Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites

Chen Feng a,⁎, Yong Xiao a, Aaron Willette b, Wes McGee b, Vineet R. Kamat a

a Department of Civil and Environmental Engineering, University of MI, Ann Arbor, USA
b College of Architecture and Urban Planning, University of MI, Ann Arbor, USA

ABSTRACT

Unlike robotics in the manufacturing industry, on-site construction robotics has to consider and address two unique challenges: 1) the rugged, evolving, and unstructured environment of typical work sites; and 2) the reversed spatial relationship between the product and the manipulator, i.e., the manipulator has to travel to and localize itself at the work face, rather than a partially complete product arriving at an anchored manipulator. The presented research designed and implemented algorithms that address these challenges and enable autonomous robotic assembly of freeform modular structures on construction sites. Building on the authors' previous work in computer-vision-based pose estimation, the designed algorithms enable a mobile robotic manipulator to: 1) autonomously identify and grasp prismatic building components (e.g., bricks, blocks) that are typically non-unique and arbitrarily stored on-site; and 2) assemble these components into pre-designed modular structures. The algorithms use a single camera and a visual marker-based metrology to rapidly establish local reference frames and to detect staged building components. Based on the design of the structure being assembled, the algorithms automatically determine the assembly sequence. Furthermore, if a 3D camera is mounted on the manipulator, 3D point clouds can be readily captured and registered into a same reference frame through our marker-based metrology and the manipulator's internal encoders, either after construction to facilitate as-built Building Information Model (BIM) generation, or during construction to document details of the progress. Implemented using a 7-axis KUKA KR100 robotic manipulator, the presented robotic system has successfully assembled various structures and created as-built 3D point cloud models autonomously, demonstrating the designed algorithms' effectiveness in autonomous on-site construction robotics applications.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Several studies have argued that among all industries, construction has seen a significant productivity decrease over the last several decades compared to other industries [30]. Construction has also been documented to have some of the highest rates of workspace injuries and fatalities [6]. Automation and robotics in construction (ARC) has the potential to relieve human workers from repetitive and dangerous tasks, and has been extensively promoted in the literature as a means of improving construction productivity and safety [3]. Compared to the tangible benefits of automation and robotics identified by the manufacturing industry, the construction industry is still exploring feasible and broadly deployable ARC applications [3]. This can be attributed to several commercial and technical challenges. From the commercial perspective, the fragmented and risk-averse nature of the construction industry leads to little investment in ARC research causing construction to lag behind other industries [31]. On the other hand, as described next, there are several technical complexities inherent in construction that have contributed to hindering the successful development and widespread use of field construction robots.

1.1. Technical challenges

1.1.1. Unstructured construction environments

Automated and robotized manufacturing facilities are typically considered as structured environments, since both the machines and evolving products either stay in their predefined locations or move on predesigned and typically fixed paths. In general, such environments do not change shape or configuration during the performance of manufacturing tasks, making the enforcement of tight tolerances possible [23]. In contrast, construction sites can typically be considered unstructured since they are constantly evolving, and dramatically changing shape and form in response to construction tasks. Building components are moved around without fixed paths or laydown/staging areas. Various physical connections are established through improvisation in response to in-situ conditions, making tight tolerances hard to maintain and enforce [24].
1.1.2. Mobility of construction manipulators

In manufacturing, factory robotics typically involves robotic platforms that are generally stationary (or have limited linear mobility) and partially complete products that arrive at robot workstations and precisely localize themselves in the robots’ base reference frames. Precision is achieved by controlling the pose of the moving (and evolving) product, and the robots themselves are programmed to manipulate the products through fixed trajectories. Thus, from a mobility and cognitive perspective, a factory robot has little responsibility and autonomy. Control is achieved by enforcing tight tolerances in moving and securing the product in the manipulator’s vicinity. However, this spatial relationship is reversed in construction. A construction robot has to travel to its next workplace (or be manually set up there), perceive its environment, account for the lack of tight tolerances, and then perform manipulation activities in that environment. This places a significant mobility and cognitive burden on a robot intended for construction tasks even if the task itself is repetitive.

This discussion highlights that factory-style automation on construction sites requires development of robots that are significantly more mobile and perceptive when compared to typical industrial robots. Such on-site construction robots have to be able to semantically sense and adjust to their unstructured surroundings and the resulting loose tolerances. This paper proposes a new high-accuracy 3D machine vision metrology for mobile construction robots. The developed method uses fiducial markers to rapidly establish a local high-accuracy control environment for autonomous robot manipulation on construction sites. Using this method, it is possible to rapidly convert a portion of a large unstructured environment into a high-accuracy, controllable reference frame that can allow a robot to operate autonomously.

The rest of the paper is organized as follows. Related work is reviewed in Section 1.2. The authors’ technical approach is discussed next in detail in Section 2. The experimental results of both assembly and scanning are shown and discussed in Section 3. Finally, in Section 4, the conclusions are drawn and the authors’ future work is summarized.

1.2. Previous work

1.2.1. Robotic manipulators in construction

The construction community has pursued research on robotic manipulators for several decades: for example, Slocum and Schena [33] proposed the Blockbot for automatic cement block wall construction; Pritschow et al. [29] identified the needs and requirements of a bricklaying robot for masonry construction and developed a control system for such robots.

A large portion of the construction robotic manipulator research focused on mechanics and control of specific construction activities. Fukuda et al. [13] discussed the mechanism and the control method of a robotic manipulator in construction based on human–robot cooperation. Yu et al. [37] proposed an optimal brick laying pattern and trajectory planning algorithm for a mobile manipulator system, with computer simulation to test its efficiency. Hansson and Servin [17] developed a semi-autonomous shared control system of a large-scale manipulator in unstructured environments, with a forwarder crane prototype to test its performance. Chung et al. [7] proposed a new spatial 3-degree-of-freedom (DOF) parallel type master device for glass window panel positioning and adjust to their unstructured surroundings and the resulting loose tolerances. This paper proposes a new high-accuracy 3D machine vision metrology for mobile construction robots. The developed method uses fiducial markers to rapidly establish a local high-accuracy control environment for autonomous robot manipulation on construction sites. Using this method, it is possible to rapidly convert a portion of a large unstructured environment into a high-accuracy, controllable reference frame that can allow a robot to operate autonomously.

1.2.2. 3D as-built modeling in construction

3D as-built modeling (e.g., BIM) plays an important role in a wide range of civil engineering applications. This modeling process usually starts with collecting 3D point clouds of sites of interest. Paul et al. [27] utilized a 6DOF anthropomorphic robotic arm to get the 3D mapping of a complex steel bridge with a laser range scanner. Brilakis et al. [5] outlined the technical approach for automated as-built modeling based on point clouds generated from hybrid video and laser scanning data. Akula et al. [2] explored different 3D imaging technologies, e.g., 3D image system, image based 3D reconstruction and Coherent Laser Radar scanner, to map the locations of rebar within a section of a railway bridge deck in order to assist a future drill operator in making real-time decisions with visual feedback. Zhu and Donia [39] investigated the advantages and drawbacks of RGBD cameras in as-built indoor environments modeling, with evaluation on the accuracy of collected data, the difficulty of automatic scan registration and the recognition of building elements, demonstrating RGBD camera’s potential in as-built BIM modeling. In this research, the automatic planning, scanning and registration of point clouds obtained from a 3D camera mounted on the manipulator are achieved with the visual marker-based metrology and the manipulator’s internal encoders.

Once 3D point clouds are obtained, CAD-like geometric models can be generated. For example, Son et al. [34] automatically extracted 3D pipeline models from laser scanning data based on the curvature and normal of point clouds; Han et al. [16] proposed an automated and efficient method to extract tunnel cross sections from terrestrial laser scan (TLS) data. While this research focuses more on the automatic registration of different frames of point clouds, the resulting registered point clouds could be input into such algorithms to generate semantically meaningful CAD-like geometric entities for as-built BIM.

1.2.3. Robotic manipulators in architecture

Recently the architectural design community has also shown an increased interest in industrial robotics, with many academic programs investing in their own robotic work cells. Capitalizing on the machines’ inherent flexibility, they have leveraged the industrial robot as a development platform for the exploration and refinement of novel production techniques in which material behavior is intrinsically linked to fabrication and assembly logics. As part of the general ecosystem of industrial robotics, computer vision systems have begun to play an increasingly important role in these research initiatives, with a number of architectural research groups developing interfaces for accessible hardware such as the Microsoft Kinect.

Initially the majority of architectural robotic research utilizing computer vision has revolved around its application at the micro scale, using a vision feedback system to make incremental adjustments to a robotic strategy based upon local variations. Examples of this include Dierich et al.’s [8] research into poured aggregate structures at the Institute for Computational Design at University of Stuttgart and Dubor and Diaz’s [9] project Magnetic Architecture from the Institute for Advanced...
Architecture of Catalonia, two robotics projects which use the Microsoft Kinect to provide information on local variations from the intended design geometry, which is then used to generate incremental adjustments in succeeding operations. While beneficial as a means to adjust for material variation and machine error, these implementations are not robust enough for the in-situ robotics due to the complexities of construction sites.

Acknowledging the limitations of processes developed within the safety of the research laboratory, architects have slowly begun to explore application of computer vision at the macro scale for adaptive path planning. At the forefront of this work has been the research conducted at the ETH Zurich led by Fabio Gramazio and Matthias Kohler. Heavily invested in the potential of construction site robotics, their work has included the development of hardware/software solutions that allow industrial robotics to dynamically adjust their operations at both the micro and macro levels. This research is best exemplified in their ECHORD project, in which an eight-meter-long module wall was assembled by an ABB robot mounted to a mobile track system [19]. Constructing a stacked module wall along a gestural path captured by the robot’s computer vision system, this mobile robot also used the same system to reposition itself on the construction site (expanding its operable reach envelope) and make local adjustments based on topographic variation. Without using an extensive sensor suite like the one used in ECHORD, the robotic platform described in this paper successfully built similar module walls purely based on perceived information from a single camera; moreover, as-built 3D point clouds of the assembled outcome can be readily obtained if the robotic platform is equipped with a 3D camera on the manipulator, for purposes like as-built BIM generation or construction progress documentation.

2. Technical approach

2.1. System overview

The developed robotic in-situ assembly and scan system consists of several major components, as shown in Fig. 1. The assembly workflow of the system consists of an offline design process and an online building process. During the offline process, a designer models the intended structure in 3D, which is then analyzed and validated by the assembly planner, outputting an assembly plan for the online process. The online assembly process uses a fixed camera mounted on the base of the robot for providing images to the pose estimator to detect staged building components and estimate their poses, and more importantly localize the robot itself in the local building reference frame. Having computed this information, the plan achiever then sequentially transforms each step from the automatically generated assembly plan into an executable command, which can be interpreted by the robot controller and subsequently executed by the robot manipulator. In addition, the visualization component also receives the information generated by the pose estimator as well as the robot’s real-time pose feedback from the controller, to simultaneously represent the actual on-site assembly process into a 3D virtual environment for improved monitoring.

Similarly for the scan workflow, based on design information, the scan planner can feed a scanning plan to the plan achiever, controlling the robot manipulator to stop and capture 3D images at a series of desired poses using the 3D camera mounted on the manipulator. Since the robot platform is aware of both its base’s pose relative to the local building reference frame through our marker-based metrology, and the 3D camera’s pose relative to its base through its internal encoders, the scan register can then readily transform each frame of captured 3D images into the same local building reference frame, resulting a 3D point cloud describing the construction site.

2.2. Calibration of pose estimator

Before introducing the details of other components of the system, it is important to discuss how the pose estimator is calibrated, since this is crucial to the level of assembly/scan accuracy that the system can achieve. This process includes two steps to be finished before system operation: intrinsic and extrinsic calibration of the camera. Intrinsic calibration involves estimating the camera’s focal length, principle point’s position on the image plane, and distortion parameters. On the other hand, extrinsic calibration aims to determine: 1) the relative 6-DOF pose of the camera in the robot’s base coordinate frame for assembly, and 2) the relative 6-DOF pose of the 3D camera in the robot’s tool coordinate frame for scan registration. It must be noted that both intrinsic and extrinsic calibration are one-time processes, as long as the camera/3D camera is fixed-focus and rigidly mounted in the robot’s base/tool coordinate frame (e.g., fixed installation on the robot’s base/end-effector).

2.2.1. Intrinsic parameters

Unlike the popular plane-based camera calibration method [38] implemented in OpenCV2 and Matlab Calibration Toolbox,5 the authors chose to calibrate the camera using a 3D rig, which is similar to the classic calibration in photogrammetry. This 3D rig was made by attaching N Aprillights [26] to two intersecting planes forming a 90° angle (as shown in the top of Fig. 2) so that the 3D coordinate X of each Apriltag’s center could be readily measured.

The process of calibration was then simply taking a sequence of M images of the rig and inputting them into the author-developed camera calibration tool,6 which takes advantage of the Apriltag detection algorithm to detect the 2D image coordinate U of each tag center and establish correspondence with X. Then the initial camera intrinsic and extrinsic parameters can be obtained through Direct Linear Transform (DLT) [18] and subsequently optimized by bundle adjustment [35], which minimizes the re-projection error by tuning both intrinsic (K) and extrinsic (R and t) camera parameters as following Eq. (1):

$$\arg \min _{K, \{R, t\}} \sum _{i=1}^{M} \sum _{j=1}^{N} \left\| U_{ij} - \pi (K (R C_{0} + t) X_{ij}) \right\| ^{2}, \quad (1)$$

where the perspective division function $\pi : \mathbb{R}^{3} \rightarrow \mathbb{R}^{2}$ converts a 2D homogeneous coordinate into a 2D Cartesian coordinate.

Benefitting from the high corner detection accuracy of Apriltag as well as the 3D rig, this intrinsic calibration produces more robust and repeatable results than the alternatives mentioned above.

2.2.2. Extrinsic parameters

Once the camera’s intrinsic parameters are calibrated, the camera’s relative pose in the robot’s base coordinate frame, $T_{c} = \begin{bmatrix} R_{c} & t_{c} \end{bmatrix}$, can be estimated using an extrinsic calibration marker containing L (L > 1) Aprillights with known size and spacing.

As shown in Fig. 3(a), $T_{c}$ can be composed from the two other poses, $\nu_{m}$, the robot’s pose in the extrinsic calibration marker’s coordinate frame, and $T_{m}$, the marker’s pose in the camera coordinate frame, by $T_{c} = (T_{m} \circ \nu_{m})^{-1}$. This extrinsic calibration process of the camera used for assembly consists of the following steps:

1) Fix the marker in the camera’s field of view;
2) Manually control the robot manipulator to pinpoint at least 4 non-collinear points on the marker and record their 3D coordinates $X$ in the robot’s base coordinate frame; also measure their local 3D

---

2 http://dfab.arch.ethz.ch/web/e/forschung/198.html.

4 http://www.vision.caltech.edu/bouguetj/calib_doc/.

5 Available at https://code.google.com/p/cvxmlg/4apriltag.
coordinates $m\mathbf{X}$ in the marker’s reference frame (by setting all Z coordinates to be zero);

3) Take an image from the camera and detect the L Apriltags’ 4L corners’ 2D image coordinates $\mathbf{U}$.

With this information collected, the $m\mathbf{T}$ can be estimated using the well-known rigid body registration [4] from 3D point set $m\mathbf{X}$ to $m\mathbf{X}$, while the $m\mathbf{t}$ can be estimated by decomposing the homography between $m\mathbf{X}$ and $\mathbf{U}$ using the previously calibrated camera intrinsic parameter $K$ [10,38].

In order to improve the extrinsic calibration’s accuracy, a non-linear optimization of $m\mathbf{t}$ is also performed in addition, since during the homography decomposition, a polar decomposition is performed to get a valid rotation matrix $R_m$, which causes the result to be non-optimal. This optimization, as shown in Eq. (2), can be done by tuning the initial $m\mathbf{t}$ to minimize the re-projection error:

$$\arg\min_{m\mathbf{R}, \ m\mathbf{t}} \sum_{j=1}^{P} \left\| \mathbf{U}_j - K (R_m m\mathbf{X}_j + t_m) \right\|^2.$$  (2)

While the extrinsic calibration of the 3D camera used for scanning could be done in a similar manner, a faster and more efficient solution is to perform the classic robot hand–eye calibration: by moving the robot end-effector to different poses relative to its base ($B_i$, $B_j$) and correspondingly estimate through Apriltags the extrinsic calibration marker’s pose relative to the 3D camera reference frame ($A_i$, $A_j$), equations $A_iB_j = A_jB_i$ can be established where $X = m\mathbf{T}$, representing the 3D camera’s pose relative to the robot’s tool coordinate frame, as shown in Fig. 3(b). After rearranging such equations into the form of $AX = XB$ where $A = A_i^{-1}A_j$ and $B = B_jB_i^{-1}$, the calibration can be solved by Tsai and Lenz’s [36] methods. It is worth noting that the previous extrinsic calibration of the camera for assembly could also be solved as a hand–eye calibration problem, even though the camera is not fixed on the manipulator’s end-effector, to further simply the calibration process.

2.2.3. Calibration validation

The calibration can be validated by the following procedure:

1) Fix the extrinsic calibration marker to a new pose that is different from the one used in the calibration;

2) Measure, in robot’s base frame, the 3D coordinates $\mathbf{X}$ of a set of P corner points on the marker (e.g., the 4L corners used previously);

3) Take an image and find out the corresponding 3D coordinates $\mathbf{X}$ in the camera reference frame using Apriltags;
reinforcing the implications of William Mitchell’s statement that “architects tend to draw what they can build, and build what they can draw” [25].

2.3.2. Assembly plan generation and simulation

Given the final positions and orientations of all the building blocks in the design, the assembly plan is generated and written into a text file stored for the plan achiever to process later during the online building phase.

This assembly plan file contains a list of sequential instructions for the robot manipulator to build the designed structure. Each line in the file corresponds to an instruction. For example, the following plan file segment will instruct the robot manipulator to first grab a building component named “block0” directly from above (line 1–4), then lift it vertically up for 500 mm (line 5) and finally place it at its destination in another reference frame named “building” (line 6–8):

```
Gripper 0
Goto block0 0 0 500 0 0 0
Goto block0-12-10-10 0 0 0
Goto building 200.00-300.00 500.00-63.92 0.00 0.00 0.00
Goto building 200.00-300.00 19.05-63.92 0.00 0.00 0.00
Gripper 0
```

Currently, 3 types of instructions are implemented in the system:

1. **Gripper 0/1**
   - Control the manipulator’s gripper to open (0) or close (1);

2. **Goto reference_frame x y z a b c**
   - Control the manipulator to move to a new pose \((x, y, z, a, b, c)\) in the reference frame, in which the \((x, y, z)\) is the new position and \((a, b, c)\) specifies the new orientation as three Euler angles in “ZYX” order;

3. **Shift x y z a b c**
   - Control the manipulator to incrementally move by \((x, y, z, a, b, c)\).

This assembly plan can also be simulated in Rhinoceros to check if there exists any self-collision between the robot manipulator and the wall during the building process, as shown in Fig. 4.

2.4. Vision-based plan achiever

2.4.1. Rapid setup of building reference frame

As previously mentioned, the reversed spatial relationship of product and manipulator on construction sites poses a significant challenge for autonomous mobile robots. This is notably different from typical autonomous manufacturing spatial configurations, where robots’ bases are either stationary or have finite mobility, and materials/components
can be readily staged at fixed locations within the manipulators' static workspaces. In contrast, for mobile robots to autonomously perform building tasks on unstructured construction sites, their bases require significant mobility, and consequently their manipulators' workspaces are not fixed with respect to the construction site. In order to complete building tasks at the correct locations and assemble materials into their intended poses, a robotic system must be able to establish the accurate 6-DOF transformation between the robot's base and the building reference frame at all times. As pointed out in [32], this requires the localization accuracy to be at least at centimeter level, which could not be easily achieved using state-of-the-art Visual SLAM style techniques for mobile robots.

In order to address this challenge, the authors propose a convenient and accurate solution using planar marker-based pose estimation [10, 26], as shown in Fig. 5. By 1) attaching fiducial markers at appropriate locations on-site where building tasks are to be performed, 2) surveying their poses $^{m}T_b$ in the building reference frame using a total station, and 3) storing these poses inside the system's database, a mobile robot can readily estimate its base's pose $^{r}T_b$ inside the building reference frame using Eq. (4) whenever its on-board camera detects such a marker, based on previous calibration results:

$$^{r}T_b = \left(^{m}T_b^c \cdot {T_m^c}^{-1}\right)^{-1}.$$  

2.4.2. Conversion from plans to commands

With the information input from the pose estimator, the vision-based plan achiever starts to execute the assembly plan generated beforehand, according to the following procedure:

1) Read a single plan step (i.e., one line) from the assembly plan file;
2) Wait until all the poses needed to convert this step into a building command become available;
3) Convert this step into a command that is executable by the robot controller;
4) Send the command to the robot controller;
5) Wait for the controller to complete the command;
6) Repeat this process unless all plan steps are completed, i.e., the plan is achieved.

It must be noted that the core step of this procedure is the conversion from a plan step to an executable command. This is because the poses stored in the previously generated assembly plan are not completely specified in the robot's base reference frame. Recall that every pose in the "Goto" step is specified in a "reference_frame" relative-ly. Specifying all steps in the assembly plan in the robot base reference frame is not possible because: a) during the design phase the designer conceives all component locations in the building reference frame; b) the robot's base is expected to be mobile during the building phase; and c) more importantly, the building components will be arbitrarily transported and staged in the building reference frame in the vicinity of the robot manipulator's workspace. This, in fact, is one of the core differences between on-site construction automation and manufacturing automation.

In the authors' approach, this conversion is facilitated by the aforementioned rapid setup of the building reference frame using markers. As long as the pose estimator can detect and report the transformation between the on-board camera and the marker used to specify the building reference frame, the poses in the plan steps can be readily converted to the robot base frame using equations similar to (4). Similarly, by attaching markers on the building components, the robot manipulator can detect and clasp them autonomously after the corresponding plan steps are converted.

2.5. As-built point clouds from 3D camera

Due to the operation and maintenance requirement, nowadays construction project owners are often more satisfied if given access to not only an as-designed BIM but also an as-built BIM of a project. As reviewed previously, a common practice is to start with various 3D scanners such as TLS to create 3D point clouds of the built environment. These point clouds are then manually or semi–automatically abstracted into CAD-like 3D geometric objects conveying semantic meanings, such as walls, floors, doors, windows and utility pipelines.

Since more often than not limited by the 3D scanner's field of view, a single scan cannot fully cover a construction project's outcome, multiple scans need to be carefully planned, performed and then registered into a single reference frame. As pointed out by previous work, the automation of such planning and registration becomes important to improve the productivity as well as the quality of the final point clouds.

Facilitated by the proposed visual marker–based metrology and the robot's internal encoders, the process to automatically plan, scan and register different frames of 3D point clouds becomes readily available on the proposed robotic platform with minimal effort of mounting and calibrating a 3D camera on the manipulator, when reusing existing
components for autonomous assembly. Whenever a new frame of 3D point clouds $P_i (\forall i = 1, 2, 3 \ldots)$ is captured (and thus expressed naturally in the 3D camera’s reference frame), controlled by the scan planner, they are transformed into the building reference by Eq. (5):

$$bP_i = b_Tr_tr_tr_cP_i; (\forall i = 1, 2, 3 \ldots)$$

where $T_c$ is the 3D camera’s pose relative to the robot’s tool reference frame, i.e., hand–eye calibration result obtained in Section 2.2.2, $T_t$ the robot’s tool’s pose relative to its base reference frame, maintained by the robot’s internal encoders. As noted earlier, the robot’s base pose in the building reference frame, $b_tr_r$, is estimated through our marker-based metrology in Eq. (4).

3. Results and discussion

3.1. Assembly experiments

The authors implemented the designed algorithms into a robotic system using a 7-axis KUKA KR100 robotic arm with sub-millimeter level accuracy, a Point Grey Firefly MV camera, and a laptop with an Intel i7 CPU, connected through the Robot Operating System (ROS). Each component in Fig. 1 except for the assembly planner and robot manipulator is a process corresponding to a ROS node. The camera node sends images of size 1280 pixels by 960 pixels to the pose estimator that implements the Apriltag detection algorithm in C/C++. The plan achiever was implemented in Python. The robot controller was also developed using Python to send and receive control signals via Ethernet through KUKA’s native Robot Sensor Interface (RSI). Inside this controller, a 6-DOF PID control algorithm was employed to drive the robot manipulator (with a two-finger gripper) to its destination pose when executing commands from the plan achiever. The involved inverse kinematics computations are performed inside the KUKA manipulator’s controlling middleware.

In the first phase of experiments, the robot was tasked with assembling a section of a curved wall, as designed in Section 2.3.1. The design was shown in Fig. 1. The building components were a set of 170 × 70 × 20 mm³ medium-density fiberboard (MDF) blocks, each affixed with two different 56 × 56 mm² Apriltags. The building reference frame was setup by 4 different 276 × 276 mm² Apriltags. The overall goal of the experiment was to test the robot’s ability to autonomously build the designed wall.

The system was first calibrated and validated using the methods discussed earlier. The validation residual calculated using Eq. (3) was found to be less than 1 mm. With the accurately calibrated intrinsic and extrinsic parameters, the online building process proceeded smoothly. The building blocks were affixed with smaller markers, which decreased the building block localization accuracy to centimeter level (2 cm) during the clasping process. The error was however compensated by the tolerance of the gripper.

A working cycle of the autonomous building process was shown in Fig. 6(a)–(d). Since the pose estimator was constantly monitoring and updating the poses of each marker, the system was naturally capable of automatically adapting to pose changes on-site. As shown in Fig. 6(e)–(h), when a building block’s pose changed, the robot manipulator was automatically able to pick it up at its newest location. A video recording of the experiment can be found online at the following URL: http://youtu.be/fj7AXRpi97o. A fully assembled three-layer curved wall approximately 1.5 m in length, and another three-layer circular wall, are shown in Fig. 7.

3.2. Scanning experiment

The authors implemented the scan module on the same KUKA robotic arm with a Microsoft Kinect as the 3D camera, as shown in Fig. 8, with the scan planner and the scan register implemented in Matlab. The Fabrication Lab (FabLab) in the College of Architecture of the University of Michigan was scanned as an experiment demonstrating the scan module’s effectiveness. Using the methods described in Section 2.5, each frame of newly captured Kinect RGBD image of 640 × 480 pixels is first converted into a frame of colored point cloud.
and then transformed into the same building reference frame. The resulting final point cloud is then visualized in Point Cloud Library (PCL),9 as shown in Fig. 9.

3.3. Limitations and future work

There are several limitations in the above implementation of the proposed vision guided robotic assembly and scanning solution that need to be overcome to facilitate its future applications in real world construction sites. These limitations include:

1) Feasibility and robustness of the marker-based pose estimation under different illumination conditions;
2) Occlusion of markers on construction sites may impede the feasibility or even performance of such systems;
3) Substantial effort may be necessary to setup such systems onsite due to requirements to survey numerous markers and register their poses into the project coordinate frame;
4) Additional effort is needed for attaching markers on each construction material/component such as blocks for the robot to recognize and pick them up;
5) The durability of markers is a critical limitation for applications in rugged environments and difficult weather conditions;
6) Lack of systematic and quantitative comparisons of such robotic system with traditional manual methods.

The first concern on marker-based pose estimation’s robustness noted above has been addressed by the authors’ in another research study with various field experiments proving the feasibility, robustness and accuracy for applying such technique in both indoor and outdoor construction environments [11].

Even though the second concern about occlusion is a common limitation of all vision based methods, it is not such a critical problem in this proposed method. This is because many occlusions of markers onsite are temporary due to moving equipment or humans. Once the mobile robotic manipulator observes those markers and estimates its own pose in the project reference frame, as long as its base stays static during the occlusion period, the robot can still accurately maintain its pose using its internal encoders.

The third concern about the workload of installing and surveying markers onsite indeed exists in the current system implementation. Yet, compared to the efforts of setting up other pose estimation and localization methods that require powered hardware infrastructure such as WLAN or UWB, or methods that only work well in open-air outdoor environments such as GPS, this effort could be a worthwhile tradeoff. Moreover, the authors are working on a novel method that integrates the markers with Structure-from-Motion technique so that there will be no need to survey these markers using specialized surveying equipment such as total station. Instead, by simply taking a set of images of those installed markers, their poses can be automatically estimated.

The authors also recognized the fourth and fifth concerns, and worked on removing the requirement of attaching markers on raw construction materials. For example, although not closely related to the

---

9 http://pointclouds.org/.
core contribution of this paper, the block detection and grasp can be achieved without markers using Kinect sensor and the author’s newly developed fast plane extraction algorithm [12], as shown in Fig. 10 and demonstrated at the following URL: http://youtu.be/CyX4Pr_xly0. For markers to be attached on indoor construction sites, even printing on papers can achieve reasonably good durability, which is very cost effective and can be efficiently and conveniently replaced if destroyed. For outdoor situations, markers can be spray painted on existing planar surfaces (e.g., walls) or manufactured wood, foam, or plastic boards.

Lastly, quantitative comparisons with tradition manual methods are a logical future direction in this research. Such comparisons will be more meaningful when this research advances to the next stage where the mobile manipulator can actually build stable structures with cement mortar, rather than in this research as the proof-of-concept stage.

4. Conclusions

This paper reported algorithms and an implemented robotic system that is able to automatically generate assembly plans from computational architectural designs, achieve these plans autonomously on construction sites, and create as-built 3D point clouds. In order to address the localization accuracy challenge, the authors proposed a computer-vision-based sub-centimeter-level metrology that enables pose estimation using planar markers. The conducted evaluation experiments used the designed robotic system to autonomously assemble various structures such as a curved wall of MDF blocks, proving the algorithms and the system’s ability to meet the accuracy requirement when building computational architecture designs. In sum, with markers, an unstructured construction site can be rapidly configured to allow autonomous mobile manipulators to localize themselves and thus perform assembly and scanning tasks. In this way, the challenges of unstructured environment and mobility can be efficiently addressed.

In addition to the long term future goals mentioned in Section 3.3, the authors’ current and planned work in this research direction in the short term is focused on continuously improving the fundamental methods used in this system along the following directions:

1) Perception: using camera on manipulator to further improve the pose estimation accuracy through multiple observations and sensor fusion; improving 3D perception to help improve the safety of the robotic system, preparing for its collaboration with human workers on-site.

2) Navigation: incorporating state-of-the-art SLAM algorithms with the proposed marker-based metrology in the system for increasing its range of autonomous movement.

3) Hardware and control: designing a suitable robotics platform with mobile base and on-board manipulator of sufficient payload capacity for indoor and outdoor construction activities that are more complex.

![Fig. 8. Scan module of the designed robotic system.](image-url)
than the block laying activity described in this paper, exploring more sophisticated control algorithms to enable such complex construction tasks.

4) Applications: extending the proposed autonomous system in other construction or architecture applications.

References


