Accurate and Adaptive In situ Fabrication of an Undulated Wall using an On-Board Visual Sensing System

Manuel Lussi¹, Timothy Sandy¹, Kathrin Dörfler², Norman Hack², Fabio Gramazio², Matthias Kohler² and Jonas Buchli¹

Abstract— In this paper we present a system for the in situ fabrication of a full-scale, load-bearing, and doubly-curved steel reinforced concrete wall. Two complementary vision-based sensing systems provide the feedback necessary to build a 12 meter long steel wire mesh as part of a novel digital building process. The sensing systems provide estimates of the robot pose, referenced to the CAD model of the building site, as well as feedback on the accuracy of the built structure over the course of construction. This second piece of information is used to adapt the building plan to compensate for system inaccuracies and material deformations which occur during buildup. In this way, the structure was successfully built with 98% of the total geometry within 2 centimeters of the designed position. To the best of our knowledge, this is the largest structure which has been built by a mobile robot using solely vision-based sensing.

I. INTRODUCTION

To date, the majority of the impact robotic systems have had on the building construction industry have been focused on pre-fabrication, with the use of industrial robots to build structures off-site in an industrial environment. In situ fabrication using mobile robots offers an exciting next step in implementing new construction methods directly on the building site. In comparison to fixed base robots in structured environments, mobile in situ construction robots have to address two major challenges to build effectively. First, they must handle the construction of large-scale building elements, exceeding their own static work space. This means that the robot has to travel during fabrication and therefore also localize itself in reference to the structure it is building in a globally consistent reference frame. Second, the robot needs to be able to handle the imperfect nature of the construction site. It needs to be able to build accurately in the presence of dirt, temperature changes, a non-flat and non-smooth ground, and imperfect dynamics resulting from non-rigid attachment to the ground. For these reasons, the in situ fabrication of fully performance-capable building elements at an architectural scale is still a significant challenge today.

We are working to address these challenges with a novel type of robot called “In situ Fabricator” (IF) [1]. We previously showed that IF can autonomously re-position itself, localize relative to its work-piece, and continue building with sub-centimeter accuracy in a laboratory environment. In this paper we expand that system to support a novel building process, constructing a full-scale wall directly on a building construction site. The constructed wall, which is S-shaped and 12 meters long, is part of the living unit DFAB HOUSE of the Empa NEST building [5] which has been constructed as a proving ground for novel digital building technologies. The wall is built using the Mesh Mould construction process. The Mesh Mould process consists of robotically fabricating a steel wire mesh which acts both as reinforcement and form-work for reinforced concrete structures. Once constructed, the wire mesh is manually filled with concrete. Mesh Mould facilitates the construction of geometrically differentiated structures in a more material and labor efficient way. This paper is focused on the sensing system required for the adaptive fabrication of the steel wire mesh directly on site. For more information about the Mesh Mould process, the reader is referred to [2].

II. PRIOR WORK

Mobile building systems have the strong advantage that the size of the structure they build is not limited by the workspace of the manipulator used for construction. Leveraging this advantage of mobile systems has proven to be difficult, however. While the earliest works in robotic bricklaying

https://www.empa.ch/web/nest/digital-fabrication
https://www.empa.ch/web/nest/
appeared over 20 years ago [3] [4], only now are industrial bricklaying systems moving towards commercial usage. More recently, Helm et al. developed a system to stack blocks over multiple re-positioning maneuvers in a semi-autonomous fashion [5]. Feng et al. showed that computer-vision-based pose estimation algorithms using fiducial markers can support an autonomous assembly process using bricks [6]. While Keating et al. showed that they can build large-scale extruded structures [7], they did not show the ability to reposition and continue building. We believe that more advanced sensing solutions are required to best exploit the mobility of these mobile building machines.

One of the main arguments for in situ fabrication is that a construction process can be adapted considering ‘as-built’ measurements during build-up, whether this relates to measurements of the existing surrounding architecture, or of the fabricated structure itself. Adaptive fabrication methods have been studied in the digital fabrication in architecture community for various construction processes [8][9][10]. These works are typically executed at the prototype scale and in more abstract building tasks, however. In this work we use adaptive fabrication methods to support a novel digital fabrication process at building scale. To the best of our knowledge this is the first time adaptive fabrication has been used within a mobile building system.

The sensing solutions most commonly used on the construction site today are laser-based referencing systems, for example the Leica Tracker[6]. Such systems have been successfully employed for on-site building construction [4] and appear to be a key part to some of the emerging commercial on-site construction systems[7][8]. While these sensors can provide high accuracy positioning information, they require line of sight to the point being tracked. This means that the space between the sensing station and the machine must remain clear and difficulties arise due to self-occlusions from the robot and ensuring visibility while building complex geometries or in small spaces. For these reasons, we did not want to rely on such sensors but instead investigate what accuracy can be achieved using on-board commodity sensors and process-specific software. We believe that this work is the first time purely camera-based sensing has been used to build a structure at full building scale.

III. SYSTEM OVERVIEW

For the purpose of constructing a steel wire mesh on site, the mobile robot IF is equipped with a proprietary weld, cut and feed end effector[11] and enhanced with the vision-based sensing system presented in this paper. The sides of the steel wire mesh are constructed as a grid of welded wires. As shown in Fig. 2 and in the accompanying video[9], layers of continuous curved wires (6 mm ribbed steel) in the vertical direction are welded to discrete straight wires (4.5 mm ribbed steel) in the horizontal direction. The two sides are bound with additional steel elements manually after a portion of the mesh has been fabricated. As shown in Fig. 2 the mesh is constructed expanding horizontally, such that the robot traverses the path of the wall once during construction. To ensure reachability over the full height of the wall, IF is repositioned manually using a joystick after building each roughly 1 m section of the mesh in the horizontal direction. Legs are then lowered to increase the robot’s stability and building continues.

To ensure a suitable interface for the ceiling/floor system which will be mounted on top of the finished wall, the target global accuracy for the constructed mesh is ±2cm. The pose of the end effector of IF must therefore be known with this accuracy throughout the building process. A camera on the IF...
end effector is used to observe rigidly mounted AprilTag [12] fiducial markers placed along the base foundation of the wall. Some of the tags serve as reference tags and their position is known relative to the foundation, while the others can be placed arbitrarily. In a one time calibration step, the relative poses of the tags along with the alignment of the tags with the CAD model is estimated. Measurements of the tag positions in images over the course of construction can then be used to find the pose of the robot in a globally consistent reference frame. The main benefit of this positioning system is that it minimizes the working area which must be clear for the robot to function. The only space that needs to be free is the space between the robot and the wall. In this way, other operations on the construction site do not interfere with IF.

Robot localization alone is not sufficient to ensure an accurate construction of the mesh. Due to internal forces that build up within the mesh during construction, even if the welds are made at the desired locations, the mesh tends to deflect away after the robot is released from the mesh. This causes not only inaccurate positioning of the mesh, especially towards the top of the wall and in areas of high curvature, but also unpredictability of where the mesh is when the robot must re-attach to it to continue building. Because these deflections are difficult to model and prevent, our strategy is to adapt the building plan on-the-fly to compensate for measured deflections in the mesh as it is built. To achieve this, we use a wide-baseline stereo camera pair located on both sides of the robotic end effector to identify the as-built position of welding nodes. This sensing system not only allows the robot to locate the mesh for re-attachment, but also allows for the measurement of the mesh contour after the robot is released from the mesh. The mesh contour measurements are then used to adapt the building plan to compensate for observed deflections. While industrial high-resolution laser scanners, such as those from Micro-Epsilon\(^\text{10}\) could have been used to make the required measurements, their size and weight made them unsuitable for use near the tip of the end effector.

The accuracy, reliability and appropriate interaction of these two sensing systems is vital to enable an accurate and collision-free construction of the mesh. The software for the sensing system is implemented in C++ and runs in Linux alongside IF’s internal controller. The CAD model and building sequence management is implemented in the architectural planning software Grasshopper for Rhino 3D\(^\text{11}\) running in Windows. Sensing system functions are called from within Grasshopper in a service-based fashion using a TCP/IP connection.

The remainder of the paper is organized as follows. The global robot pose estimation system is described in Section \(\text{IV}\) The mesh detection system is described in section \(\text{V}\) The strategy used to adapt the building plan to correct for system inaccuracies and material deformation is presented in Section \(\text{VI}\) Results are given in Section \(\text{VII}\) with the overall building accuracy shown with a high-resolution laser scan of the finished wall. Conclusions and future work are described in Section \(\text{VIII}\).

### IV. Robot Global Localization

This section describes the global robot pose estimation used during mesh construction. We use a 2448x2048 pixel PointGrey Blackfly camera with an 8mm fixed focal length lens rigidly mounted on the Mesh Mould robotic end effector. The calibration and pose estimation routines involve solving non-linear optimization problems. The cost functions optimized are presented in the respective sections. We use the Ceres Solver [13] to solve the problems using the Levenberg-Marquart algorithm.

#### A. Notation and Conventions

We use the notation \(T_{ab}\) to indicate the pose of reference frame \(b\) with respect to frame \(a\). Poses consist of a translational \((t_{ab})\) and rotational \((q_{ab})\) part, where rotations are parametrized as unit quaternions. Variables shown with a hat \((\hat{t})\) and a tilde \((\tilde{q})\) indicate measured and optimized quantities, respectively. We define the distance between two poses as

\[
T_1 - T_2 = \begin{bmatrix} t_1 - t_2 \\ q_1 \square q_2 \end{bmatrix}
\]

where the rotational distance \((q_1 \square q_2)\) between two quaternions is defined as two times the vector part of \(q_1 \times q_2^{-1}\). Optimization of rotational quantities is performed in the tangent space of the current estimate.

We define a camera measurement as the observation of a tag in a rectified image and it is comprised of four tag corner measurements. The pixel coordinates of tag corner \(e_k\) from tag \(j\) at camera pose \(i\) is written as \(\hat{m}_{C_{ij}e_k}\). We define a coordinate frame located in the middle of a tag with the \(z\) axis perpendicular to the tag plane (coordinate frame 2 in Fig. \(\text{IV}\)). The positions of the corners \(e_k=0..3\) in tag \(T_j\) (written \(T_{j,e_k}\)) are given by the tag size.

---

\(^{10}\)http://www.micro-epsilon.co.uk/2D_3D/laser-scanner/model-overview

\(^{11}\)http://www.grasshopper3d.com
B. Calibration

Before starting construction, a one time calibration of the localization system is performed. The calibration procedure is composed of three steps: estimation of the relative poses of the tags, fitting the tag poses to the CAD model, and finally calibrating the pose of the camera on the robotic end effector.

1) Tag Map Calibration: We use a batch estimation approach to estimate the relative poses of all tags with respect to the first tag \(T_{R_{0}}\), generating what we call the tag map. Moving the camera by hand, we measure all of tags in the work space from different static camera poses. After recording such a data set we optimize over the tag map, such that it minimizes

\[
\sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{k=0}^{3} \|m_{i,j,k} - \pi(T_{C,T_{0}}T_{R_{0},T_{j}}t_{T_{j},c_k})\|_{\Sigma_{m}}^{-1} \tag{2}
\]

where \(\pi()\) denotes the projection of a point in space to the rectified image plane. \(\Sigma_{m}^{-1}\) indicates that we weight the distance between a pixel measurement and its projection with the inverse covariance of the camera sensor model (we use 1 pixel standard deviation for both the x and y directions). Note that we must also optimize for the camera poses \(T_{C,T_{0}}\) from which the images were taken, however these values are not used after the calibration is complete.

2) World Frame Calibration: The goal of this procedure is to align the world reference frame defined in the CAD model of the wall with the tag map estimated above. This is achieved with the use of what we call reference tags, which are printed precisely and mounted accurately with respect to the CAD model on the foundation plate of the wall. This foundation plate was installed by an external company prior to fabrication of the mesh. 30 tags are equally spread over the length of the wall, with 6 of them serving as reference tags. \(T_{W_{R_{0}}}\) is optimized such that it minimizes

\[
\sum_{j=0}^{n} \|T_{W_{R_{0}}}T_{R_{j}} - T_{W_{R_{0}}}T_{T_{0},T_{j}}\|_{S_{F}}^{-1} \tag{3}
\]

\(T_{W_{R_{0}}}T_{R_{j}}\) is the world pose of reference tag \(R_{j}\) defined in the CAD model and \(S_{F}^{-1}\) indicates that we weight each term with the inverse covariance of a pose measurement \(T\) (we use standard deviations of 0.005 meter for the translational part and 0.02 rad for the rotational part). Note that the tag map \(T_{T_{0},T_{j}}\) was optimized in the previous step and is now constant. Once complete, the world pose of every tag, referenced to the CAD model, is given as \(T_{W_{R_{j}}} = T_{W_{R_{0}}}T_{T_{0},T_{j}}\).

3) Camera Pose Calibration: As a final calibration step, we estimate the pose of the camera with respect to the robotic end effector \(T_{E_{C}}\). The camera is mounted on the end effector and images are taken with different manipulator configurations while the base of the robot is fixed. We use the same cost function as in (2), but with the projection term in the form:

\[
m_{C_{i,j},c_k} = \pi(\pi(\pi(T_{E_{C}})^{-1}T_{E_{C},R_{i}})^{-1}T_{R_{i},T_{j}}t_{T_{j},c_k}) \tag{4}
\]

where \(T_{R_{i}}\) is the end effector pose with respect to the robot base for image \(i\), as given by the arm’s odometry and kinematics. We assume that noise from the arm odometry is negligible. As in (2) we must also optimize for the pose of the robot base with respect to the measured tags \(T_{R_{j}}\) but this is not used after calibration is complete.

C. Robot Pose Estimation

After calibrating the system, we can now solve for the world pose of the robot base \(T_{W_{R}}\) in the same way as in (2) by using the following projection term:

\[
m_{C_{i,j},c_k} = \pi((T_{E_{C}})^{-1}(T_{E_{C},R_{i}})^{-1}(T_{R_{i},T_{j}})^{-1}T_{T_{j},c_k}) \tag{5}
\]

\(T_{E_{C}}\) and \(T_{W_{R_{j}}}\) are parameters estimated during the calibration procedure which are now constant. \(T_{E_{C},R_{i}}\) is once again the pose of the end effector with respect to the robot base. During construction, after the IF base is repositioned, we take 4 images from different arm configurations at targeted locations (see Section IV-D), detect all of the tags in the images, and solve for \(T_{W_{R_{j}}}\). This base pose is then used to generate the arm commands to position the end effector for construction.

D. Tag Accuracy Measurements

To evaluate the accuracy of the proposed tag-based localization approach, the construction site tag setup was simulated in our lab and the system calibration and base pose estimation routines were performed while recording ground truth measurements with a Nikon iGPS system. The iGPS system gives pose measurements at 40 Hz with sub-millimeter nominal accuracy. The first two steps of the calibration procedure were used to build the tag map in the iGPS system’s reference coordinate frame. The third calibration step was used to estimate the pose of the camera with respect to the iGPS tracked coordinate frame. Setting \(T_{R_{E_{i}}} = I\), we could then compute the camera pose in the iGPS coordinate frame with (5).

Results are shown in Fig. 5. An accurate pose estimation is dependent on many factors: camera calibration (we apply a standard open-CV monocular camera calibration using a 8x6 checkerboard with 4 cm squares), light conditions, tag size accuracy, camera resolution, and system calibration accuracy. After calibrating our system, we found that the system could consistently reach a global positioning accuracy of approximately 1.6 cm and 0.5 degrees. To further improve system accuracy, we found that the tag measurements should be taken from a similar distance and angle as they were taken during the calibration procedure, ideally also appearing in a similar region of the image frame. By more carefully choosing the poses from which the tag measurements are taken, the system global accuracy falls below 0.5 cm and 0.15 deg. While this indicates that some of the sensing system’s unmodeled errors were over-fit during the calibration step, the flexibility afforded by mounting the localization camera on the robot end effector makes it easy to generate localization measurements similar to those use for calibration to decrease the impact of those over-fit parameters. Because the
system accuracy was observed to vary with the position of the measured tags in the image plane. Camera calibration is most likely one of the main sources of error. Future work is planned to investigate more accurate camera distortion models and camera calibration procedures.

V. WIRE DETECTION

In order to build the wire mesh with sufficient accuracy, the system needs to be able to identify and react to deflections in the mesh which build up during the construction process. The wire detection system enables this by identifying the positions of welding nodes. Such a welding node consists of one discrete and one continuous wire (see Fig. 2) which we represent as two line segments in 3D. To detect such a welding node, a wide-baseline stereo camera pair is mounted on both sides of the end effector. Blinders are additionally mounted on these sides to reduce the influence of background clutter on the wire detection algorithm (see Fig. 5). Using the robot localization described in the previous section, the end effector is positioned at the expected welding node location. We then use the CAD model of the mesh geometry to locate the fabricated target welding node in the stereo image pair. For this system, we use two center-tilted 808x608 pixel PointGrey cameras with a 6 mm fixed focal length. Calibration of the stereo camera’s intrinsic and extrinsic parameters is done using a checkerboard and

OpenCV’s standard function[12]. We use tags to calibrate the pose of the left camera on the end effector in a similar way as presented in [16-B.3]. Finally, we use the current estimate of the robot base pose to express the detected welding node in the global reference frame.

A. Image Processing

Upon receiving a stereo image pair, distortion is removed from the images and they are rectified to align their epipolar lines. The images are blurred with a normalized 3x3 box filter and then the Canny-Edge detector[14] and probabilistic Hough Transformation[15] are used to detect the wires edges of the spatial mesh as lines in the images. We then classify the extracted lines as either continuous or discrete wire candidates based on their angle in the image plane (Fig. 6).

B. Wire 3D Calculation

The output from the previous step is a set of continuous and discrete line candidates in the left and right stereo images. In order to find the 3D position of the desired welding node, we must identify which line candidates correspond to the desired continuous and discrete wires. This is achieved by finding the pair of continuous and discrete line candidates in the left and right images which minimize a measure of the quality of the match. Through brute force search, all sets of discrete and continuous wire candidates in image are sequentially evaluated. For a stereo line pair \( l_l \) and \( l_r \) we use the disparity of their start and end points to project the line into 3D, written as \( \tilde{W}_c \) for a continuous and \( \tilde{W}_d \) for a discrete wire candidate. We use the following cost function, which takes a continuous and discrete wire candidate \( \tilde{W}_c \), \( \tilde{W}_d \) as input and compares it to the position of the target welding node of our CAD model \((\tilde{W}_c, \tilde{W}_d)\):

\[
\begin{align*}
\text{cost}(\tilde{W}_c, \tilde{W}_d, \tilde{W}_c, \tilde{W}_d) &= \text{dist}(\tilde{W}_c, \tilde{W}_c) \\
&+ a \cdot \text{angle}(\tilde{W}_c, \tilde{W}_d) + b \cdot \text{dist}(\tilde{W}_d, \tilde{W}_d) \\
&+ c \cdot \text{angle}(\tilde{W}_d, \tilde{W}_d) + d \cdot \text{weld}(\tilde{W}_c, \tilde{W}_d)
\end{align*}
\]

\(a - d\) are weighting constants for the cost metrics. \(\text{dist}\) computes the shortest distance between two line segments in 3D. \(\text{angle}\) computes the angle between the two lines. Additionally, since we know that the continuous and discrete wires are welded together, the last term evaluates how close the continuous and discrete wires are to each other. \(\text{weld}\) therefore computes this distance similar to \(\text{dist}\) but considers the input line segments as infinite lines, since the extracted line segments might not extend to reach the welding point.

The two line pairs that, when projected to 3D, result in the lowest value of \(\text{cost}\) are used to define the target welding node. Since the extracted lines are found at the edge of the target wires, we move the best-match line pairs to the middle of the wire, as seen in Fig. 6. This is done by a fixed-distance parallel move towards the region that is darker. We do that by summing up the pixel intensity values on both sides of
the line in 2D. We move the line if one sum is a factor b bigger than the other sum. The final step of the wire detection algorithm is to unify the position and length of the detected continuous and discrete wire since the detected lines typically do not extend all of the way to their intersection point. The detected welding node can now be expressed in the global reference frame by using the current robot base pose. It can then be compared to the CAD model to adapt the building plan accordingly as described in Section VI.

C. Algorithm Performance

The wire detection algorithm is only locally accurate, as it finds the closest welding node to the expected location. If the true position of the target welding node is off by more than one half of the mesh grid size, wire detection will most likely find the wrong welding node. In this way the failure rate of the wire detection algorithm relies on the accuracy of the robot pose estimation and the accurate fabrication of the mesh. Over the entire mesh building process, we observed a total wire detection failure rate of 2.2% (123 wrong matches of the total 5594 scanned welding nodes). Wire detection failure was automatically detected by geometrically comparing the distances to previously measured welding nodes and the operator was notified of the failure.

The computation time for finding one welding node is dependent on the number of lines detected in the stereo images. We set the parameters of the line detector such that an average of 15 discrete and continuous lines are found per image. Using these parameters, computation on the IF’s main on-board computer takes around 0.1 seconds per welding node (Intel i3-3220T 2.8GHz running on a single thread).

VI. Adaptive Building Plan

This section describes the incremental adjustment of the building plan to compensate for observed deflections in the mesh during construction. The output of the wire detection algorithm described in the previous section is the 3D position of a welding node, expressed as two line segments representing the discrete and continuous wire. Estimating the 3D position of all welding nodes of a fully built layer of the mesh is referred to as ‘scanning’ that layer. This scanning process can not take place while fabricating a layer, because the end effector is attached to the mesh and tends to pull the wire away from its relaxed position. Scanning is therefore performed after the fabrication of a layer is complete. After scanning a layer, the mesh’s deformation is computed by calculating the difference between the expected (from the CAD model) and estimated (from wire detection) location of each welding node. To correct for measured deflections, the angle of insertion for the discrete wire elements of subsequent layers is adjusted. This essentially adjusts the path of the mesh in the horizontal direction to bring it back to the desired shape. The rotational correction angle $\alpha$ is calculated for each welding node as follows (see Fig. 7 for illustration):

$$\alpha = \tan^{-1}\left(\frac{d}{n f(c, h)}\right)$$

where $d$ is the distance between the expected and estimated continuous wire from the previous layer, $n$ is the number of layers over which the correction should ideally bring the mesh back to the desired shape and $l$ is the distance between two consecutive layers (this varies from 28 to 42 mm in the constructed wall). $f(c, h)$ (which is called the correction factor) is a weighting that adjusts the aggressiveness of the building plan adaptation based on the curvature of the steel wire ($c$) and the height of the mesh ($h$). Through many tests in our lab and while fabricating on the construction site it was observed that these two parameters have the greatest impact on how much the mesh deflects. In future work, it would be interesting to try to either model the mesh deflection behavior and improve the adaptation strategy accordingly, or use machine learning techniques to improve the building plan adaptation over subsequent tests. For this work, we simply used a rough discrete mapping tuned by hand.

In the interest of construction speed, scanning is performed only after every fourth layer built, though a better result could be achieved by scanning and correcting after every layer. Additionally, only one side of the double-leaf mesh is scanned and it is assumed that both sides deflect by the same amount. The time needed to scan one layer consisting of 34 welding nodes takes about 2.5 minutes, coming mostly from the time it takes to move the industrial arm between welding nodes. During tests in our lab before going to the construction site, this adaptive strategy was observed to greatly improve the overall fabrication accuracy. Additionally, it improves the robustness of the construction plan accordingly as described in Section VI.
process, as the mesh remains closer to its expected position making unexpected robot collisions less likely.

VII. RESULTS

The doubly-curved steel wire mesh was successfully constructed over roughly 125 production hours on the construction site. The resulting mesh is 2.8m high, 12 m long, has a varying thickness averaging 8cm, and is composed of 335 layers with more than 22000 welding nodes. To validate the fabrication accuracy, we measured the finished steel mesh with a Faro Focus3D X 330 laser scanner, generating a point cloud with ±2mm precision and 0.4mm standard deviation, and compared it to the original mesh design. After box filtering, denoising and down sampling the point cloud, we fit our CAD model to the cloud, such that the squared distance between the edge line elements of the CAD representation of the steel mesh and each point in the cloud was minimized. For every welding node, we calculated the average distance from each welding node to the 10 nearest points in the point cloud. Results are shown in Fig. 8.

Two error sources can be observed. First, errors in the global robot pose estimation create visible jumps in the error between the layers in which the robot was repositioned. Second, it can be seen that the errors in the mesh position are highest towards the top of the mesh and in regions of high curvature, indicating that mesh deflections after fabrication are still present despite the building plan adaption. While we observed during tests in our lab that these deflections would have been significantly worse without the building plan adaption, this indicates that there is still room for improvement in the adaption strategy. Overall, the average observed error is 9.47mm, with a maximum error of 38.63mm.

VIII. CONCLUSION

In this paper we have presented a sensing system to enable the in situ fabrication of an undulated steel wire mesh on a real construction site which, through the Mesh Mould process, will become a load-bearing reinforced concrete wall. Through this construction, we have shown that it is possible to accurately fabricate a construction-scale geometrically differentiated structure using off-the-shelf cameras and process-specific software. We demonstrated the performance of the localization system with ground truth measurements and the accuracy of the constructed mesh with a high-resolution laser scan.

As the future work on IF and in situ robotic building construction, we plan to move to more dynamic sensing and control solutions. While we have shown that a mobile robot can build with sufficient accuracy at the building scale, the speed of construction is severely limited by the fact that all building and sensing steps are executed in a quasi-static manner. To enable the safe execution of more dynamic building tasks, we are investigating lighter and more compact machines and more sophisticated control methods to achieve the same accuracy as the current heavy and stiff systems. Such systems will also require dynamic sensing solutions that can sense the end effector pose in real-time such that the system can react to the imperfect behavior that will arise as robot motions get faster. These systems will then have the potential to expand the possibilities for adaptive fabrication by supporting the real-time adaptation of continuous building processes, for example the robotic finishing of concrete surfaces.

ACKNOWLEDGMENT

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research (NCCR) Digital Fabrication (Agreement #51NF40_141853) and a Professorship Award to Jonas Buchli (Agreement #PP00P2_138920). The authors would like to thank Alexander Nikolaus Walzer for his continuous support on hardware modifications and experiments on the robot, as well as operating the robotic system during the mesh fabrication on site. Furthermore, the authors would like to thank Michael Neunert for his implementation of the TCP/IP interface.
REFERENCES