Marker Assisted Structure from Motion for 3D Environment Modeling and Object Pose Estimation

Chen FENG[1], Vineet R. KAMAT[2], and Carol C. MENASSA[3]

[1] Ph.D., Dept. of Civil and Environmental Engineering, University of Michigan, 2350 Hayward St., Ann Arbor, MI 48109. *Corresponding Author: cforrest@umich.edu, Tel: +1-734-764-4325.

[2] Professor, Dept. of Civil and Environmental Engineering, University of Michigan, 2350 Hayward St., Ann Arbor, MI 48109; email: vkamat@umich.edu.

[3] Assistant Professor, Dept. of Civil and Environmental Engineering, University of Michigan, 2350 Hayward St., Ann Arbor, MI 48109; email: menassa@umich.edu.

ABSTRACT

Accurately modeling as-built environments and tracking moving objects' poses are critical for many Architecture, Engineering, Construction, and Facility Management (AECFM) automation applications. Equally important are the reliability, operating range and cost efficiency of such solutions for their broad deployment in unstructured, dynamic, and sometimes featureless AECFM sites. In this paper, a flexible vision-based technique is developed for accurate, robust, low-cost, and scalable pose estimation and as-built modeling in AECFM applications. This technique combines marker-based pose estimation and structure-from-motion (SfM). In the preparation phase, a sparse set of visual markers are installed in the target environment. During the operation phase, a set of unordered images are taken with a calibrated RGB camera. These images are immediately processed by a SfM system to estimate those markers' poses and generate a sparse point cloud, which can be used by robots or other mobile clients for either moving objects' pose estimation, or dimensional analysis of that environment. Furthermore, for as-built modeling, the RGB camera is replaced by a RGBD camera to create both a dense 3D point cloud and a concise planar model of the environment. Experiments have demonstrated sufficient accuracy (average absolute error within 5mm over a 9m scale) of the proposed technique.

INTRODUCTION

3D geometric modeling in either construction sites or built environment has attracted increasing research interests in AECFM due to its importance for various construction and maintenance activities, such as as-built documentation, interior design and facility management. No matter what sensors are used for such modeling, a fundamental step is to find out different sensor poses (positions and orientations) in a same coordinate frame so as to reach a unified and meaningful result from raw data captured under different local coordinate frames of sensors. This is closely related to object pose estimation, another core problem appearing in many AECFM automation applications, such as safety and productivity monitoring of construction machinery.

Among different technologies for 3D modeling or pose estimation, computer vision based methods have been introduced and investigated recently for potential construction applications. Whether the end result is a 3D model or an object's pose,

these methods operate based on identification of same physical elements' corresponding images (e.g., points, lines, planes, objects, etc.) across different camera views. Only with enough accurate and robust correspondences, can these methods estimate camera poses relative to each other, or triangulate 3D models across views. The assumption for this correspondence identification process (termed feature correspondence problem in computer vision) to be possible and reliable is that the target scene should be rich in locally distinguishable features that can be captured by cameras. So this assumption becomes the implicit condition of successful application of these computer vision based methods. Yet many industrial environments, such as indoor construction sites before finishing, often do not satisfy such assumption. They (e.g., walls, floors, ceilings) are frequently featureless or texture-less, or with repeated features, which particularly increases the difficulties for reliably, accurately, and efficiently solving the feature correspondence problem.

To address this challenge, the *camera marker network* has been proposed (Feng et al. 2015) by adding necessary fiducial markers in such featureless environments, which forms an observation system with multiple cameras and markers for estimating poses of objects attached with those markers or cameras. These fiducial markers not only resolve the reliability and accuracy issue of that feature correspondence problem, but also improve the time-efficiency to enable real-time automation applications such as excavation monitoring. It is worth to note that in the original proposed camera marker network method, the poses of a set of fixed markers need to be calibrated by conventional surveying so that any camera's pose can be linked to the world coordinate frame when observing these markers. However, this surveying requirement could be costly and slow, especially when the number of markers increases, or some fixed markers' poses need frequent update.

Thus in this paper, the marker assisted structure from motion is proposed to relax the abovementioned requirement of surveying poses of every fixed markers in a camera marker network. This technique combines marker-based pose estimation and the well-known structure from motion (SfM) in computer vision. SfM essentially means to recover the geometric structure of a scene by moving cameras and observing the scene from different poses. Similarly, when observing those fixed markers from different poses, one can recover their relative poses, thus replacing the need of surveying with simple image capturing. Based on this marker assisted SfM and the resulting fixed markers' poses, one can measure sparse critical information (distances, angles) of the target environment, perform object pose estimation within that space, or create 3D models describing the details of that space.

The rest of the paper is organized as follows. Firstly relevant previous work is briefly discussed. Then the details of the marker assisted SfM are explained from the basic marker SfM to the marker and plane SfM as the RGBD extension. After that, two sets of experiments are discussed to demonstrate sufficient accuracy of the proposed technique. Finally conclusions and future directions are summarized.

PREVIOUS WORK

Geometric modeling of built environment and object pose estimation may be addressed by different methods. As-built survey is a widely applied method using conventional surveying equipment (such as total stations) to measure positions of key

points in the built environment and then generating as-built 2D plans or 3D models (usually in wireframe form). Such point-by-point surveys are accurate but inefficient.

3D scanning technology is thus being increasingly adopted due to the fact that it can efficiently collect millions of 3D points forming a point cloud to describe the environment being scanned (Han et al. 2012; Cho and Gai 2014). The data collection device in 3D scanning is typically a terrestrial laser scanner (TLS) due to its high accuracy. Various algorithms have been proposed in this area, including data collection, registration, shape representation, and object recognition. Interested readers are referred to the literature review by Tang et al. (2010) for more details.

The TLS based 3D scanning methods also have disadvantages, such as high costs of TLS and corresponding data processing software, requirement experts to design and perform the scan and post-process raw scan data, and the large volume and weight of TLS. Thus it is of interest to develop cost-efficient, easy-to-use, and lightweight solutions with sufficient accuracy. Off-the-shelf commercial digital cameras are thus attractive due to their relatively lower costs, weight and volume. Especially with the recent progress in SfM (Snavely et al. 2006) and visual simultaneous localization and mapping (VSLAM) (Engel et al. 2014), digital cameras become promising data collection devices to achieve tradeoffs between cost and accuracy. Thus in construction, computer vision based 3D modeling methods have been investigated with different applications (Brilakis et al. 2010; Golparvar-Fard et al. 2011; Dai and Lu 2012).

All these computer vision based methods inherit the previously mentioned assumption about features, thus potentially suffer from inaccurate and unstable reconstruction or pose estimation in featureless areas. Marker based methods were thus proposed to tackle the challenge (Feng et al. 2015). It is worth noting that active 3D sensors such as Kinect provides another low-cost way of address featureless challenge and has also gain interests in both robotics and construction research (Taguchi et al. 2013; Zhu and Donia 2013). The marker and plane SfM method proposed in this paper combines the marker based approach with such RGBD sensors to create a more compact and concise description of the target environment mixing points and planes, similar to Taguchi et al. (2013).

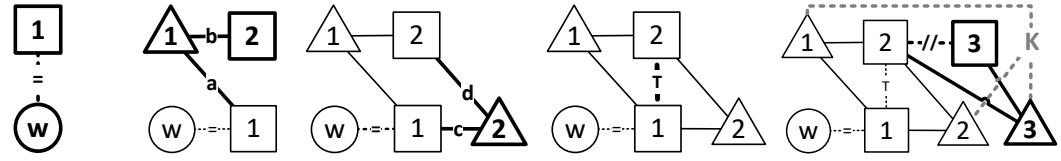
TECHNICAL APPROACH

The marker assisted SfM proposed here contains two methods. The basic method uses only an ordinary RGB camera. The second method extends the first one by replacing the RGB camera with an RGBD camera. They are all based on the theory of *camera marker network* which was detailed by Feng et al. (2015).

Marker Structure from Motion

This first method is a direct application of camera marker networks, which is also the foundation of the proposed solution. The basic idea is to attach markers on planes of the target environment. Then an intrinsically calibrated RGB camera is moved to different proper poses to take a sequence of images of those markers, forming a sequence of views. This results in a dynamic camera marker network of multiple views and multiple markers. When the marker poses are estimated in this network, poses of the corresponding planes can be determined, since it is reasonable

to assume these printed markers are on those planes. Because the poses of all markers cannot be finally estimated without moving the single camera to different views, this method is dubbed as marker SfM, as it is very similar to traditional point-based SfM. As illustrated in Figure 1, the marker SfM contains several operations to grow and maintain a dynamic camera marker graph, which are detailed in the following subsections.



(a) Initialization (b) Discover New Markers (c) Re-observe Old Markers (d) Detect Marker Constraints (e) Final Optimization

Figure 1. Different operations in marker structure from motion

Initialization

Before taking the first image, a camera marker graph needs to be initialized. As shown in Figure 1(a), this initialization is to specify the relationship between at least one marker node (denoted as a square) and the world node (denoted as the circle). Essentially this operation is to define what the world coordinate frame is in the resulting 3D model. In this graph, such relationships are represented as soft constraints (dotted lines), i.e. a triplet $\mathbf{c} = (s, e, \mathrm{g})$ where s and e are the indices of the two connected nodes $\mathbf{x}_s, \mathbf{x}_e$ and g is the constraint function. In initialization, this constraint is to directly specify the pose parameters of a marker s in the world coordinate frame e and thus termed as the *fixed-node* constraint (denoted with '='):

$$\mathrm{g}_{=}(\mathbf{x}_s, \mathbf{x}_e) = \mathbf{P}_{=}^{-1/2}(\mathbf{x}_s - \bar{\mathbf{x}}_s) \tag{1}$$

where $\bar{\mathbf{x}}_s = [\bar{\mathbf{e}}_s^{\mathbf{T}}, \bar{\mathbf{t}}_s^{\mathbf{T}}]^{\mathbf{T}}$ is the known pose parameters (orientation **e** and position **t**) of the markers s, and $\mathbf{P}_{=}$ is the cross-covariance matrix of the known values $\bar{\mathbf{x}}_s$ for properly weighting this constraint (in the sense of Mahalanobis distance). The markers added in this operation are termed as control markers and have similar purposes as control points in conventional surveying and photogrammetry.

Discover New Markers

After initialization, a user can start taking photos of those markers. Each photo will correspond to a new view node (denoted as a triangle) to be added to the initialized camera marker graph. An important rule of thumb of choosing proper poses for taking photos is that at least two markers (and generally the more the better) should be detectable in the photo, and their images should be as far apart as possible in this photo, leading to better conditioning of the system equations.

Whenever a new photo is taken, the corresponding node needs to be initialized and added to the graph. There are three situations. The first one is that none of the markers detected in this new photo have been seen before. In this case, this view cannot be readily initialized to the graph because of no connections to existing markers. Thus this photo can be either discarded or cached for later processing whenever such connections can be found.

The second situation is that there exist some markers that have been observed before, while others are newly detected. In this case, the new view can be initialized by calculating the relative pose between those observed markers and this view, using either homography decomposition or solving the perspective-n-point problem (Feng et al. 2015). Once the view is initialized to the graph, those newly detected markers can now be subsequently initialized to the graph. An example is shown in Figure 1(b). After the initialization in (a), firstly a new view (view 1) is added by initialize the view's pose using edge a. Then the new marker (marker 2) is added using edge b.

Re-observe Old Markers

The third situation is that all of the detected markers are old markers in the graph. In this case, the new view can be added using the same method as in the second situation while no new markers to be added. For example Figure 1(c), the new view (view 2) can be added using edge c or d or both.

It is however worth noting that in marker SfM, if only one old marker is detected in a new photo, this photo is of little value to be added to the graph. Because if conditioned on this re-observed old marker's pose, the new view's is and will always be independent with all other views and markers, since no more edges can be linked back to this view. This is different with marker nodes since by adding more views the conditional independence could be removed, e.g., the marker 2 in Figure 1(b) is conditional independent on the view 1, but not any more after the view 2 is added in Figure 1(c).

Detect Marker Constraints

After multiple marker nodes are initialized and added to the graph, their geometric relationship can be examined to detect potential pose constraints between markers. Typical constraints include *parallelism*, *perpendicularity*, *coplanarity*, and the aforementioned *fixed-node* constraint, which are all enforced as soft constraints. It preserves the uniform representation of each node's pose parameter, and represents constraints as a special type of observations. During optimizations, these constraint residuals are minimized together with ordinary observation residuals. For example, the perpendicularity constraint (denoted by '$\perp$'), and the parallelism constraint (denoted by '//'), and the coplanarity constraint (denoted by 'p'), can be represented respectively in equation (2), where $\mathbf{r}_3(\mathbf{e})$ is the third column of the rotation matrix $\mathbf{R}(\mathbf{e})$ computed by the well-known Rodrigues' formula , representing a marker's normal direction in the world coordinate frame; $\sigma_{//}^{-1}$, $\sigma_{\perp}^{-1}$, and σ_{p}^{-1} are the weighting factors for the constraints indicating the user's confidence of these constraints:

$$\begin{aligned} \mathrm{g}_{\perp}(\mathbf{x}_i,\mathbf{x}_j) &= \mathrm{g}_{\perp}([\mathbf{e}_i^{\mathbf{T}},\mathbf{t}_i^{\mathbf{T}}]^{\mathbf{T}},[\mathbf{e}_j^{\mathbf{T}},\mathbf{t}_j^{\mathbf{T}}]^{\mathbf{T}}) = \mathbf{r}_3(\mathbf{e}_i)^{\mathbf{T}}\mathbf{r}_3(\mathbf{e}_j)\sigma_{\perp}^{-1} \\ \mathrm{g}_{//}(\mathbf{x}_i,\mathbf{x}_j) &= \left[1-\mathrm{abs}\left(\mathbf{r}_3(\mathbf{e}_i)^{\mathbf{T}}\mathbf{r}_3(\mathbf{e}_j)\right)\right]\sigma_{//}^{-1} \\ \mathrm{g}_{\mathrm{p}}(\mathbf{x}_i,\mathbf{x}_j) &= \left[1-\mathrm{abs}\left(\mathbf{r}_3(\mathbf{e}_i)^{\mathbf{T}}\mathbf{r}_3(\mathbf{e}_j)\right), \mathrm{abs}\left(\mathbf{r}_3(\mathbf{e}_i)^{\mathbf{T}}\mathbf{t}_i\right)-\mathrm{abs}\left(\mathbf{r}_3(\mathbf{e}_j)^{\mathbf{T}}\mathbf{t}_j\right)\right]^{\mathbf{T}}\sigma_{\mathrm{p}}^{-1} \end{aligned} \quad (2)$$

Final Optimization

After a new photo is processed by the above three operations just described, a full optimization of the all poses can be performed as in the original camera marker

network method (Feng et al. 2015). When all photos are processed, a final optimization adjusting the camera intrinsic parameters with all poses is performed considering that camera intrinsic parameters were calibrated previously independent to this estimation. This slightly modifies the optimization equation as:

$$\hat{\mathbf{K}}, \hat{\mathbf{X}} = \arg\min_{\mathbf{K},\mathbf{X}} \left\| \hat{\mathbf{Z}} - \mathbf{F}_C(\mathbf{K}, \mathbf{X}; \mathbf{Y}) \right\|^2_{\mathbf{P}_{\hat{\mathbf{Z}}}} + \left\| \mathbf{G}(\mathbf{X}) \right\|^2 \quad (3)$$

where the camera intrinsic parameters vector **K** becomes a part of system state, instead of the original parameters for the function **F** in Feng et al. (2015). This is illustrated in Figure 1(e) where the **K** is shown in grey with dotted edges linking to all the views whose poses are directly affected when adjusting **K**.

The above five operations can thus be summarized in Algorithm 1, which described a post processing version of marker SfM. It can be conveniently converted to online processing by 1) performing step 1 to 8 for each new photo; 2) evaluating pose uncertainties as in (Feng et al. 2015); 3) performing a final optimization after the user stops and all markers are estimated with sufficient certainty.

Algorithm 1. Marker Structure from Motion

Initialize a camera marker graph ***G*** using equation (1);
For i = 1 to N:

1. Detect markers in photo $\boldsymbol{I}_i$;
2. If no detected marker exists in ***G***, swap $\boldsymbol{I}_i$ and $\boldsymbol{I}_{i+1}$, and redo step 1;
3. Initialize a new view node ***v*** in ***G*** using detected markers that exist in ***G***;
4. Add an edge between ***v*** and each detected existing marker in ***G***;
5. Initialize a new marker node for each detected markers that is not yet in ***G***;
6. Add an edge between ***v*** and each of these newly added marker nodes;
7. Perform an optimization of all nodes as in Feng et al. (2015);
8. For each new marker ***n*** added in step 5, and each node ***m*** in ***G*** other than ***n***:
   a. Calculate constraint residuals between ***n*** and ***m***, e.g. using equation (2);
   b. If any residual is smaller than a pre-defined threshold, request user approval for adding a constraint edge between node ***n*** and ***m***;

Final optimization of intrinsic parameters and all nodes using equation (3).

Marker and Plane Structure from Motion

Many applications can be readily addressed using the marker SfM proposed above, for example, measuring dimensions for interior design. Because markers' poses can be used to calculate distances between parallel planes (e.g., height of a room, etc.), dihedral angles between planes (e.g., walls, roofs, etc.), and so on. Essentially the markers and the camera serve as a more accurate tape and protractor.

In some advanced applications, such as rapidly creating a realistic 3D model of a room, the marker SfM might not be satisfactory. With low cost 3D sensors like Kinect, a marker and plane SfM using a RGBD camera as the data collection device is thus proposed here. This method extends camera marker networks with a new type of observations, i.e., 3D planes extracted from depth image (Feng et al. 2014).

Extended Camera Marker Graph

Just as the original camera marker networks, such an extended one can be considered as a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with two types of nodes $\mathbf{V} = (\mathbf{X}, \mathbf{\Pi})$ and three types

of edges **E** (marker and plane observation edges, and constraint edges). The two new elements are plane nodes $\mathbf{\Pi} = \{\mathbf{p}_\mathrm{i} \equiv (\boldsymbol{e}_\mathrm{i}, d_\mathrm{i})\}$ and plane observation edges $\mathbf{Q} = \{\mathbf{q}_\mathrm{j} \equiv (v_\mathrm{j}, p_\mathrm{j}, \mathbf{A}_\mathrm{j})\}$. Each plane node $\mathbf{p}_\mathrm{i}$ is a 3D column vector parameterizing that plane's orientation and location. Each plane observation edge contains a view's index v_j and a plane's index p_j, and also an observation matrix $\mathbf{A}_\mathrm{j}$ 3D anchor points sampled from all points in plane p_j observed at view v_j, similar to Taguchi et al. (2013). Like equation (2), the residuals of an anchor point $\mathbf{a}_\mathrm{i}$ in such an edge $\mathbf{q}$=(v, p, $\mathbf{A}$) is $\mathrm{h}(\mathbf{x}_v, \mathbf{p}_p, \mathbf{A}) = \mathbf{P_q}^{-1/2}[\cdots, \mathbf{n}(\mathbf{e}_p)^\mathbf{T}(\mathbf{R}(\mathbf{e}_v)\mathbf{a}_i + \mathbf{t}_v) + d_p, \cdots]^\mathbf{T}$, where $\mathbf{n}(.)$ is the 3D unit normal vector of the plane p, and $\mathbf{P_q}^{-1/2}$ is the weighting matrix for this edge. This essentially calculates the point-plane distances between the plane and each anchor point transformed into the world coordinate frame.

Stacking such equations for all plane observation edges results in equation $\mathbf{H}(\mathbf{X}, \mathbf{\Pi}) = \left[\mathrm{h}(\mathbf{x}_{v_1}, \mathbf{p}_{p_1}, \mathbf{A}_1)^\mathbf{T}, \cdots, \mathrm{h}(\mathbf{x}_{v_S}, \mathbf{p}_{p_S}, \mathbf{A}_S)^\mathbf{T}\right]^\mathbf{T}$. Thus the original optimization and the full optimization including camera intrinsic parameters are extended to:

$$(\hat{\mathbf{X}}, \hat{\mathbf{\Pi}}) = \underset{\mathbf{X},\mathbf{\Pi}}{\arg\min} \left\|\hat{\mathbf{Z}} - \mathbf{F}(\mathbf{X}; \mathbf{Y}, \mathbf{K})\right\|^2_{\mathbf{P}_{\hat{\mathbf{Z}}}} + \left\|\mathbf{H}(\mathbf{X}, \mathbf{\Pi})\right\|^2 + \left\|\mathbf{G}(\mathbf{X})\right\|^2 \tag{4}$$

$$(\hat{\mathbf{K}}, \hat{\mathbf{X}}, \hat{\mathbf{\Pi}}) = \underset{\mathbf{K},\mathbf{X},\mathbf{\Pi}}{\arg\min} \left\|\hat{\mathbf{Z}} - \mathbf{F}_\mathrm{c}(\mathbf{K}, \mathbf{X}; \mathbf{Y})\right\|^2_{\mathbf{P}_{\hat{\mathbf{Z}}}} + \left\|\mathbf{H}(\mathbf{X}, \mathbf{\Pi})\right\|^2 + \left\|\mathbf{G}(\mathbf{X})\right\|^2 \tag{5}$$

Algorithm 2. Marker and Plane Structure from Motion

Initialize a camera marker graph ***G*** using equation (1);
For i = 1 to N:

1. Perform step 1 to 8 of Algorithm 1 on the photo $\boldsymbol{I}_i$;
2. If the photo was swapped in 1, swap the point cloud $\boldsymbol{D}_i$ with $\boldsymbol{D}_{i+1}$;
3. Extract planes based on Feng et al. (2014) on $\boldsymbol{D}_i$;
4. For each extracted plane ***p***:
    a. Transform ***p*** to the world coordinate frame using the pose of the new view node ***v*** just added in step 3 of Algorithm 1;
    b. Find the best matching plane ***q*** of ***p*** in all planes in ***G***, by equation (6);
    c. If the differences between ***q*** and ***p*** are within pre-defined thresholds:
        i. Add a new plane observation edge between ***v*** and ***q*** to ***G***;
        ii. Expand the boundary of ***q*** by ***p***;
    d. Otherwise:
        i. Add ***p*** as a new plane node to ***G***;
        ii. Add a new plane observation edge between ***v*** and ***p*** to ***G***;
5. Perform an optimization of all nodes using equation (4);

Final optimization of intrinsic parameters and all nodes using equation (5).

Plane Matching

It is worth noting another difference in this extended graph, which is the matching of a currently observed plane to a plane in the graph. While matching markers across different views in the original graph is straightforward with markers' unique IDs, 3D planes extracted from point clouds cannot be matched in that way. However with marker SfM, a newly added view's pose is already estimated in the world coordinate frame approximately. Thus each extracted 3D plane in this view can

be transformed to the world coordinate frame and matched to its most "similar" plane in the extended graph, in terms of criteria such as the normal and distance deviations:

$$\mathrm{d}(\mathbf{p}_a, \mathbf{p}_b) = [\mathrm{acos}\left(\mathbf{n}(\mathbf{e}_a)^{\mathbf{T}}\mathbf{n}(\mathbf{e}_b)\right), \mathrm{abs}(d_a - d_b)]^{\mathbf{T}} \quad (6)$$

where the second term may be replaced by the average distance between one plane and anchor points of the other plane, to increase matching robustness.

Finally, the extended marker and plane SfM is summarized in Algorithm 2. Note that constraint edges in marker SfM can be add between plane nodes in similar ways as described in step 8 of Algorithm 1, thus not repeated here.

EXPERIMENTAL RESULTS

Accuracy of Marker Structure from Motion

Both Algorithm 1 and Algorithm 2 were implemented in MATLAB. 30 Apriltags were printed on A4 papers, each tag of size 172mm. These markers are then attached on walls, floors and ceilings of a two-story apartment. The marker structure was recovered using 66 photos.

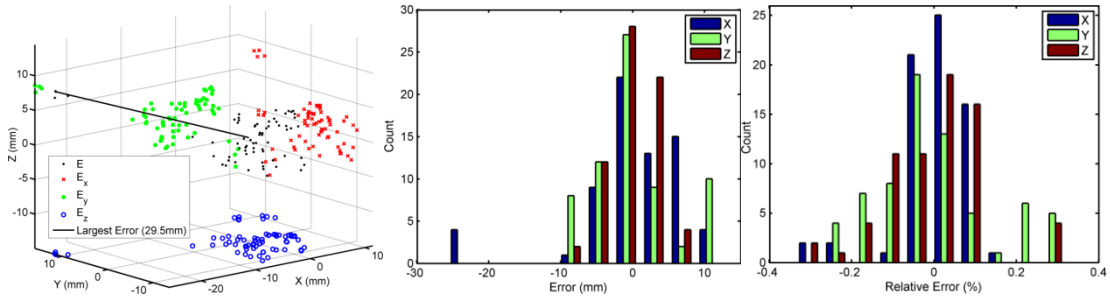


Figure 2. Marker SfM results and position errors

To evaluate the accuracy of marker SfM, a Topcon PS 101A total station (nominal precision: 2mm and 0.5 second within 100m) was employed to survey those markers' poses as a baseline for comparison. To avoid station registration errors in this baseline result, the total station was setup in a single station that can directly observe a maximum of 17 markers. Then the four corners of these 17 markers were surveyed, resulting in ${}^{s}\mathbf{X}$. The corresponding estimated corner positions, ${}^{w}\mathbf{X}$, were calculated from marker SfM. Using the well-known rigid body registration, one can calculate ${}^{w}\mathbf{R}_s, {}^{w}\mathbf{t}_s$ that transforms surveyed points ${}^{s}\mathbf{X}_i$ to the world coordinate frame, and thus the discrepancies $\mathbf{E}_i = {}^{w}\mathbf{X}_i - ({}^{w}\mathbf{R}_s {}^{s}\mathbf{X}_i + {}^{w}\mathbf{t}_s)$.

The bottom-left of Figure 2 shows the error vector $\mathbf{E}_i$ in black '**.**', and its projection onto each side planes ($\mathbf{E}_x$, $\mathbf{E}_y$, $\mathbf{E}_z$). The bottom-middle shows a histogram of $\mathbf{E}_i$ and bottom-right shows the relative error ($\mathbf{E}_i$ dividing by the scale of all these markers' distribution on the X, Y and Z directions). The majority of errors are **within 10mm** with the largest positional error of 29.5mm, while the absolute errors on average are **5, 4, 2mm** on X, Y, Z directions respectively. Note that these markers are distributed with **a scale of about 9m** along the X direction. Thus the maximum relative error of this marker SfM is about **0.3%**, which is of sufficient accuracy

considering that it was achieved using an ordinary webcam-style RGB camera (of image size 640x480 pixels) on the Kinect device (depth image to be used below).

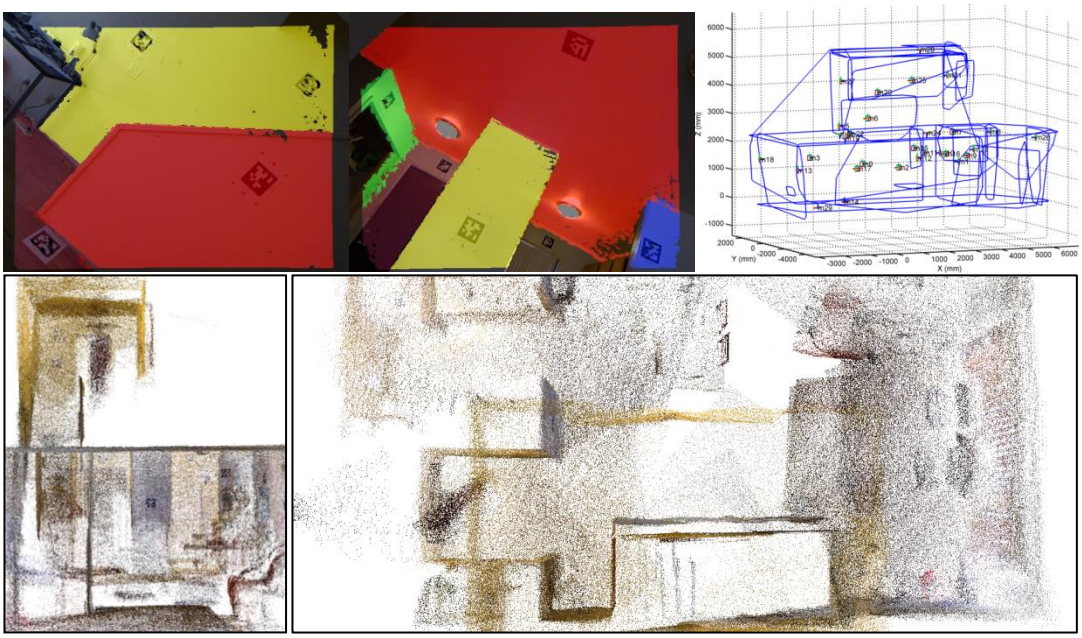


Figure 3. Results of marker and plane SfM

As-built Models from Marker and Plane Structure from Motion

A second experiment was performed to test the marker and plane SfM algorithm, using the previous 66 RGB images, and the corresponding 66 depth image. Some intermediate plane extraction results are shown in the top-left and top-middle of Figure 3. The 66 point clouds were transformed into the world coordinate frame using estimated poses for each view and then merged into a single point cloud, whose front, and top views are shown in the bottom of Figure 3 (down sampled for clarity). This is the point cloud form of the as-built model of the apartment, with only a few paper markers and a low cost Kinect camera. A more concise form is shown in the top-left of Figure 3. These polygons are the observed boundaries of each plane. Compared to the point cloud form, this polygon form is more close to the parametric models used in BIM with more semantics.

CONCLUSIONS

In conclusion, this research extended the camera marker network method, resulted in two types of novel 3D modeling and pose estimation techniques. The two techniques require only a low cost RGB/RGBD camera and a few markers to perform dimension measurements, 3D scanning and modeling, and also pose estimation. Furthermore, the experiments have shown these techniques' satisfactory accuracy at sufficiently large scales. Moreover, the marker and plane SfM algorithm enables automatic wireframe model generation which could help as-built BIM generation.

The future direction for this research contains several aspects. Firstly the marker and plane SfM algorithm's accuracy needs to be improved since the errors in the point cloud from Kinect adversely affect the marker SfM accuracy. Secondly a

better user guidance algorithm needs to be integrated where end users will be guided by the algorithm step by step to take photos at the best poses generated by the algorithm. This will greatly shorten the amount of time for end users to learn such techniques. Thirdly, the potential of multiple cameras needs to be explored to increase the flexibility and reduce the amount of views needed. Last but not least, VSLAM needs to be integrated to further increase the efficiency and range of applications.

REFERENCES

Brilakis, I., Lourakis, M., Sacks, R., Savarese, S., Christodoulou, S., Teizer, J., and Makhmalbaf, A. (2010). "Toward automated generation of parametric BIMs based on hybrid video and laser scanning data." *Advanced Engineering Informatics, 24*(4), 456--465.

Cho, Y., and Gai, M. (2014). "Projection-Recognition-Projection Method for Automatic Object Recognition and Registration for Dynamic Heavy Equipment Operations." *J. Comput. Civ. Eng., 28*(5), A4014002.

Dai, F., and Lu, M. (2012). "Three-dimensional modeling of site elements by analytically processing image data contained in site photos." *J. Constr. Eng. Manage., 139*(7), 881--894.

Engel, J., Schops, T., and Cremers, D. (2014). "LSD-SLAM: Large-scale direct monocular SLAM." In *Computer Vision--ECCV 2014* (834--849). Springer.

Feng, C., Dong, S., Lundeen, K. M., Xiao, Y., and Kamat, V. R. (2015). "Vision-Based Articulated Machine Pose Estimation for Excavation Monitoring and Guidance." *Proceedings of the 32nd ISARC.*, Oulu, Finland

Feng, C., Taguchi, Y., and Kamat, V. R. (2014). "Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering." *IEEE ICRA*, Hong Kong, China, 6218-6225.

Golparvar-Fard, M., Bohn, J., Teizer, J., Savarese, S., and Pena-Mora, F. (2011). "Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques." *Automation in Construction, 20*(8), 1143--1155.

Han, S., Cho, H., Kim, S., Jung, J., and Heo, J. (2012). "Automated and efficient method for extraction of tunnel cross sections using terrestrial laser scanned data." *J. Comput. Civ. Eng., 27*(3), 274--281.

Snavely, N., Seitz, S., and Szeliski, R. (2006). "Photo tourism: exploring photo collections in 3D." *ACM transactions on graphics (TOG), 25*, 835--846.

Taguchi, Y., Jian, Y.-D., Ramalingam, S., and Feng, C. (2013). "Point-Plane SLAM for Hand-Held 3D Sensors." *IEEE ICRA*, Karlsruhe, Germany, 5182-5189.

Tang, P., Huber, D., Akinci, B., Lipman, R., and Lytle, A. (2010). "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques." *Automation in construction, 19*(7), 829--843.

Zhu, Z., and Donia, S. (2013). "Potentials of RGB-D cameras in as-built indoor environments modeling." *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, Los Angeles, CA, 23--25.