Monitoring Changes of 3D Building Elements from Unordered Photo Collections

Mani Golparvar-Fard
Virginia Tech
golparvar@vt.edu

Feniosky Peña-Mora
Columbia University
feniosky@columbia.edu

Silvio Savarese
University of Michigan- Ann Arbor
silvio@eecs.umich.edu

Abstract

This paper proposes an automated method for monitoring changes of 3D building elements using unordered photo collections as well as Building Information Models (BIM) that pertain information about geometry and relationships of elements. This task is particularly challenging as existing construction photographs are taken under different lighting and viewpoint conditions, are uncalibrated and extremely cluttered by equipment and people. Given a set of these images, our system first uses structure-from-motion, multi-view stereo, and voxel coloring algorithms to calibrate cameras, reconstruct the construction scene, quantize the scene into voxels and traverse and label the voxels for observed occupancy. The BIM is subsequently fused into the observed scene by a registration-step and voxels are traversed and labeled for expected visibility. Next, a machine learning scheme built upon a Bayesian model is proposed that automatically detects and tracks building elements in presence of occlusions. Finally, the system enables the expected and reconstructed elements to be explored with an interactive, image-based 3D viewer where construction progress deviations are automatically color-coded over the BIM. We present promising results on several challenging construction photo collections under different lighting conditions and sever occlusions.

1. Introduction

Accurate and efficient monitoring of building components that are under construction or for a building that is damaged by a disaster (during disaster rescue operations) is an important research problem. It directly supports construction control decision making [2], [3], [11], disaster response operations [19] and has applications in autonomous robotics [15]. Current monitoring methods include manual data collection and extensive data extraction from construction drawings and work schedules (i.e., sequence of activities through which building is constructed). There is a need for a systematic method to automatically track building structure changes, allowing data to be collected easily and at almost no cost, processing automatically and reporting back in a format useful for all project participants.

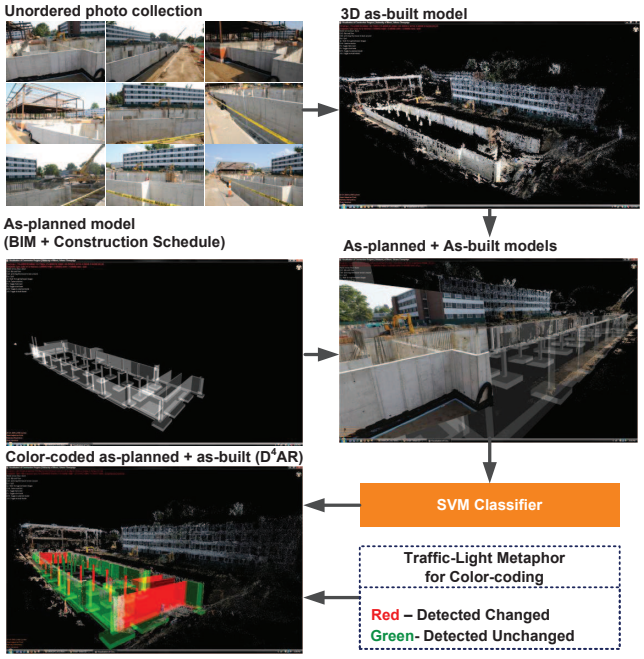

Figure 1. An overview of the proposed progress monitoring.

Nowadays, cheap and high resolution digital cameras, low cost memory and increasing bandwidth capacity have enabled capturing and sharing construction photographs on a truly massive scale. For example, on a 200,000 sq.ft. building project, an average of 500 photos/day is collected by construction professionals. The availability of such rich imagery - which captures dynamic construction scenes from almost every conceivable viewing position and angle at minimal cost - may enable geometrical reconstruction of building sites at high resolution. In the meantime, Building Information Models (BIM) are also increasingly gaining attention as binding components of construction contracts. These models are formed similar in shape to conventional CAD models, yet contain *semantics* about structural geometries and their spatial and material properties [7]. They can

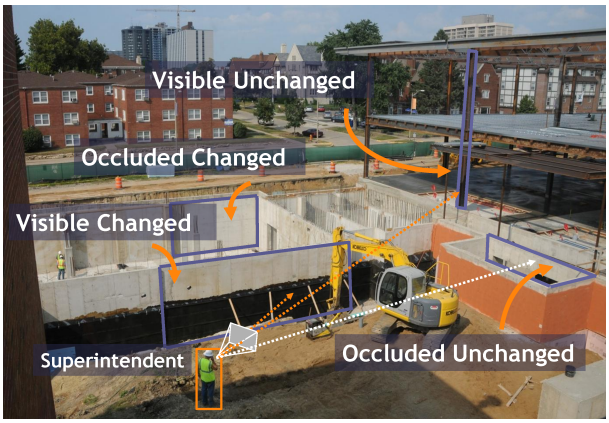

Figure 2. Monitoring and progress visibility.

also be linked with schedules, cost and health monitoring information, forming sequential models that allow expected construction or pre-disaster structure to be analyzed in 4D (3D + time), and monetary savings of an automatic process to be realized. Application of 4D BIM for the purpose of progress monitoring is desirable since: (1) these models (from now on, we will call them as *as-planned* models) link 3D model of the scene with spatial structural attributes as well as temporal construction schedule (how buildings are expected to be constructed) and form proper baselines for measuring expected progress; (2) if linked with unordered photo collections, these models integrate both expected and actual progress perspectives and enable automated measurement and visualization of progress deviations.

Nonetheless, automated linking of unordered photo collections with *as-planned* models is challenging. First, such imagery is **unordered**, **uncalibrated**, with widely unpredictable and uncontrolled lighting conditions. Second, **visibility order** and **occlusions** need to be considered for successful alignment. In particular one needs to account for two types of occlusions: (1) **Static occlusions**: self-occlusions caused by progress itself (e.g., a façade blocking observation of elements at interior) or occlusions caused by temporary structures (e.g., scaffolding or temporary tenting); and (2) **Dynamic Occlusions**: rapid movements of construction machinery and workers during the time photographs are taken. Developing computer vision techniques that can effectively work with such imagery to monitor building element changes has been a major challenge.

In this paper, we introduce a new approach for monitoring building elements from unordered photographs based on *a priori* (*as-planned* models). First using Structure-from-Motion (SfM), scene point cloud is reconstructed and images are automatically calibrated (from now on, we will call it *as-built*). Subsequently the *as-built* is registered over the *as-planned* model and improved by Multi-View Stereo (MVS). At this stage a new voxel coloring and labeling algorithm is used to generate a volumetric reconstruction, labeling different areas according to visual consistent observations. Same labeling process is conducted on the *as-planned* model to identify occupied and visible areas expected to be monitored. Finally a Bayesian probabilistic model is introduced to automatically monitor changes and assess progress of *as-planned* elements (as construction site evolves in time) by comparing measurements with dynamic thresholds learned through a Support Vector Machine (SVM) classifier. The algorithm automatically accounts for occlusions and recognizes if building elements are missing because of occlusions or because of changes. This makes our model to be *the first probabilistic model for progress monitoring and visualization of deviations that incorporates both as-planned models and unordered daily photographs in a principled way*. Fig. 1 shows an overview of the proposed progress monitoring. Unlike other methods that focus on application of laser scanners [2, 3, 8] or time-lapse photography [11, 14, 27, 17], our model is able to use existing information without adding burden of explicit data collection on Architecture/ Engineering/ Construction (AEC) professionals and reports competitive accuracies in monitoring progress compared to [2, 14, 17] especially in presence of occlusion in observations. Fig. 2 highlights technical challenges of a vision-based building tracking system under which changes in elements need to be detected.

1.1. Related work

The proposed algorithm in this work, builds upon a set of SfM algorithms where the objective is to reconstruct the scene without any strong prior [1, 4, 5, 20, 10, 21, 23, 24, 26]. In some of these techniques such as Zebedin *et al*. [26], aerial images are used for reconstructing building models. In others such as Agarwal *et al*. [1] entire city is sparsely reconstructed from unordered photographs collected from Internet, or as in Cornelis *et al*. [4], and Pollefeys *et al*. [20] building façades are reconstructed from car-mounted videos. Our work in volumetric reconstruction of the scene is closest to Furukawa *et al*. [10] However, compared to [10], our images are widely distributed in the scene as the focus is to use existing images that are of immediate importance to AEC professionals. Therefore they may not have enough overlap for Multi View Stereo reconstruction. In addition, we do not assume building interiors are predominantly piece-wise planar surfaces as during construction, building elements may have different shapes/forms. Finally, the quality of reconstruction is not the focus, rather we focus on detecting changes in elements and the scene given partial occlusions.

Unlike other semi-automated building monitoring methods that focus on application of laser scanners [2, 3, 8] or time-lapse photography [11], [14, 27, 17], our model is

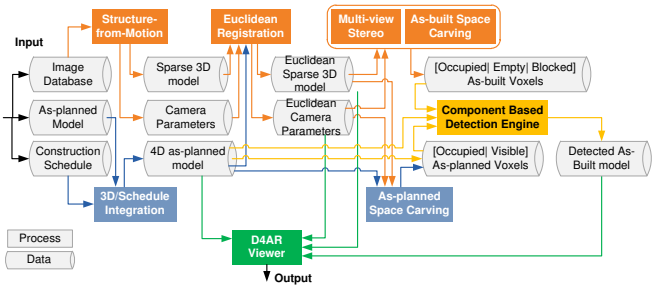

Figure 3. An overview of data and process.

able to use existing information without adding burden of explicit data collection and reports competitive accuracies compared to [2, 14, 17] especially in presence of occlusions.

2. Overview of tracking and analysis

As shown in Fig. 3, our work is based on application of construction imagery, *as-planned* models and schedules to automatically monitor progress of building elements and generate a D^{4}AR (4 Dimensional Augmented Reality) model for visualizing progress deviations. Using SfM techniques, first we generate a 3D *as-built* point cloud. Subsequently the *as-planned* model is superimposed with the *as-built* model. The Euclidean point cloud as well as the camera parameters are fed into MVS algorithm [10]. The results are fed into a new voxel coloring algorithm developed in this research to get a dense reconstruction of the *as-built* site and label scene for *as-built* occupancy. Using a similarly structured voxel coloring algorithm, *as-planned* scene voxels are labeled for *occupancy* and *visibility*. The labeled *as-built* and *as-planned* spaces are fed into a novel Bayesian model and monitored by dynamically classifying the results through a Support Vector Machine (SVM) classifier. Finally, detected *as-built* along with camera configurations, plus 4D *as-planned* model are fed into the D^{4}AR viewer to visualize *as-built*, *as-planned* and progress deviations in an integrated fashion. In the following, the SfM and other steps for progress monitoring are presented.

3. As-built and alignment with as-planned

First, similar to [23], a point cloud is automatically reconstructed from a set of images. This module consists of the following steps: (1) Analyzing images and extracting SIFT feature points [16]; (2) Matching features across the set of images [12]; (3) Find an initial solution for the 3D locations of these features, calibrating cameras for an initial pair and reconstructing the rest of the observed scene plus estimating motion of the cameras based on bundle adjustment [18, 25] and finally (4) Registering point clouds that are generated from each photo collection to build a 4D *as-built* model. In our case, there is no need to infer temporal order as in [21]. Rather that information is automatically extracted from EXIF tag of JPEG images. Yet, for *as-built* reconstruction, we only use images collected in one day (since construction may significantly change how the site looks like at different days). We analyzed performance of this reconstruction module on two sets of 112 and 160 images that are collected in two different weeks on a Residence Hall (*RH*) construction project. In both cases, field engineer causally took images within a few minutes. Fig. 4a–b represents reconstructed sparse scene from the same image subset and illustrate registered cameras in the D^{4}AR virtual environment. Once a camera is visited, camera frustum is textured with the image so user can interactively zoom-in and visually acquire information on progress, quality, and safety as well as workspace logistics. Fig. 4c shows location of a camera frustum; 4d shows the site through the camera; and 4e demonstrates the image textured on camera's viewing plane.

To align *as-built* point cloud with the *as-planned* model, transformation between these two Cartesian coordinate systems needs to be found. In this case, given an *as-built* point cloud that is reconstructed from photos collected at a time (t), we use the *as-planned* model that is updated up to time (t') ($t' \leq t$);i.e., the *as-planned* model shows progress up to time (t'). The alignment transformation can be formed as a rigid-body motion and hence can be decomposed into rotation and translation. However in SfM, the reconstruction can be up to an unknown uniform scale. To solve this transformation with uniform scale (7 DOF), we need at least 3 corresponding points. These points could be surveying control points or a set of points that represent geospatial location of the site. In our case, a user selects these points from corners of the walls and columns as their detection and correspondence is visually easier. Let there be n corresponding points from *as-planned* and *as-built* models for registration. We denote the two coordinate system points by $r_{p,i}$ and $r_{b,i}$, respectively, where i is the number of corresponding points which ranges from 1 to n, $r_{p,i}$ and $r_{b,i}$ be the Cartesian coordinates of *as-planned* and *as-built* models respectively. We look for transformation of the form:

$$r_p = sR(r_b) + T \tag{1}$$

where s is a uniform scale factor, T is the translational offset and $R(r_b)$ is the rotated version of the *as-built* model. This can be formulated as:

$$\sum_{i=1}^{n} \|e_i\|^2 = \sum_{i=1}^{n} \|r_{i,p} - sR(r_{i,b}) - T\|^2 \tag{2}$$

We follow Horn [13] to get a closed-form solution to the least square problem of absolute orientation. In our system, this procedure needs to be done only once to have the initial point cloud registered to the 3D model. From then after,

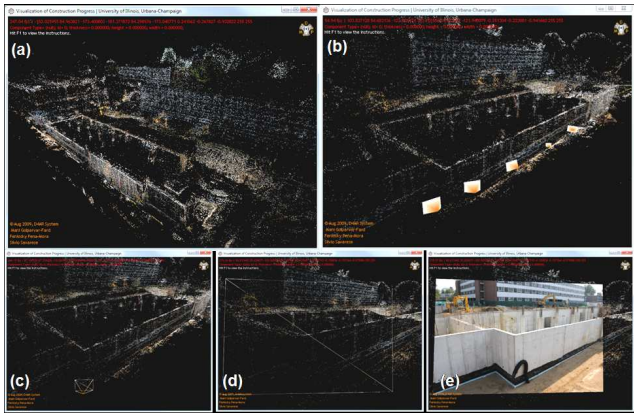

Figure 4. (a) Synthetic view of the reconstructed as-built; (b) Five camera frustra representing location/orientation of the superintendent when photographs were taken; (c) One camera frustum is rendered and its location/orientation is visualized; (d) The as-built point-cloud observed through same camera frustum; and (e) camera frustum textured visualizing 3D point-cloud and photograph.

we only need to register the point clouds that are generated from new images to the underlying point cloud. For this purpose, we use a set of images that are showing part of the scene which is not significantly changed from one-day to another. For this dataset we use an ICP algorithm that can solve for scale as well (Du *et al*. [6]). This method automatically finds a random set of points from each point cloud and automatically aligns the new point cloud to the former one, in turn having the new point cloud registered with the *as-planned* model. This allows 4D *as-built* point clouds to be generated wherein user can navigate the *as-built* scene both spatially and chronologically. The 4D *as-built* registered with the 4D *as-planned* model allows expected and the actual schedule of the project to be compared as well. Fig. 5 shows 3 snapshots from the *RH* project. In Fig. 5a and b two separately reconstructed point clouds are shown while in Fig. 5c the two point clouds are registered and visualized together. In cases (a) and (b), reconstructions are based on 112 and 160 images collected from outside of the basement. Table 1 reports high accuracies, though the accuracy is not sensitive to how the control points are selected. Since usually more than the minimum number of control points (3) is selected, user selection error is minimized.

In order to detect progress, we first discretize the integrated *as-built* and *as-planned* scene Ω into a finite set of opaque voxels (volume element in space) along dominant Euclidean axes wherein each voxel (v) occupies a finite homogeneous volume of the scene ($\delta x \delta y \delta z$) and has a consistent visual appearance. This approach allows us to reason about progress in small elements within the space. In our model, voxels are assumed to be equilateral; therefore resolution of the voxel grid is determined by δ. Given an image

Figure 5. Registration of two as-built point-clouds (a) and (b). The violet point-cloud belongs to 08/20/08 (from 112 images - a) while the orange point-cloud belongs to 08/27/08 dataset (from 160 images- b) (Images best seen in Color).

Π_i, $proj_i(v)$ is used to denote voxel reprojection (in form of a set of pixels) over image i and is measured as:

$$\ni k = 1, 2...8 \rightarrow [u, v, 1]'_k = K_i[R_i|T_i][x, y, z, 1]'_k \quad (3)$$

wherein k is index of voxel corners, K_i is the intrinsic parameters, R_i,T_i represent camera rotation and translation. Since we analyze all images within the SfM step, intrinsic/ extrinsic parameters for all cameras are known at this stage.

3.1. Voxel traversing and labeling

The next step is to traverse the scene and assign two sets of labels (*as-built*, *as-planned*) as well as a color to each voxel. This step allows expected and actual progress in each voxel to be sensed. It is critical to traverse the voxels in a certain order otherwise the reconstruction will not be unique. In order to address this issue, we introduce an *ordinal visibility constraint* similar to Seitz and Dyer [22] allowing certain invariant voxels whose colorings are uniquely defined to be found. Rather than only using this constraint to address uniqueness of the solution, in our approach we find areas within the space that are occupied and visible. First we transform the integrated scene to a new coordinate system wherein the axes are aligned with the dominant axes of the *as-planned* site. This will minimize the search, since we only reason about where progress is expected to be observed. Then we start traversing the scene from the closest voxel to the camera convex hull in a plane normal to the convex hull and eventually in a front-to-back order (Fig. 6). As we march, we verify visibility constraint and for every voxel, define two sets of labels $l(v_{i,j,k})$: (1) *As-built* and (2) *As-planned* labels.

As-built labeling: We first check if a voxel already contains reconstructed SIFT/MVS points. We label that voxel as Occupied (O_b), have that voxel reprojected back on all images that observe it (3) and mark the reprojected pixels in a marking-board. Next, if a voxel does not contain SIFT/MVS points (more often the case), we check for visual consistency. In such cases if voxel reprojections on the image-set do not overlap with marked pixels (i.e., is not fully occluded from all images), and it contains part of the

	as-planned + as-built (a)	as-planned + as-built (b)	as-built models (a)&(b)
Image Size	2144×1424	2573×1709	–
Number of corresp. Points	7	9	Randomly chosen by ICP
e_{mm}	0.20 mm	0.65 mm	0.43 mm

Table 1. Registration error measured on reconstructions shown in Figure 4.

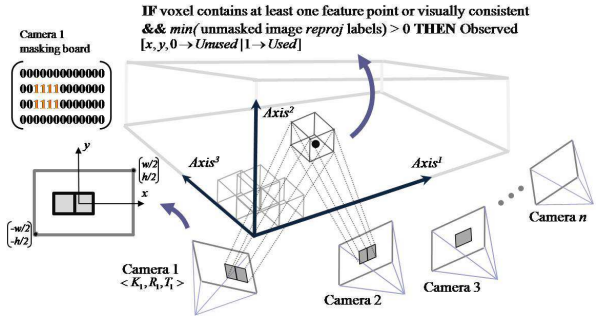

Figure 6. A representation of the as-built site and camera configurations; Reprojections of the voxel are shown on camera frusta 1 and 2. Masking for camera-1 is also shown on the left side. In this case voxel is detected as Occupied; therefore all pixels belonging to reprojection of the voxel on all images are marked "1".

as-built scene (without considering noise or quantization effects), it needs to have equal radiance reprojections. In presence of these effects, we evaluate correlation of pixel colors to quantify voxel consistency:

$$\lambda_v = \frac{(n-1)SD^2}{\sigma^2} \leq thresh \tag{4}$$

Where *SD* is the standard deviation of color values, and σ^2 is the accuracy of irradiance measurement (sensor color measurement error), and finally *n* is number of all images that observe the voxel. If λ_v is less than a maximum allowable correlation error (*thresh*), we label that voxel as visually consistent (O_b) and have that reprojected on all observing images and mark their reprojections accordingly. In our experiments there is a minimum allowable number of reprojected pixels for each voxel from all images (20 pixels). If consistency is not satisfied or the voxel does not contain SIFT/MVS 3D points, we label the voxel as Empty (E_b) and finally if the minimum allowable number of pixels is not satisfied, it means the voxel is occluded from all views and we denote that voxel as Blocked (B_b). In our experiments we have chosen *thresh* by analyzing completeness vs. accuracy for *as-built* reconstruction. This process will have two significant outputs: (1) Labeling all voxels in *as-built* as [$O_b \mid E_b \mid B_b$], allowing reasoning to be made in presence of both static and dynamic occlusions; and (2) creating *as-built* range images. Fig. 7a shows a plan-view of voxel labeling while in 7b reprojected voxel is marked on the image. In 7c unchanged vs. progress observation concept is visualized.

As-planned labeling: The *as-planned* model by itself accounts for static occlusions, though by placing the non-overlapping areas of the *as-built* scene (e.g., equipment, temporary structures) over the *as-planned*, we induce dynamic occlusions to the model. Now we march the scene in a similar fashion to the *as-built*. This time, if an *as-planned* element has at least one of its corners inside a voxel, we label that as Occupied [O_p]. Subsequently we will have a voxel reprojected back on all images that observe that voxel and mark reprojections. In case of non-overlapping *as-planned* and *as-built* areas, we check the consistency from the *as-built* marking and have visually consistent voxels reprojected back on all images for marking pixels. This allows us to track occlusions since if the reprojections contain the minimum unmarked pixels, we can label the voxel as Visible [V_b]. In our model, all labels are independent and are marked with binary values (1 or 0). Image boards are also marked so that if a pixel is observed, the pixel is labeled with 1 and if not observed, remains as 0 (See Fig. 6). Such labeling allows reliable reasoning about progress in partially visible areas.

3.2. Probabilistic progress detection and discriminative learning

Now that the scene is labeled for occupancy, visibility and occlusion, we can formulate progress (observation per expected *as-planned* element *i*) as a binary value (E^i): E^i =1 if progress is detected and E^i = 0 if not. First, we break the *as-planned* model into independent elements given the desirable level of detail for monitoring. Let's look into the example of an *exterior brick wall* in Fig. 7c. In this case we first check for observation of each expected building elements associated with a construction activity (each element *i* as E^i). Let's assume that each element E^i associated with this activity consists of *n* voxels. We introduce a set of probability events: Within a given volume in the scene (ω^i): Let η be the event of observing an occupied *as-built* element, θ_p be the event of observing as-planned element, and θ_T be the event that an *as-planned* element is occupied. We define probability of observing progress for element E^i associated with a given schedule activity as a conditional probability:

$$P(\eta^i|\theta^i_T) = \frac{P(\theta^i_T|\eta^i)P(\eta^i)}{P(\theta^i_T)} \tag{5}$$

Where $P({\theta^i}_T|\eta^i)$ is probability of observing expected *as-planned* element given some evidence of occupancy; $P(\eta^i)$

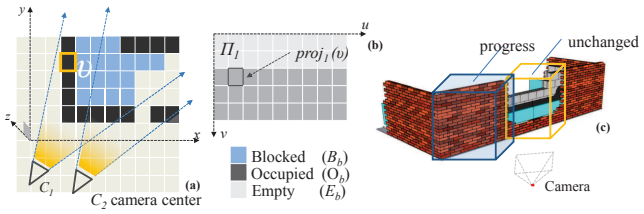

Figure 7. (a) Plan view of *as-built* voxel coloring; (b) image marking for C1 - Projection of (v) is marked as occupied. (c) progress vs. unchanged observations.

probability of observing expected *as-built* element (a function of confidence in coloring the voxel; occupancy within element belonging to expected *as-built*) and P($\theta^i{}_T$) probability of observing expected progress. Each element is computed as follows:

For as-built:

$$P(\theta^i_T|\eta^i) = [\frac{\Sigma O_b}{\Sigma O_b + \Sigma E_b}]_{\theta_p} \tag{6}$$

For the as-planned:

$$P(\theta^i_p) = [\frac{\Sigma V_p}{\Sigma O_p}]_{\theta^i_p} \tag{7}$$

$$P(\theta^i_T) = (\frac{t}{d})V \tag{8}$$

here $P(\theta^i{}_T)$ is the probability of expecting progress (percentage of visibility from the camera set), d is the total duration of construction activity, and t represents the t_{th} day within this duration (d) and finally V is the volume of expected as-built element. We use $P(\eta^i|\theta^i{}_T)$ to estimate progress with a threshold Γ_i. Choosing an optimal value for the threshold for each element is problematic. For example given a 10% visibility [$P(\theta^i{}_p)$] and 25% completeness of reconstruction $P(\theta^i{}_T)$,$P(\eta^i|\theta^i{}_T)$ may be susceptible to reconstruction noise/inaccuracy. Therefore it may not be reported as detected. This selection is particularly difficult, because (1) to achieve a desired accuracy, for different element types/ materials, different *thresh* needs to be used; (2) Progress monitoring task is subjective by nature and needs an expert's opinion as to whether it has taken place or not. Thus we use a machine learning model to estimate such dynamic thresholds in a principled way. We express Γ_i as:

$$\Gamma^i = f(\theta_p(t), P(\eta|\theta_T), t/d, T^i, \Psi(t), \delta, thresh, \epsilon_{Reg}, \epsilon_{Rec}) \tag{9}$$

where t is construction activity duration from t=0 to d, T^i is the element type (e.g., column, beam), $\Psi(t)$ is the visual appearance of the element i's surface (e.g., concrete, steel), δ voxel resolution, *thresh* the voxel consistency threshold and finally ϵ_{Reg} and ϵ_{Rec} are the accuracy in registration of *as-planned* model over point cloud and the accuracy of underlying reconstruction algorithms. As shown in Table 1, we assume there is minimal error in registration and the underlying mechanisms of as-built reconstruction. Γ^i can be learned by casting the problem into linear classification problem. That is by learning the hyper-plane that separates the two classes in a multi-dimensional feature space. The feature space is defined by $P(\eta^i|\theta^i{}_T)$, $\theta_p(t)$, t/d, T^i, $\Psi(t)$, δ, and *thresh*. The two classes are E^i=1 and E^i=0. The optimal hyper-plane can be learned in a supervised fashion using a linear SVM (Fan *et al*. [9]). Once the classifier is learned, given a new omeasurement of progress $P(\eta^i|\theta^i{}_T)$ along with the measured features ($\theta_p(t)$, t/d, T^i, $\Psi(t)$, δ and *thresh*) we can establish whether progress has occurred or not by feeding observation into the classifier and retaining the output.

4. Experiments and results

In order to validate our proposed reconstruction pipeline as well as automated progress detection over arbitrary set of daily photographs and in presence of occlusions, we conducted experiments on three different image collections. These datasets were collected under different viewpoints and lighting conditions and were used for evaluating this task. These datasets, which include 152 and 255 building elements repesectively, are two sets of 112 and 160 images from *RH* project (RH#1 and RH#2) and a set of 288 images from a Student Dining (*SD*) construction project. The images are all taken at the basement level of project while significant occlusion is observed in both *RH* cases as the images were not taken from inside the basement area. Rather they were all taken along a side walk of the project (See Fig. 4-b). We synthetically reduced the resolution of these images to 2MPixel to test robustness of our approach to the image resolution. We initially set voxel resolution to $\frac{1}{5}$ *ft* (0.06*m*). Fig. 8(a1 to 4) illustrates the results of dense reconstruction for *RH* presented in Fig. 5b and (a5 to 8) present the results for *SD* project.

4.1. Comparison of detection accuracy

In our experiments, we analyze performance by (1) Recall: The fraction of truly recognized *as-planned* elements relevant to the total number of elements . (2) Precision: The fraction of truly recognized *as-planned* elements relevant to the total number of elements that are recognized with progress. In our approach, the *SVM* kernel machine classifies progress with a binary value (progress/no progress). We trained the *SVM* model over *RH#1* 112 images which have significant occlusion. The hyper-plane dynamically learned though experiments reports that if the expected observable area is less than 20% of the the *as-planned* element and the volumetric reconstruction is only able to reconstruct the expected areas up to 50%, this element should not be recognized. The performance of training is cross-checked by asking two field engineers and a superintendent to label training

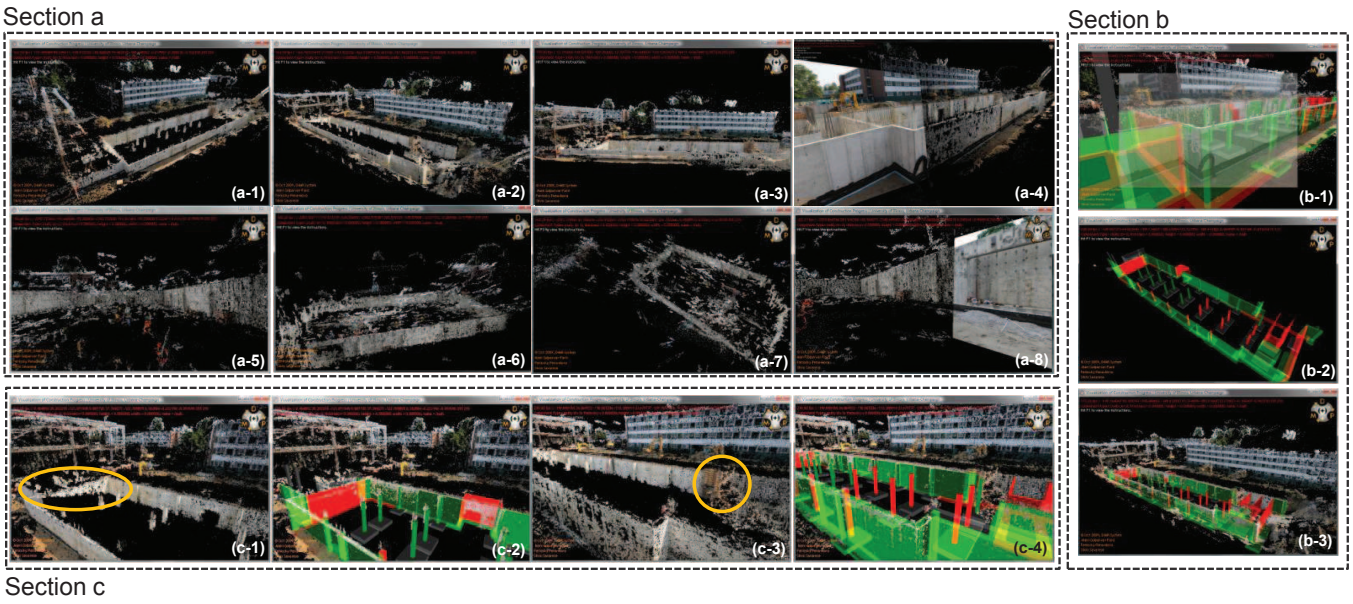

Figure 8. (a1 – 4) Dense as-built reconstruction for RH as in Fig.4-b; (a4-8) Dense as-built reconstruction for SD; (b-1 to 3) Progress deviations for RH color-coded and visualized in the D^4AR environment. (c1 & 2) False Positive – the formwork should not be detected as evidence of progress. (c3 & 4) False alarm – the wall should be detected for progress though it is severely occluded.

classification results. The accuracy of training was experienced to be 87.50%. We tested performance of the classifier on *RH2* 160 and *SD* 288 image collections. It is noted that the datasets used for the experiments are from actual construction image photo collections which makes their application very appealing for training and testing purposes. The results of average accuracy for our experimental datasets are presented in Table 2. We also studied the relationship between expected/observable progress. As shown false positives mostly fall under 20% visibility (Fig. 9c). We also studied how occlusion is affecting the accuracy. The results are showcased in Fig. 9d and it indicates that although we use severly occluded images, yet our SVM model is resulting in high precisions.

Project	# of elements	# of images	accuracy
RH1	152	112	87.50%
RH2	152	160	82.89%
SD	255	288	91.05%

Table 2. Avg accuracy of SVM detection for testing samples.

We also studied precision-recall and TP/FP. Fig. 9a-b illustrate experimental results. The precision is promising and shows our approach is not sensitive to formation of the hyper-plane. Finally we represent changed/unchanged elements with red/green. Fig. 8(b-1 to 3) shows the result of our progress detection for RH#2 dataset. The behind and on-schedule elements are color-coded with red and green accordingly. For those elements that progress is not reported, we color them in gray. Fig. 8(c-1 to 4) show examples of false positive and false alarms in our detection. As observed in Fig. 8(c-3/4), since our model does not contain appearance information (e.g., operational details), concrete form is falsely detected as finish of a concrete element. In Fig. 8(c-1/2) highlighted wall should be detected, but due to occlusions it is not properly reconstructed and consequently not detected.

5. Conclusions and summary

A method for progress monitoring using site images and 4D *as-planned* models is presented. In our approach, images are widely distributed, yet robustly generate dense point clouds. The initial point cloud is registered with other point clouds as well as the *as-planned* model, generating an integrated 4D *as-built* and *as-planned* model. Our *as-built* and *as-planned* voxel coloring demonstrates high accuracy in labeling construction scenes for occupancy and visibility. The SVM model shows promising results in detecting progress. Application of our system is observed to minimize the time required for *as-built* data collection and *as-planned* data extraction; removing subjectivity of progress detection through a systematic detection; and finally interactive visualization to minimize the time required for progress coordination leading to a better decision-making for project control. We need to conduct more conclusive experiments on the dense reconstruction and progress detection (especially at building interiors). We also need to incorporate visual appearance information $[P(\eta^i)]$ (As in Fig. 8c-

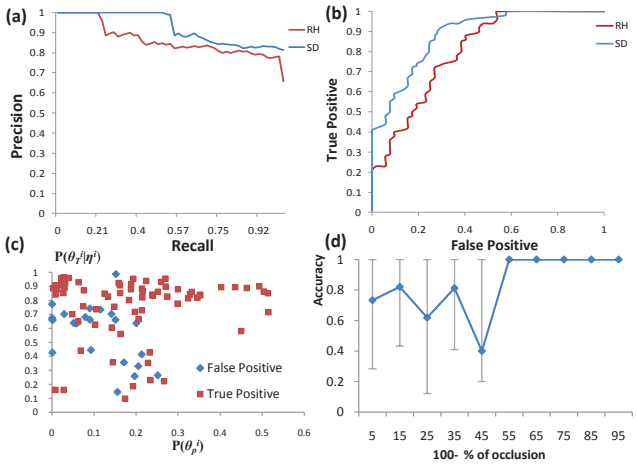

Figure 9. (a) Precision-Recall graph; (b) True-Positive/False-Positive graph; (c) Expected progress vs. Expected observable regions for RH #1 testing dataset; (d) Accuracy of detection vs. % of occlusion.

3) to consider element surface appearance in progress monitoring.

6. Aknowledgements

We acknowledge the support of the NSF grants 1054127, 060858, and CMMI-0800500. We also thank Turner Construction Co. for their help with data collection, as well as Min Sun, Wongun Choi, Ying-Ze Bao, and Byung Soo Kim for their help and support with pre-reviewing of this paper.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. *IJCV*, 2009. 2

[2] F. Bosche. Automated recognition of 3d cad model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *J. of Advanced Eng. Informatics*, 2009. 1, 2, 3

[3] F. Bosche, C. T. Haas, and B. Akinci. Automated recognition of 3d cad objects in site laser scans for project 3d status visualization and performance control. *ASCE J. Computing Civil Eng.*, 23(6):391–404, 2009. 1, 2

[4] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 78(2). 2

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. Computer Graphics Proceedings, pages 11–20, 1996. 2

[6] S. Du, N. Zheng, S. Ying, Q. You, and Y. Wu. An extension of the icp algorithm considering scale factor. 2007. 4

[7] C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM Handbook*. Wiley Publishing, 2008. 1

[8] S. El-Omari and O. Moselhi. Integrating 3d laser scanning and photogrammetry for progress measurement of construction work. *J., Automation Construction*, 18(1):1–9, 2008. 2

[9] R. Fan, K. W. Chang, C. J. H. C.J., X. Wang, and C. Lin. Liblinear: A library for large linear classification. *J. of Machine Learning Research*, 2008. 6

[10] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *ICCV*, pages 80–87, 2009. 2, 3

[11] M. Golparvar-Fard, F. Pena-Mora, and S. Savarese. D4ar- a 4-dimensional augmented reality model for automating construction progress data collection, processing and communication. *J. of ITCON*, 14(1):129–153, 2009. 1, 2

[12] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 3

[13] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. of Optical Society*, (4):629–642, 1987. 3

[14] Y. Ibrahim, T. Lukins, X. Zhang, E. Trucco, and A. Kaka. Towards automated progress assessment of work package components in construction projects using computer vision. *J. of Advanced Eng. Informatics*, 2009. 2, 3

[15] R. Laganire, H. Hajjdiab, and A. Mitiche. Visual reconstruction of ground plane obstacles in a sparse view robot environment. *Graphical Models*, 68(3):282 – 293, 2006. 1

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3

[17] T. Lukins and E. Trucco. Towards automated visual assessment of progress in construction projects. 2007. 2, 3

[18] D. Nister. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):756–770, 2004. 3

[19] Pena-Mora, Aziz, Golparvar-Fard, Chen, Plans, and Mehta. Review of emerging technologies to support urban resilience and disaster recovery. In *Proceedings of Urban Safety of Mega Cities in Asia*, pages 1–10, Beijing, China, 2008. 1

[20] Pollefeys and etal. Detailed real-time urban 3d reconstruction from video. *IJCV*, 2008. 2

[21] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. *CVPR*, 2008. 2, 3

[22] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2), 1999. 4

[23] N. Snavely, R. Garg, S. Seitz, and R. Szeliski. Finding paths through the world's photos. volume 27, pages 11–21, 2008. 2, 3

[24] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. pages 835–846, 2006. 2

[25] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment-a modern synthesis. *Int. Vision Alg.*, pages 153–177, 2004. 3

[26] L. Zebedin, Bauer, Karner, and Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *ECCV (4)*, pages 873–886, 2008. 2

[27] X. Zhang, N. Bakis, T. C. Lukins, Y. Ibrahim, S. Wu, M. Kagioglou, G. Aouad, A. Kaka, and E. Trucco. Automating progress measurement of construction projects. *J. of Automation in Construction*, 18(3):294–301, 2009. 2