Online Synchronization of Building Model for On-Site Mobile Robotic Construction

S. Ercan Jenny[a], H. Blum[b], A. Gawel[b], R. Siegwart[b], F. Gramazio[a] and M. Kohler[a]

[a]Department of Architecture, ETH Zurich, Switzerland
[b]Department of Mechanical and Process Engineering, ETH Zurich, Switzerland

ercan@arch.ethz.ch, hermann.blum@mavt.ethz.ch, abel.gawel@mavt.ethz.ch, rsiegwart@ethz.ch, gramazio@arch.ethz.ch, kohler@arch.ethz.ch

Abstract -
This research presents a novel method for a data flow that synchronizes building information with the robot map and updates building components to their "as-built" states, in order to facilitate an on-site mobile construction process. Our experiments showcase mobile mapping and localization of a robotic platform featuring segmentation, plane association and quantitative evaluation of deviations. For the users of the on-site mobile robotic system, we present a suitable interface that allows for task level commanding and the selection of target and reference building components (i.e. walls, floor, ceiling). Additionally, this interface seamlessly integrates the online workflow between building construction and the robot map, updating the target building components to their "as-built" states in real time and providing a visual representation of additional task-specific attributes for building components in the robot map, in addition to geometries. This is presented as a first step toward integrating users of the system into the proposed robotic workflow to develop decision-making strategies for fitting building tasks to local references on-site.

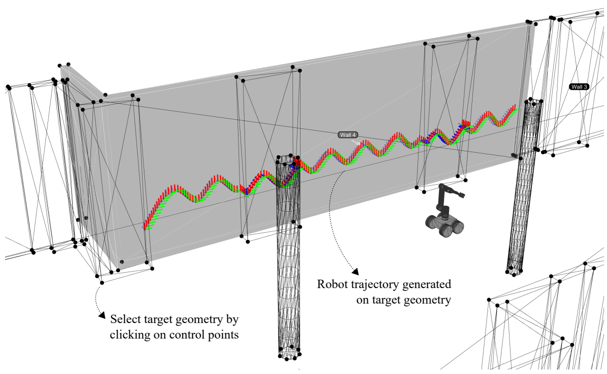

Figure 1. Robot trajectory representing the building task, generated on selected target geometry.

Keywords -
On-Site Mobile Construction; Localization; Construction Robotics; Building Model; As-Built; Deviation Analysis; Robotic Construction Workflow with User Interaction

1 Introduction

Robotic technologies are widely applied in the off-site prefabrication of building components, where the strength of high-tech assembly lines can yield their full potentials: robots and work-pieces are in fixed locations with constant conditions, and building components can be mass-produced without the need to dynamically adapt the process. However, the majority of building tasks are executed directly on construction sites. In contrast to off-site prefabrication, on-site building construction often deals with dynamically changing conditions of large scale building components in spatially complex and cluttered environments. Furthermore, on-site work inherently generates deviations from the "as-planned", which is the state of the design as it should be. This requires craftsmen to register the differences and adapt the building tasks according to the "as-built" condition, which is the state of the construction as it is [1].

To facilitate an on-site construction cycle in an unbroken digital chain, a mobile construction robot must be able to understand the context within which it is working: it must localize itself via a robot map, both globally and locally, in reference to the already built components and the task being executed. In addition, it must be able to detect and understand any divergences between "as-planned", and "as-built" conditions. To achieve this in an on-site robotic construction process, a key challenge is linking the building information to the mobile robot's internal representation of the world (referred to as a robot map) and perception of its surroundings, using on-board sensing. This information is often represented as point clouds where the underlying geometric relationships are unknown. Such linkages between the building information and the mobile robot allow to cope with any divergences and the associated inaccuracies of the building materials and components, and to apply decision-making strategies while executing building tasks. To facilitate this link, there must be a flow of data between the complex building information and the construction robot.

In this paper we present:

- An online digital workflow between the building model and robot perception

- A suitable interface that allows the users of the on-site mobile robotic system to fit building tasks to local references on-site
- Real-time adaptation of the building components (i.e. walls) with respect to measured "as-built" state

2 State of the Art

Recent developments indicate an increased use of mobile robots on jobsites to monitor, track progress[1], or register differences[2]. Still, the problem of how to manage the flow of data between mobile robots that build and the complex building information is an emerging topic. Early automation attempts in the 1990s sought to replace manual processes with robotic building technologies in the construction sector, and resulted in early-stage mobile construction robots such as [2, 3, 4, 5], all of which lacked the hardware ability to interface to the complex building information. An early attempt to have a mobile construction robot with a limited interface using integrated sensing abilities, for localizing itself with respect to the building components for executing building tasks on-site is exemplified in [6], but still not enabling real-time adaptation of building components.

A large amount of research is dedicated to automated modelling of "as-built" states of building components [7, 8], and localizing camera and laser sensors in a Building Information Model (BIM) as well as tracking construction progress [9, 10, 11, 12, 13, 8]. While these works show an integration of BIM with static and mobile sensing, the assumption is that the "as-built" status reflects the "as-planned" status and the problem of progress tracking is then defined as detecting the presence or absence of discrete building components. In this paper, we address the challenge of having an online data flow for an on-site mobile construction process by detecting and communicating metric deviations, such as walls being placed several cm differently than "as-planned". Detection of such metric deviations enable the adaptation of building models beyond mere presence or absence of building components.

Current research in the field, such as [14, 15], has put forward novel approaches for software interfaces that facilitate the adaption of "as-planned" building information in an on-site mobile construction process. In the case of [14], for the task of automated brick-laying, the planned geometries of two pillars are fitted to the robot map, and their as-built location is extracted to generate the robot tasks for fabricating a brick wall between them in a mobile robotic construction process. In the case of [15], a similar approach is implemented locally. By tracking visual features of the geometry that is being built, each discrete steel member is registered with stereo vision and locally corrected, all facilitated by the interface between the robot perception and the building information during the execution of the building task. These works demonstrated the feasibility and necessity of deviation measurements to adapt and execute construction tasks online, but they were limited to specific pre-programmed geometries and tasks.

It has not been demonstrated so far to establish an online data flow for an integrated robotic construction process and a flexible work area selection directly on job sites, localizing the construction robot both globally and locally in reference to the already built components (local references such as walls, floor, ceiling). With this research, we propose the online deployment of the building model into the robotic construction process, aiming to introduce a real-time method to plan a robotic workflow by fitting building tasks to local, selected references on-site, via a suitable interface.

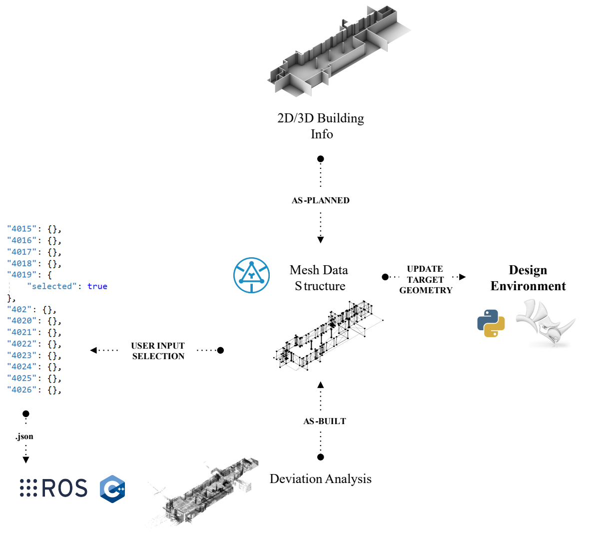

Figure 2. Overall workflow.

3 Method

Towards a generalized workflow, our method considers an abstraction where a building task is executed (i.e. as a robot trajectory) on a selected target geometry. To execute such a task, the robot requires poses of the trajectory in a local coordinate frame, which we anchor at given reference geometries. In this section, we describe a method that can execute such a task although the location and orientation of the target mesh faces (belonging to a target geometry), with respect to the reference mesh faces (belonging to reference geometries), may diverge from the "as-planned" state.

The procedure for the proposed method consists of the

[1] https://www.doxel.ai/
[2] https://www.scaledrobotics.com/

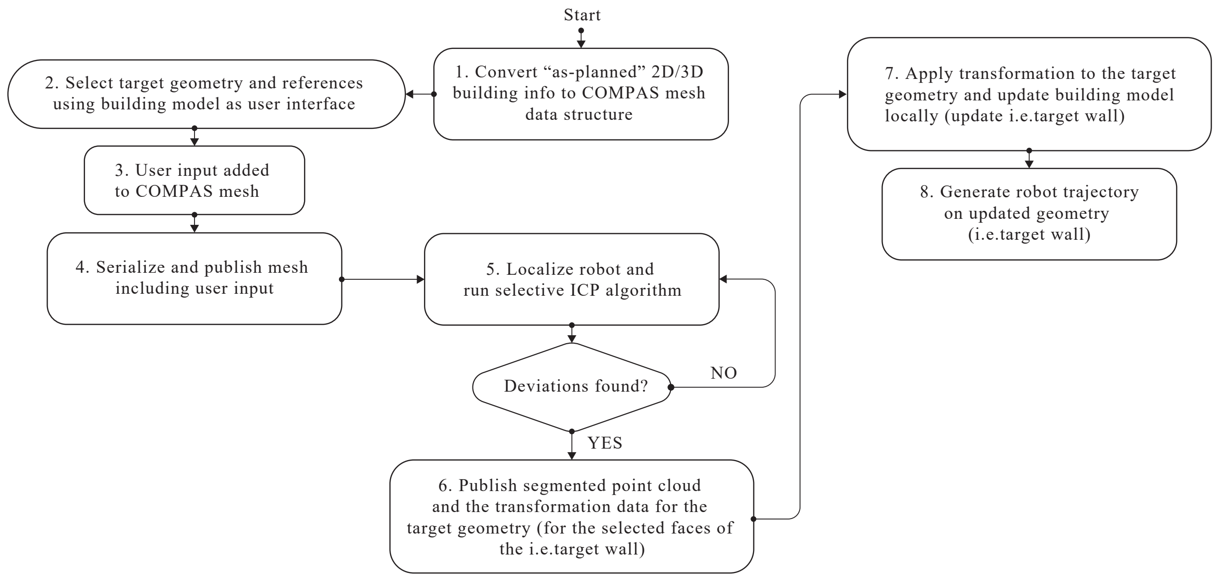

Figure 3. The procedure for the proposed method.

following steps (Figure 3): Firstly, the mesh data structure representing the building components is generated from the “as-planned” 2D/3D building information, as described in subsection 3.1. Secondly, the user selects the target geometry and the references by clicking on the control points for bounding the areas of interest, and the program executes necessary steps to find the target and reference mesh faces that contain all given points as vertices. This is then included in the mesh data structure via labeling the target mesh faces as "selected" and the reference mesh faces as "reference", as described in subsection 3.2. Following this, the mesh data structure containing the user input is serialized and published as a message in the ROS environment running the robot controller. Next, the robot is localized and the ICP algorithm is executed on-site, as described in subsection 3.3, using local references (coined selective ICP) [16]. The calculated transformation is published back and imported into the building model. As described in subsection 3.4, the transformation is applied to the target geometry and the building model is updated locally in the design environment. Finally, the robot trajectory representing the building task is generated on the "as-built" target wall. These steps are also shown in an accompanying video[3].

3.1 Building Model Data Structure

A schematic overview of the data representation communicated to the robot is shown in Figure 4. This

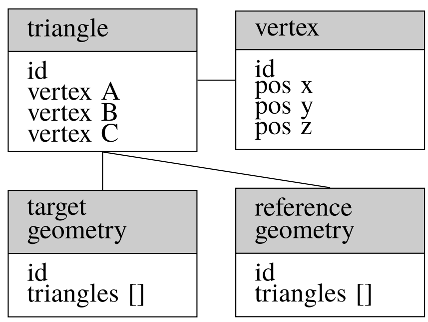

Figure 4. Data representation communicated between the building model and the robot.

data representation is initially generated from the "as-planned" 2D/3D building information, using the open-source, Python-based computational framework COMPAS[4] and visualised in 3D modeling software Rhino (Figure 2). It contains the mesh data structure of the COMPAS framework that allows for storing and adapting geometry and topology, aiming to represent the continuously changing building information and the robot trajectory (representing the building task) in relation to *target geometries*, i.e selected faces of the target wall. Like this, robot trajectory planning and generation from the design environment is aligned with the "as-built" state and the robot map (Figure 1). The communication between the robot controller - running in a ROS[5] environment - and the Python-based

[3] https://youtu.be/pu4hb_nZNUw

[4] https://github.com/compas-dev/compas

[5] https://www.ros.org/

design environment is established using the ROS Bridge library roslibpy[6]. Additionally, the kinematic model of the mobile robot is visualized in the design environment using the COMPAS FAB package of the COMPAS framework. This allows users to visualize the current robot state and task status in relation to the building model in the later stages of the process as well (such as i.e. feedback- based plaster spraying - that comes after Step 8 of the procedure shown in Figure 3) - which is not included within the scope of this paper.

3.2 Building Model as User Interface

The building model described in 3.1 - with the necessary abstraction level - is used directly on-site for task level commanding and for the selection of the target geometries and the necessary references, which are the relevant faces of the neighboring meshes, constraining the work area for the building task in the x, y and z axes. Firstly, the user selects the target geometry by clicking on the control points (located on the corners of the geometries) for bounding the area of interest (Figure 1), and the program executes necessary steps to find the target mesh faces that contain all given points as vertices. The same steps are repeated for the selection of the references. Next, the selected vertices are labeled in the mesh data structure, which is then serialised and communicated to the robot.

3.3 Robot Localization

The robot localizes against the building model using measurements from the 3D LiDAR described in subsection 4.1. In this way, the user provides an initial coarse alignment during robot start-up. The robot subsequently aligns the LiDAR scan with a sampled point cloud from the mesh as described in [17], using the Iterative Closest Point (ICP) algorithm [18]. To increase accuracy and overcome ambiguities, the alignment is constrained to a few *reference mesh faces* [16]. Here, these reference mesh faces are selected through the interface described in subsection 3.2 and communicated to the robot as described in subsection 3.1. Consequently, all LiDAR scans are aligned to the selected reference frame when measurements on the *target mesh faces* are extracted.

3.4 Adapting the Building Model to As-Built

When the robot is localized, deviations between "as-planned" and "as-built" can be measured on the target geometry. Candidate planes from the LiDAR scan are extracted in order to find the plane associated to the target geometry. Given measured distances d_{it} between points i on the candidate plane c and the target plane t, we associate

[6]https://github.com/gramaziokohler/roslibpy

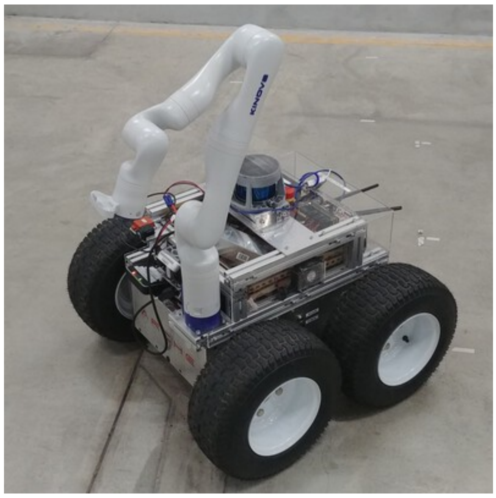

Figure 5. Mobile robotic platform used in our experiments, here with an additional arm mounted.

planes by:

$$\arg\min_c \ \lambda_1 \min_i d_{it} + \lambda_2 \frac{1}{N} \sum_i d_{it} + (1 - \lambda_1 - \lambda_2)\,\mathrm{rot}(c, t)$$

where λ_1, λ_2 are tuning weights. The geometric transformation between "as-planned" and "as-built" can then be found as the translation between the plane centroids and rotation $\mathrm{rot}(c, t)$ between the face normals of the target geometry. As the current sensor setup does not allow the measuring of points over the whole height of the building components (i.e. walls), the z coordinate of the centroid cannot be estimated and therefore the z translation is not considered in the experiments. Upon detection of the geometric transformation, it is applied to the target geometry and the building model is updated locally.

4 Experiments

Experiments are presented in three subsections. The first subsection describes the robotic platform; the second subsection showcases the online selection of references and the work area and the third section focuses on the deviation detection and update of the target geometries locally in the building model. All experiments are conducted at a parking garage, which is set up as a mock construction scene at ETH Zurich.

4.1 Robotic Platform

The experiments are conducted on our open-source robotic platform[7] that consists of a wheeled base with LiDAR and IMU sensors, as shown in Figure 5. The platform's capabilities of high-accuracy manipulation [17] and

[7]https://github.com/ethz-asl/eth_supermegabot

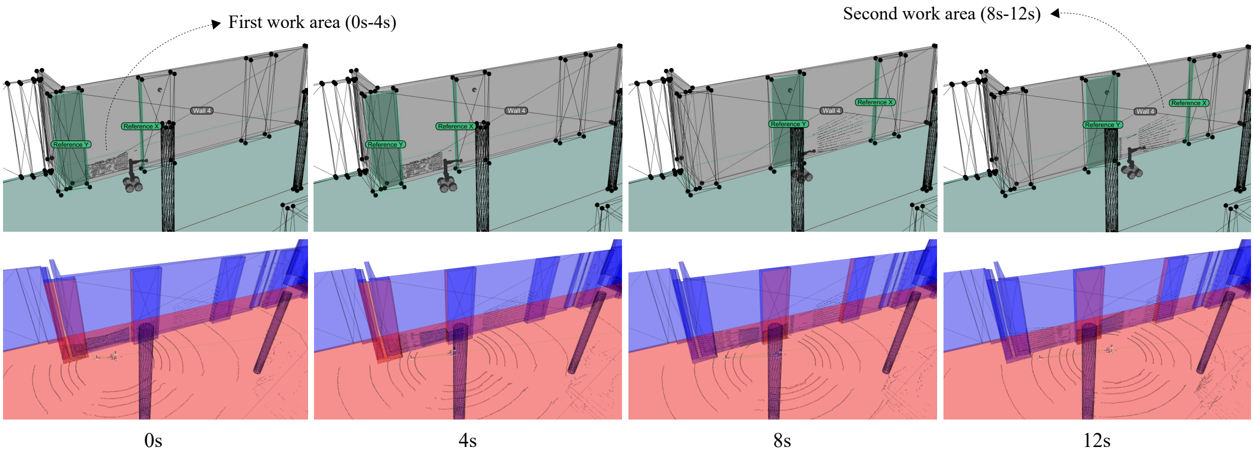

Figure 6. Top row: Online selection of references and work area. Bottom row: Robot map and perception.

localization in cluttered environments [16] were validated in earlier works. Since no manipulation for the execution of the building task is tested within the scope of this paper, the robotic arm was not mounted on the mobile platform in our experiments.

The LiDAR sensor used has an accuracy of 2 cm, 16 beams, and an opening angle of 30°, leading to height measurements on target walls of approximately 1 m. The scope of this work is therefore not a precise deviation analysis or sensing benchmark, which is out of the capabilities of the used sensors, but the demonstration of an integrated online workflow. The demonstrated workflow is not specific to the sensors used here, and can in fact be applied to any robotic platform that has the capability to localize in a robot map and to measure spatial information on selected geometries.

4.2 Online Selection of References and Work Area

In the first set of experiments, steps 1-4 of the procedure shown in Figure 3 are tested in order to showcase the initial steps of the online data flow proposed for synchronizing building information and the robot map. Initially, for the first work area shown in Figure 6, relevant references X, Y, and Z are selected, where X refers to the reference constraining the work area in the x-axis, and Y and Z, refer to the ones constraining the work area in the y and z axes. Without the need to reload a new building model to the robot off-site, the references are shifted on-site (as described in subsection 3.2) to the ones shown in Figure 6, at 8s. This successfully demonstrates a flexible, online method for work area selection to execute a building task (i.e. plaster spraying), for which fitting to relevant and local references on-site is crucial.

4.3 Deviation Detection and Update

In the second set of experiments, steps 1-8 of the procedure shown in Figure 3 are tested in order to showcase the update of building components to their as-built states for tolerance handling to facilitate an on-site construction process. For a selected set of target geometries shown in Figure 7, Walls 1-3, the same set of references X, Y, and Z (shown in Figure 7) are selected as described in subsection 3.2, constraining the work areas in x, y, and z axes to execute the selective ICP algorithm on-site. Robot trajectories are then generated on updated target geometries.

Table 1. Geometric transformations calculated for the target geometries, resulting from deviation detection: Translation of the mesh face centroid and rotation of the mesh face normal

Target	X Translation	Y Translation	Rotation
Wall 1	−30 ± 11 mm	−23 ± 34 mm	1.1 ± 1.4°
Wall 2	−13 ± 1 mm	−109 ± 1 mm	2.0 ± 1.4°
Wall 3	−5 ± 50 mm	36 ± 42 mm	0.9 ± 0.1°

5 Results

In all conducted experiments, the robot correctly associated its measurements to the selected target geometry and was able to provide associated sensor readings in the form of a point cloud, in addition to a parametric analysis of the deviation. This is important, as it allowed for updating the target geometry and for adapting the robot trajectory, representing the building task. The geometric transformations calculated for each target geometry are presented in Table 1.

Within the scope of this paper, the experiments focused

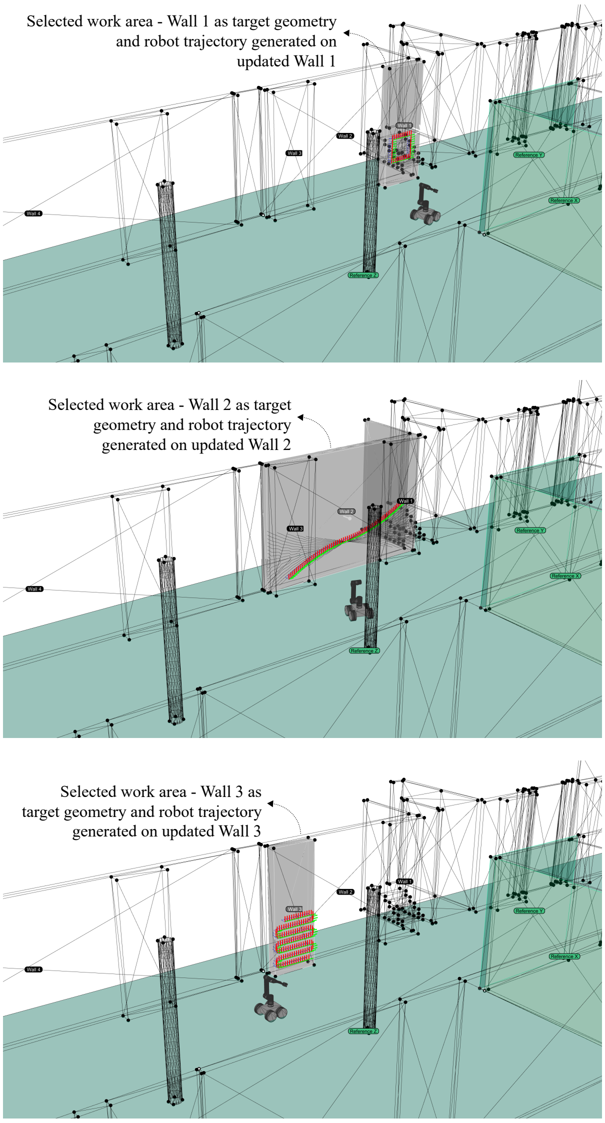

Figure 7. Deviation detection on selected target geometries, Walls 1-3, and generation of robot trajectories on updated target geometries.

on demonstrating the online workflow of the robotic construction system. In order to assess achievable degrees of precision, tests will be performed with higher-quality sensors on the robotic platform, and results will be compared to a ground truth site survey derived from measurements with a tripod system.

6 Conclusion and Outlook

In this paper, we present the first implementation steps of a novel method for an online data flow to synchronize building information with the robot map for facilitating an on-site construction process in an unbroken digital chain. This is established via linking the generation of robot trajectories, representing the building tasks, to the "as-built" states of selected building components directly on-site. Within the experiments presented in this paper, this is achieved by updating the target geometries in the building model, via the selective ICP algorithm executed directly on-site. Online deployment of the building model as an interface for including human actors into the robotic construction process is also tested, introducing a flexible method to plan a robotic workflow by fitting building tasks to local references on-site. However, these experiments do not extend to the actual execution of the tasks, which involve i.e. feedback-based plaster spraying, grinding, chiseling, etc. For these steps, in-process robot trajectory adaptation will be established with continuous process control, using visual feedback for acquisition of the current state of target geometries i.e. spraying, chiseling, and grinding surfaces. The overarching goal of this research is the development of a digital toolbox, so that the compatibility of the proposed workflow can be tested on different mobile robotic platforms deployed for on-site construction.

Within the scope of the experiments presented in this paper, the proposed method facilitated the digitization of crucial task information, i.e. selection of relevant references for building tasks for importation back into the building model and robot map. In further development, we will explore methods of dispatching task-specific instructions via different types of interfaces and experiment with on-site robotic workflows based on human collaboration and aim to further leverage the strengths of both humans and robots, enhancing the capabilities of digital construction processes.

Acknowledgements

We would like to thank Julian Stiefel for his contributions on mobile robotic deviation analysis. This work was partially supported by the Swiss National Science Foundation (SNF), within the National Centre of Competence in Research Digital Fabrication (NCCCR DFAB) and by the HILTI group.

References

[1] Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction*, 19(7):829–843, 2010.

[2] Jurgen Andres, Thomas Bock, Friedrich Gebhart, and Werner Steck. First results of the development

of the masonry robot system rocco: A fault tolerant assembly tool. In Denis A. Chamberlain, editor, *Automation and Robotics in Construction XI: Proceedings of the Eleventh International Symposium on Automation and Robotics in Construction (ISARC)*, pages 87–93, Brighton, UK, May 1994. International Association for Automation and Robotics in Construction (IAARC). ISBN 9780444820440. doi:10.1016/B978-0-444-82044-0.50016-3.

[3] D. Apostolopoulos, H. Schempf, and J. West. Mobile robot for automatic installation of floor tiles. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3652–3657 vol.4, 1996.

[4] Ronie Navon. Process and quality control with a video camera, for a floor-tilling robot. *Automation in Construction*, 10:113–125, 11 2000. doi:10.1016/S0926-5805(99)00044-8.

[5] G. Pritschow, M. Dalacker, J. Kurz, and M. Gaenssle. Technological aspects in the development of a mobile bricklaying robot. In Eugeniusz Budny and Anna McCrea, editors, *Proceedings of the 12th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 281–290, Warsaw, Poland, June 1995. International Association for Automation and Robotics in Construction (IAARC). ISBN 9788386040025. doi:10.22260/ISARC1995/0034.

[6] V. Helm, S. Ercan, F. Gramazio, and M. Kohler. Mobile robotic fabrication on construction sites: Dimrob. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4335–4341, 2012.

[7] Tomás Werner and Andrew Zisserman. New techniques for automated architectural reconstruction from photographs. In *European conference on computer vision*, pages 541–555. Springer, 2002.

[8] Khashayar Asadi, Hariharan Ramshankar, Mojtaba Noghabaei, and Kevin Han. Real-time image localization and registration with bim using perspective alignment for indoor monitoring of construction. *Journal of Computing in Civil Engineering*, 33(5): 04019031, 2019.

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

[10] Viorica Pătrăucean, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, and Carl Haas. State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2):162–171, 2015.

[11] Christopher Kropp, Christian Koch, and Markus König. Interior construction state recognition with 4d bim registered image sequences. *Automation in construction*, 86:11–32, 2018.

[12] YM Ibrahim, Tim C Lukins, X Zhang, Emanuele Trucco, and AP Kaka. Towards automated progress assessment of workpackage components in construction projects using computer vision. *Advanced Engineering Informatics*, 23(1):93–103, 2009.

[13] Kevin Han, Joseph Degol, and Mani Golparvar-Fard. Geometry-and appearance-based reasoning of construction progress monitoring. *Journal of Construction Engineering and Management*, 144(2): 04017110, 2018.

[14] Kathrin Dörfler, Timothy Sandy, Markus Giftthaler, Fabio Gramazio, Matthias Kohler, and Jonas Buchli. Mobile robotic brickwork. In *Robotic Fabrication in Architecture, Art and Design 2016*, pages 204–217. Springer, 2016.

[15] Manuel Lussi, Timothy Sandy, Kathrin Doerfler, Norman Hack, Fabio Gramazio, Matthias Kohler, and Jonas Buchli. Accurate and adaptive in situ fabrication of an undulated wall using an on-board visual sensing system. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[16] Hermann Blum, Julian Stiefel, Cesar Cadena, Roland Siegwart, and Abel Gawel. Precise robot localization in architectural 3D plans. June 2020.

[17] Abel Gawel, Hermann Blum, Johannes Pankert, Koen Krämer, Luca Bartolomei, Selen Ercan, Farbod Farshidian, Margarita Chli, Fabio Gramazio, Roland Siegwart, et al. A fully-integrated sensing and control system for high-accuracy mobile robotic building construction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019)*, 2019.

[18] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.