 2013 IEEE International Conference on Computer Vision (2013)

26. Mapillary. https://www.mapillary.com/