

# Automated material selection based on detected construction progress

Tyler Stephans<sup>1</sup>, Austin McClymonds<sup>1</sup>, Robert Leicht<sup>1</sup> and Alan R. Wagner<sup>1</sup>

<sup>1</sup>The Pennsylvania State University, USA

tbs5111@psu.edu

## Abstract -

**One of the many ways in which automation may help the construction industry is on-site material management. This paper presents an automated process where materials are selected for staging by detecting construction progress from site images. The materials are then delivered to their respective workface locations by a robot. The effectiveness of the material selection process is assessed using a simulated and physical construction site. We demonstrate that our process is successful under a number of different conditions and environments. Our system contributes to the feasibility of autonomously managing materials on a construction site and reveals potential avenues for future research.**

## Keywords -

Automated material delivery; Progress detection; BIM; Robot

## 1 Introduction

This paper seeks to create a process that allows a robot to: a) recognize the current state of a construction project, b) predict which materials will be needed, and c) deliver those materials to their appropriate workface. The work presented here focuses on the first two of these components. Material management is an important part of any construction project. Consisting of the transportation, storage, identification, handling, and usage of materials, material management is directly related to efficiency, safety, and waste during construction. One case study found that the cost of inefficiencies resulting from improper material management was 5.7 times greater than the cost of implementing material management [1]. With respect to safety, Lipscomb et al. observed that 11.5% of slipping and tripping accidents on construction sites occurred while the worker was carrying an object [2]. Additional studies have found that human error, accidents, and damage during transportation to the site or on the site were all sources of construction material waste [3].

The use of robots for construction site material delivery could maximize the tracking of materials while also minimizing waste. Robots used at night or moving along dedicated pathways could also increase safety, provided

no workers are present at night. Hence, the goal of this research is to develop a system that can automatically detect construction progress and deliver the materials that will be needed for the next step in a construction plan.

The task of automating the tracking of construction progress is referred to as automated progress monitoring [4]. Research in this area typically compares a 3D model of the structure to be built (as-planned model) with another model generated using sensor data of the construction site (as-built model). The as-planned model is derived from a building information model (BIM), and contains physical and functional information about the structure. By comparing these models, individual elements of the structure such as walls, beams, or columns are recognized and their progress assessed. The expected progress is derived from a critical path schedule. In this paper however, no expected amount of progress is prescribed, and progress monitoring is used as a means to automate material selection for delivery via mobile robots. As such, we refer to this problem as automated progress *detection* as opposed to automated progress *monitoring*.

This paper presents an automated process where materials are selected for staging based on images collected by a mobile robot on a construction site. The system developed for selecting materials makes use of a pre-existing technique for automated progress monitoring. The output of this process may then be used to instruct a robot to deliver materials to specific workface locations where the material will be needed. The primary contribution of this paper is an automated process for guiding the timely distribution of construction materials.

In the next section we review related work before describing the process itself. Next, experiments demonstrate our process followed by an analysis of the experimental results. The paper concludes with directions for future work.

## 2 Related work

Recent advances in robotics have begun to make the use of construction robots a possibility [5]. Mobile robots however must cope with the cluttered and dynamic environments found in construction sites, which research in obstacle detection [6] and localization using BIM [7]

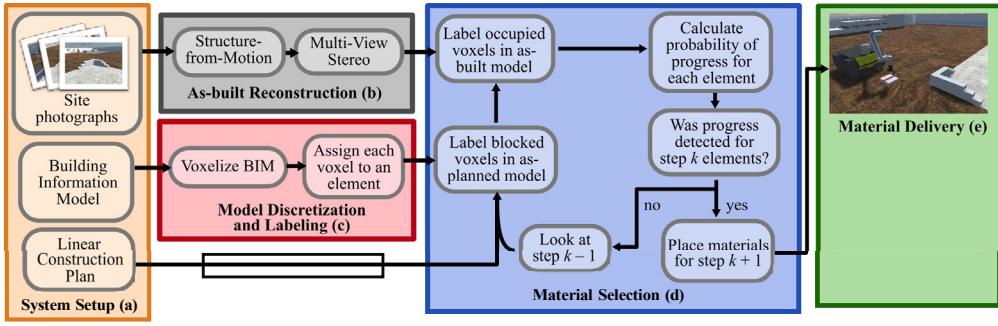


Figure 1. System Diagram: Input photographs, BIM, and construction plan are used to generate a list of materials to be delivered by a robot.

have begun to address. The capability of robots to place construction elements has also been investigated [8], and mobile manipulators capable of moving materials on construction sites have been demonstrated [9].

A great deal of research has focused on automated progress monitoring for construction sites [4, 10]. Attempts have been made to automate construction progress monitoring using either laser scanners [11, 12] or images [13, 14, 15] of the site to generate an as-built model. The model must then be aligned/registered with the BIM in order to compare them for determining progress. This typically involves a degree of manual manipulation, although studies have demonstrated automated techniques [16]. Some methods utilize machine learning to help in monitoring progress [13, 11, 15]. Others incorporate temporal information to take advantage of relationships between construction elements [11, 14]. Bayesian models [13] and computer vision have also been used to recognize construction materials [15].

Our work uses a method developed by Golparvar-Fard et al. [13]. In their process, unordered site images and a 4D BIM are used to automatically monitor the construction progress of a site. Images are used to reconstruct the as-built model by first using structure-from-motion (SfM) to create a sparse point cloud of feature points. Multi-view stereo (MVS) techniques are then used to create a dense point cloud. Once the model is aligned with the BIM, the model space is discretized into voxels to support the inference of progress over small areas of the construction site. The voxels are labeled for both the as-planned and as-built models based on their occupancy and visibility from the site images. These labels are then used to determine the probability of observing progress for each construction element by applying a Bayesian probabilistic approach. Finally, by rephrasing the problem as a linear classification problem, a support vector machine (SVM) is trained to calculate a probability threshold for each element and classify it as either having progressed or not progressed.

For the purposes of this paper, Golparvar-Fard et al.'s process is simplified by removing the SVM.

### 3 System description

Our process is depicted in Figure 1 and consists of five parts. These are system setup, as-built reconstruction, model discretization and labeling, material selection, and material delivery.

#### 3.1 System setup

The inputs for our system can be found in box (a) in Figure 1, and are 1) site photographs, 2) a BIM, and 3) a linear construction plan. These inputs serve as the basis for generating the as-planned and as-built point cloud models, and when combined with a construction plan allow for materials to be selected via progress detection.

The critical characteristic of the BIM input is that it is a 3D model composed of uniquely named construction elements. Each element also has a material label, which can be used for generating a list of materials for a robot to deliver. As a medium for design, BIM does not always contain the level of detail assumed by this paper. For example, a brick wall would be a single 3D model with the individual bricks visually represented using a mapped texture. Before discretizing the BIM, we specify a volume of interest on the model space using Cartesian bounds  $\mathbf{b}_{\text{lower}}$  and  $\mathbf{b}_{\text{upper}}$ . All 3D objects outside these bounds are removed from the as-planned model, such that only construction elements important for progress detection remain. The result is an as-planned model consisting of the set of elements  $\theta = \{\theta_1, \dots, \theta_{N_{\text{elements}}}\}$ .

To generate a list of materials for delivery, a plan is also required. When a construction schedule is generated, it is targeted for craft labor tasks, and thus it normally contains temporal information in the form of a task schedule, where multiple tasks may be performed concurrently. Our process assumes a more granular linear plan consisting of

a series of steps  $k = 1, \dots, N_{\text{steps}}$ , where one or more construction elements are placed at each step. In this paper, the word *step* specifically refers to a step in a linear plan and not some component of the material selection process.

The input images must provide enough information to discern the current step based on the plan without having to infer based on relationships between construction elements. In general, this means that the collection of images used will feature at least one element placed during the previous step and one space where an element will go in the following step. Additionally, the images must be devoid of any unplanned occlusions blocking construction elements. The only permissible occlusions are other construction elements. When collecting images in simulation, the precise location and orientation for each image was also provided. This was done to remove the need for manually aligning the as-built point cloud model with the as-planned model by selecting corresponding points, as was done for the physical experiments.

### 3.2 As-built reconstruction

In this paper, as-built reconstruction is the process of using site images to create a 3D point cloud of the construction site. As indicated in Figure 1 by box (b), the as-built point cloud model, denoted from here onward as as-built PC, is needed to detect progress and determine which construction elements have been placed.

The process of reconstructing the as-built PC from the provided camera images and poses is handled by COLMAP [17], an open source structure-from-motion (SfM) and multi-view stereo (MVS) pipeline. SfM compares identical features for a scene across multiple images to generate the scene's 3D geometry. The outputs from SfM are then used with MVS to map normal and depth information onto the images. This results in a dense point cloud  $P_b$  representing the as-built construction site, and is used as the as-built PC. Notice that the subscript  $b$  indicates a variable's association with the as-built PC.

### 3.3 Model discretization and labeling

The as-planned model is discretized into finite, fixed volume voxels of width  $\delta$  for labeling (see Figure 1 box c). By dividing the as-planned model into smaller volumes, visibility can be reasoned per voxel for each element, thereby informing the progress detection which voxels are expected to contain points from  $P_b$ .

To voxelize the as-planned model, each element is converted into a point cloud  $P_e$ , where  $e = 1, \dots, N_{\text{elements}}$  is an index for each construction element. Doing so allows each voxel to be associated with a particular element based on the  $P_e$  with the most points inside the voxel. This adds a requirement that each  $P_e$  must have the same density  $\rho$  of equally spaced points.

The as-planned model space is discretized based on combinations of  $P_e$  for all  $e$  with

$$V = \bigcup_{e=1}^{N_{\text{elements}}} P_e, \delta, \mathbf{b}_{\text{lower}}, \mathbf{b}_{\text{upper}} \quad (1)$$

where  $\text{fVoxelGrid}()$  generates voxels of width  $\delta$ , bounded between limits  $\mathbf{b}_{\text{lower}}$  and  $\mathbf{b}_{\text{upper}}$ , from a union of  $P_e$  point clouds. The result is a voxel grid  $V$  where voxels  $V_j$  for  $j = 1, \dots, N_{\text{voxels}}$  are only placed where at least one point exists. Each  $V_j$  is provided two labels, for the as-planned model and as-built PC, and will have values of either empty ( $E$ ), occupied ( $O$ ), or blocked ( $B$ ).

### 3.4 Material selection

The as-built PC is compared with the as-planned model at various steps per the linear plan. By starting with the final step and analyzing in reverse order, as shown in box (d) in Figure 1, the step which matches the as-built PC is detected. This process informs the current construction progress and determines what materials need to be delivered for the subsequent step. Due to this relationship, the terms *progress detection* and *material selection* are used mostly interchangeably in this paper.

To begin detecting progress, a new voxel grid  $V_k \subseteq V$  for step  $k$  must be created. For details on how a voxel  $V_j$  is assigned to a voxel grid  $V_k$ , see Figure 2.

#### 3.4.1 Label blocked voxels

Next, progress detection requires the blocked voxels in the as-planned model to be labeled. Only voxels expected to contain points from  $P_b$  should be checked for occupancy, so the voxels labeled as blocked are discounted.

Voxels are traversed and labeled in a manner similar to [13]. Readers are directed to the method outlined by Golparvar-Fard et al. for further details on voxel traversal and labeling blocked voxels based on the as-planned model. However, unlike Golparvar-Fard et al., we do not use the radiance of a voxel's projected pixels to evaluate if a voxel is occupied. This removes the need for a visibility constraint, and thereby enables a unique voxel label solution to be found even when considering one image at a time. Therefore, the order of voxel traversal is defined by

$$d_{i,j} = \|\mathbf{x}_j - \mathbf{x}_i\| \quad (2)$$

where  $\mathbf{x}_j$  and  $\mathbf{x}_i$  are the center of  $V_{k,j}$  and image  $i$  respectively in the world coordinate frame. The voxels in  $V_k$  are then traversed in order of increasing distance  $d_{i,j}$ .

The as-planned label for  $V_{k,j}$  is specified as ( $O_p$ ) or ( $B_p$ ) based on the number of reprojected pixels of  $V_{k,j}$  with an evaluation of 1 as described by [13]. Here, the subscript  $p$  indicates a variable's association with the as-planned model. Further,  $V_{k,j}$  can be labeled as ( $B_p$ ) if  $d_{i,j}$

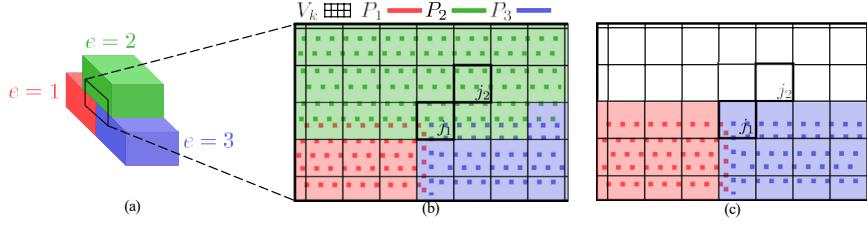


Figure 2. Model discretization and labeling visualized using a structure consisting of three elements: (a) Red, green and blue construction elements with  $e = 1, 2, 3$  respectively. Current step is  $k$ . (b) 2D zoom of voxel grid  $V_k$  with point clouds  $P_1$ ,  $P_2$  and  $P_3$ . Voxels  $j_1$  and  $j_2$  have more points from  $P_2$  than  $P_1$  or  $P_3$ , leading to  $\mathbf{e}_{j_1,k} = (2, 3, 1)$  and  $\mathbf{e}_{j_2,k} = (2)$ . (c) 2D zoom of voxel grid  $V_{k-1}$  where element 2 was removed from the voxel grid and from  $\mathbf{e}_{j_1,k-1}$  and  $\mathbf{e}_{j_2,k-1}$ , giving  $\mathbf{e}_{j_1,k-1} = (3, 1)$  and  $\mathbf{e}_{j_2,k-1} = ()$ . Voxel  $j_1$  now belongs to element  $e = 3$ .

is greater than some distance threshold as well. This was done primarily to reduce computation time.

#### 3.4.2 Label occupied voxels

The voxels not labeled as blocked in the as-planned model, are labeled for occupancy based on the as-built PC. By counting the number of occupied and empty voxels for each element, the probability of observing each element is determined.

Because the assumption was made that no unplanned occlusions were present in the images used to reconstruct the as-built PC, all voxels labeled with  $(B_p)$  are also labeled  $(B_b)$ . Each voxel  $V_{k,j}$  not labeled  $(B_b)$  is checked to see if it contains a point from  $P_b$ . If so, then it is labeled occupied  $(O_b)$ . However, when  $V$  is generated, it is possible for some voxel faces to be very close to an element face. Due to the limited precision of the reconstruction process, scenarios where points in  $P_b$  fall just outside of a voxel can be common. To address this problem for some empty voxel  $V_{k,j}$ , all voxels adjacent to  $V_{k,j}$ , which are not part of  $V_k$ , are also checked for occupancy. Here, two voxels are considered adjacent if they share a common face. If a point is found in any of the adjacent voxels, then  $V_{k,j}$  is labeled  $(O_b)$ . Otherwise it is labeled empty  $(E_b)$ .

#### 3.4.3 Plan traversal for material selection

Given a step in the linear plan, the probability of observing each element in that step is calculated. The first step found where every element placed in that step is expected to be observed, is said to be the current state of construction. By knowing the construction progress, the materials required are known and may be delivered.

Progress detection is performed to recognize element  $\theta_{k,e}$  in  $\theta_k$ , starting with  $k = N_{\text{steps}}$ . As with [13], progress is evaluated by comparing a probability to a threshold. Provided evidence of occupancy  $\eta_e$ , Golparvar-Fard et al.

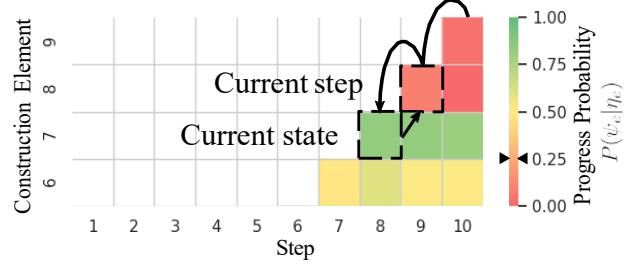


Figure 3. Example progress board from physical experiment. Columns represent steps of the plan, and rows represent construction elements organized by step. Each band of color within a cell, is the  $P(\psi_e|\eta_e)$  of element  $e$  for whatever step  $k$  the system is on. No color (white) indicates that element is not visible. Here, each step involves placing one element, so there is a single band of color per cell. The process of determining the current step is visualized by the black arrow and outlined cells.

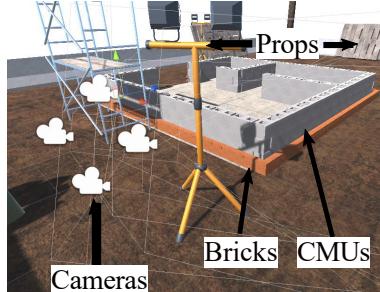
define the probability of  $\psi_e$ , the event of observing an expected element, as

$$P(\psi_e | \eta_e) = \frac{Q_e}{O_b + E_{\theta_{k,e}}} \quad (3)$$

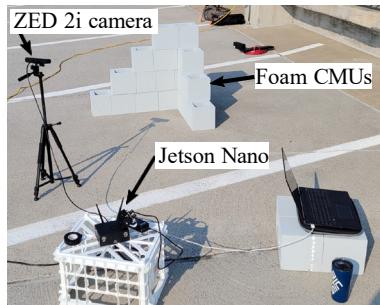
Given our assumptions, and a linear construction plan rather than a general schedule,  $P(\psi_e|\eta_e)$  is satisfactory to detect  $\theta_{k,e}$ . If  $P(\psi_e|\eta_e)$  is greater than the threshold, then the element is marked as having progressed.

After evaluating the progress for all elements in  $\theta_k$ , the current step is said to be  $k + 1$  if progress was detected for all elements placed in step  $k$ . Otherwise, progress detection is performed on step  $k - 1$  as shown in Figure 1.

We now introduce the **progress board** in Figure 3 to better explain this process. A progress board functions as



(a) Simulation experimental setup



(b) Physical experimental setup

Figure 4. BIM, experiment environment, and example image capture for physical and simulation experiments

a recording of all steps the material selection analyzes. To find the elements placed on some step, we search the grid diagonal, moving from the top right cell to the bottom left, as shown in Figure 3.

Once the current step is determined, the elements for that step are chosen to be delivered by a robot. Because each element is associated with a material, the result is the robot receiving a list of materials to deliver.

## 4 Experiments

### 4.1 Simulation experiment

The BIM used in simulation was designed to resemble the foundation of a small structure which can be seen in Figure 4(a). Images were captured of the simulation at an interim step in the plan, and progress detection was used to determine the current step. Using the progress detection results, a list of materials for delivery was output. This process was evaluated at every step in the plan.

The BIM was generated as an FBX file, with unique names for each construction element. Materials were distinguished using specific tags in the element names. For this research, the two materials used were standard bricks and concrete masonry units (CMUs). The linear plan was designed such that elements were distributed across steps

Table 1. Camera properties

camera	width (px)	height (px)	focal length $f$ (px)	principal pt. $o_x$ (px)	principal pt. $o_y$ (px)
simulation	2048	1536	1525.1	1024	768
ZED 2i	2208	1242	1063.6	1107.2	632.1

based on wall face and course. In total, the plan contained 57 steps to place 447 elements.

Unity was used for the material selection simulation. Unity offers a capable physics engine, cameras for simulating image capture, and importation of many assets into the simulation. Additional textures and construction props were added, as shown in Figure 4(a), to make the simulation look more realistic and improve the performance of the as-built reconstruction. It should be noted that if a prop was ever occluding a construction element in the camera view, it was removed for that as-built reconstruction. The lighting in the simulation was configured to represent noon at Penn State in early summer to balance realistic lighting and ease of reconstruction.

Uncompressed site images were captured using camera objects arranged in Unity. The cameras were arranged in groups of 4 for each reconstruction, as seen in Figure 4(a). The camera intrinsics were chosen to resemble a smartphone camera and are provided in Table 1. It should be noted that the simulated camera parameters differ from the physical camera because the simulation experiments were conducted before the ZED 2i camera was acquired. These camera parameters and poses were imported into COLMAP [17].

During reconstruction, COLMAP's default "low" quality settings were used to reduce computation time. Our process performed the material selection and returned a list of materials for delivery. When discretizing the BIM, the voxel width  $\delta$  and point cloud density  $\rho$  were chosen to be 0.0254 m (1 in) and 15,500 pts/m<sup>2</sup> respectively. The threshold for progress to be detected for an element was set to 0.25, and the distance threshold for a voxel to be considered visible was set to 4 m. All parameters were chosen empirically.

### 4.2 Physical experiment

The BIM used for the physical experiments consisted of a small stack of CMUs arranged in a corner shape. The completed construction can be seen in Figure 4(b). Construction was conducted on top of a parking deck to best approximate the large concrete flooring found on many construction sites. Images were then captured using the Stereolabs ZED 2i camera while varying the current step,

Table 2. Material selection results across all elements in  $\theta_k$  for  $k$  = current state, where  $P_k(\psi_e|\eta_e)$  is the probability of an element placed during step  $k$  being observed

Setting	# reconstructions	Min $P_k(\psi_e \eta_e)$	Avg. $P_k(\psi_e \eta_e)$	Max $P_k(\psi_e \eta_e)$	# mistaken steps	# bricks not selected	# CMUs not selected
morning	8	0.429	0.744	0.995	0	0	0
afternoon	18	0.502	0.830	1	0	0	0
evening	8	0.497	0.702	0.881	0	0	0
simulation	57	0.289	0.831	1	0	0	0
sim. occluded	2	0	0.585	1	2	1	0

time of day, and position of camera. Using the collected images, progress detection was performed to assess if the correct materials could be selected based on progress.

Once again, the BIM for the physical experiments was generated as an FBX file. The actual CMUs used were foam blocks (Figure 4(b)). Foam CMUs were chosen for ease of handling in future robot experiments. Unlike the simulation experiments, the linear plan for the physical experiments only had 1 CMU placed per step, with a total of 10 steps.

The physical experiments were varied in three ways: lighting, camera position, and construction step. Lighting was varied by taking images in the morning, afternoon, and late evening. Additional variation due to weather occurred naturally. Sets of images were captured around the CMUs at approximately 90° apart. For each lighting condition and position, the CMU corner was constructed to steps 6 and 10. Images were also captured and processed for every step, but these were only gathered in the afternoon at a single position around the CMU corner. In total, 34 variations of physical experiment were performed.

A ZED 2i camera was mounted on a tripod to capture uncompressed images, and its rectified intrinsics can be found in Table 1. The ZED 2i is factory geometrically calibrated, and so no manual calibration was performed prior to the experiments. For each as-built reconstruction, the tripod's center column was extended in 0.06 m increments a total of 0.3 m to capture left and right images. This yielded 12 images per reconstruction, which will now be referred to as a set of images. A Python script was written to record the images, and was run using an NVIDIA Jetson Nano Developer Kit-B01. An example of an experiment setup is in Figure 4(b).

When processing the real images, COLMAP was set to its default “high” settings to increase the final point density of the reconstruction. Although the poses of the camera were not measured for use in COLMAP, the camera intrinsics were still provided using the ZED camera Python API. Again, SfM and MVS was performed using COLMAP’s command line interface.

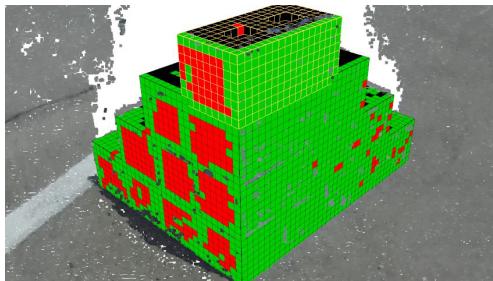
After reconstruction, because the camera poses were

not recorded, the as-built PC had to be aligned to the same coordinate frame as the BIM. CloudCompare’s Align tool was used to generate a transformation matrix using the CMU vertices as a reference point. The matrix was used to rotate and scale both the as-built PC and reconstructed camera poses during progress detection. No changes were made to the material selection process. All parameters were left at their previous values.

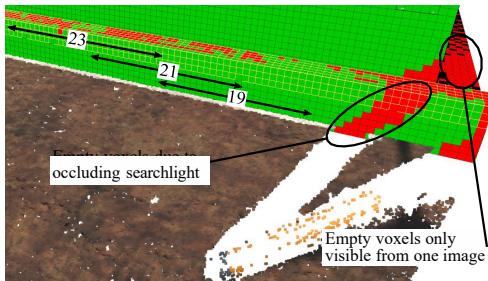
## 5 Results and discussion

Evaluation of our system in both simulation and physical environments resulted in the successful selection of the correct materials for delivery across all of the construction steps, camera locations, and lighting differences tested. As seen in Table 2, all minimum probabilities were above the threshold of 0.25 when there were no occlusions, so no materials were improperly selected. In practice, this means the material selection system will determine the correct materials to deliver for the construction site, provided ample lighting and images, and the workface is free of obstructions. While the system was successful, some experiments do demonstrate how failures could occur.

Correct materials are not selected for delivery when elements placed during the current state of construction have too many voxels labeled as empty. In the experiments conducted through this research, voxels were incorrectly labeled as empty primarily because of poor lighting and occlusions. A representative example of the effect of lighting on progress detection can be seen in Figure 5(a). Even though the sun was nearly directly overhead, the brick faces in the shade were reconstructed noticeably worse than the faces in the sun. Table 2 shows the probability  $P_k(\psi_e|\eta_e)$  that an element placed during the current state is observed was generally higher during the afternoon than morning or evening, although no materials were mistakenly selected. However, Table 2 shows both simulated reconstructions where occlusions were introduced resulted in the wrong steps being determined. Figure 5(b) depicts voxels labeled as empty (red) due to a spotlight leg occluding some of the bricks from the cameras. This resulted in the current state



(a) Foam CMU corner at noon



(b) Simulated experiment with an occlusion

Figure 5. As-built reconstruction point cloud and voxel labels: red = empty, black = blocked, green = occupied, yellow grid lines = current construction elements

being detected as 19 instead of 23. The reason for this can be observed in Figure 6. Looking at step 23 (actual current state) on the diagonal, red bands can be seen in the cell. There are multiple bands in each cell because multiple elements are placed per step. The red bands indicate that some elements were not detected, and explains why progress detection continued searching for the current step. Further down the diagonal, the elements in steps 21 and 19 were detected fairly well. In Figure 5(b) it is shown that these steps represent the brick courses directly below step 23. Therefore, step 19 was deemed the current state because it was reconstructed enough to be detected as opposed to step 21. The consequence of this error is that the number of bricks selected for delivery would be based on step 19 instead of step 23. For these two steps, this is only the difference between delivering 8 and 7 bricks, but the error could be greater or smaller if the mistake was between a different pair of steps.

Two other limitations of this material selection system are its reliance on detailed construction schedules and as-planned to as-built model alignment. As stated in subsection 3.1, this paper uses a BIM and linear plan with an uncommon level of detail. For a system such as the one presented here to be used on a real construction site, detailed schedules for element placement will need to be

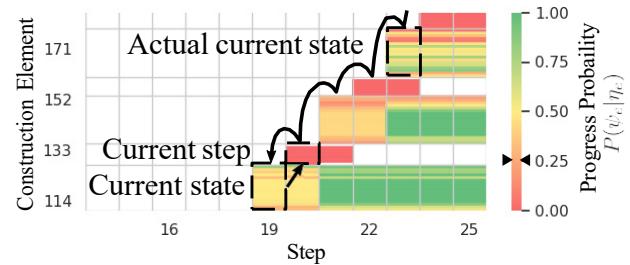


Figure 6. Progress board for simulation at step 23

incorporated into BIM, or methods will need to be developed to reverse engineer the information from current models. Assuming such scheduling data is available, care must still be placed into aligning the as-built PC with the BIM coordinate frame, as errors will be magnified by small misalignments for large models. This concern can be reduced by collecting more images and by automating as-built model alignment using GPS or registering images with on-site control markers [16].

## 6 Conclusion

This paper has demonstrated a process for automated material selection based on detecting construction progress from robot captured images. Our system was evaluated in simulation and at a simulated, physical construction site. The material selection process was shown to be robust to lighting and is capable of detecting any step in a linear plan despite limitations related to model alignment and occlusions. This paper contributes a progress detection system capable of selecting the necessary materials for further construction progress. The system presented serves as a template for future advancement, which will allow workers to spend less time documenting progress and moving materials, and more time helping the construction industry reach its productivity goals.

Future research aims to address the noted limitations and investigate the challenges of detecting progress on a real construction site. These challenges include scalability, large varieties of construction elements, non-linear construction schedules, and the generally more complex and varied environment that a construction site poses.

## 7 Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Number 1928626. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to thank Adith Calix for her help in conducting the physical experiments.

## References

[1] H. Randolph Thomas, Victor E. Sanvido, and Steve R. Sanders. Impact of material management on productivity—a case study. *Journal of Construction Engineering and Management*, 115(3):370–384, 1989. doi:10.1061/(ASCE)0733-9364(1989)115:3(370).

[2] Hester J. Lipscomb, Judith E. Glazner, Jessica Bondy, Kenneth Guarini, and Dennis Lezotte. Injuries from slips and trips in construction. *Applied ergonomics*, 37(3):267–274, 2006.

[3] B. A. G. Bossink and H. J. H. Brouwers. Construction waste: Quantification and source evaluation. *Journal of Construction Engineering and Management*, 122(1):55–60, 1996. doi:10.1061/(ASCE)0733-9364(1996)122:1(55).

[4] Marianna Kopsida, Ioannis Brilakis, and Patricio Antonio Vela. A review of automated construction progress monitoring and inspection methods. In *Proc. of the 32nd CIB W78 Conference 2015*, pages 421–431, 2015.

[5] Mi Pan, Thomas Linner, Wei Pan, Hui-min Cheng, and Thomas Bock. Influencing factors of the future utilisation of construction robots for buildings: A hong kong perspective. *Journal of Building Engineering*, 30:101220, 2020.

[6] Pileun Kim, Jingdao Chen, Jitae Kim, and Yong K. Cho. SLAM-driven intelligent autonomous mobile robot navigation for construction applications. In *Workshop of the European Group for Intelligent Computing in Engineering*, pages 254–269. Springer, 2018.

[7] JeeWoong Park, Yong K. Cho, and Diego Martinez. A BIM and UWB integrated mobile robot navigation system for indoor position tracking applications. *Journal of Construction Engineering and Project Management*, 6(2):30–39, 2016.

[8] Chen Feng, Yong Xiao, Aaron Willette, Wes McGee, and Vineet R. Kamat. Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites. *Automation in Construction*, 59:128–138, 2015.

[9] Petr Štibinger, George Broughton, Filip Majer, Zdeněk Rozsypálek, Anthony Wang, Kshitij Jindal, Alex Zhou, Dinesh Thakur, Giuseppe Loianno, Tomáš Krajiník, and Martin Saska. Mobile manipulator for autonomous localization, grasping and precise placement of construction material in a semi-structured environment. *IEEE Robotics and Automation Letters*, 6(2):2595–2602, 2021. doi:10.1109/LRA.2021.3061377.

[10] Jun Yang, Man-Woo Park, Patricio A. Vela, and Mani Golparvar-Fard. Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future. *Advanced Engineering Informatics*, 29(2):211–224, 2015.

[11] Changmin Kim, Hyojoo Son, and Changwan Kim. Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, 31:75–82, 2013.

[12] Frédéric Bosché. Automated recognition of 3D cad model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced engineering informatics*, 24(1):107–118, 2010.

[13] Mani Golparvar-Fard, Feniosky Peña-Mora, and Silvio Savarese. Automated progress monitoring using unordered daily construction photographs and IFC-based building information models. *Journal of Computing in Civil Engineering*, 29(1):04014025, 2015. doi:10.1061/(ASCE)CP.1943-5487.0000205.

[14] Alexander Braun, Sebastian Tuttas, André Borrmann, and Uwe Stilla. Automated progress monitoring based on photogrammetric point clouds and precedence relationship graphs. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 32, page 1. IAARC Publications, 2015.

[15] Jacob J. Lin, Jae Yong Lee, and Mani Golparvar-Fard. Exploring the potential of image-based 3D geometry and appearance reasoning for automated construction progress monitoring. In *Computing in Civil Engineering 2019: Data, Sensing, and Analytics*, pages 162–170. American Society of Civil Engineers Reston, VA, 2019.

[16] Sebastian Tuttas, Alexander Braun, André Borrmann, and Uwe Stilla. Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4D BIM. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 85(1):3–15, Feb 2017. ISSN 2363-7145. doi:10.1007/s41064-016-0002-z.

[17] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.