Self-Interlocking Structures, Multi-View Marker-based Pose Estimation and Impedance Control - Towards Automated Sequential Assembly

Paul Hallmann *[1] , Nicolas Nonnengießer *[1]

Abstract— **In this work, we present a new approach to automate the process of construction using self-interlocking building blocks specifically SL-Blocks. Utilizing this block type, complex structures will be automatically assembled with an industrial robot arm. The challenges inherent to the task, the need for high precision tracking, and occlusion robustness for said tracking are needed to accomplish the joining of parts with tolerances at the millimeter level. They are tackled by building on recent methods for fiducial marker-based tracking. The compliant design of the blocks combined with impedance control allows us to perform robustly despite remaining uncertainties. Our proof of concept setup is nearing completion. It is described alongside our vision for the next project stages in future work.**

I. INTRODUCTION

With the ongoing research in the field of robotics and the advancements made in the last years, new opportunities for useful applications arise. Robots are already widely used in industrial settings to perform simple repetitive tasks previously done by human workers [1]. Especially for factory assembly lines, robots have been shown to effectively automate tasks without the presence of humans in a well-defined environment. However, automation in the domain of construction work is still largely unexplored. Despite many innovative building projects which not only grow in their complexity but also in size, the core of construction work is still dominated by manual labor. This is largely due to the many complex work steps in current construction procedures, which makes it very difficult to automate those tasks using robots. As shown by Gharbia et al. [2], more research is now being conducted in the field of construction with more papers being released every year. Particularly the area of additive manufacturing as well as automated installation and assembly are popular topics. One way to do so is to change the way structures are built. This has been shown successfully by using giant 3D printers to extrude structures out of special concrete mixtures [3]. Even though this approach is still subject to research, it has been shown to be a promising alternative for constructing arbitrary structures in the future.

Besides the automation context, it is important to look at the sustainability and environmental impact of current construction methods. Since many types of contemporary constructions are based on the permanent bonding of building parts with mortar or adhesive material, it is not always possible to dismantle such structures without destroying the individual parts. This results in a lot of wasted materials. Working with reusable components would not only enable faster assembly and disassembly of such structures but also bring financial and environmental benefits. Previously used parts of structures can be reused and therefore do not need to be produced from scratch. This can be achieved using building elements that are held together by topological interlocking [4].

This work aims to develop an automated system for the construction of predefined 3D structures out of SL-Blocks [5], a self-interlocking block system. There has been some previous work on developing methods to stack SL-Blocks and other non-specifically shaped blocks in the architectural context [6] [7] [8]. However, the focus was mainly on developing motion-planning strategies for placing the blocks.

As joining multiple blocks into a structure requires millimeter level accuracy we pursue high-precision block tracking. An inherent challenge to the task is that the manipulated objects are subject to high degrees of occlusion especially when tracked from only a single perspective. We address this by applying fiducial markers to each face of the block, for which we assume the transformation to be known. Taking this as an additional assumption we modify recent multi-marker and multi-view-based object tracking methods. This enables us to infer the translation and orientation of occluded block surfaces from the detected visible markers. The known block model including marker placements is also expected to increase the tracking precision.

Based on the generative system to build self-interlocking structures introduced by [5] we develop a new approach for automated structure construction. Shih's work gives us a decomposition of the desired structure. Existing software generates a construction plan for the decomposition, meaning an order of block placements. Given block pickup poses another piece of existing software will then generate robot trajectories to pick and insert the blocks into the partial structure. Robot control will for now be done using the control software provided by Franka, the manufacturer of our robot, that follows input trajectories.

This is where our work picks up. As the first step towards automated assembly, the focus of this project lies on the development of marker-based tracking of SL-Blocks. Additionally to this the full integration of a construction pipeline using the existing parts is tackled. This includes developing the experimental setup and creating software bridges using the Robot Operation System (ROS)[9].

*All authors contributed equally
[1]TU Darmstadt

Lastly, we are using a simulation environment to evaluate our tracking performance in different scenarios. Furthermore the simulation will enable us to freely experiment without safety concerns, and possibly, later on, provide a platform for reinforcement learning (RL) to train controllers used for stacking blocks.

II. Related Work

A. SL-Block

Finding and designing new self-interlocking structures is an active field of research with possible applications in many fields. Engineers and architects are looking for different types of interlocking blocks that can be easily assembled and disassembled without using fasteners or any kind of adhesive materials like mortar or glue. Current research focuses on making use of the topological interlocking property of these building blocks with the goal of building complex structures. Regarding the recent development in automated digital fabrication technology, 3D printing technology is used more and more to fabricate complex objects. However, when it comes to printing large objects, the extrusion capabilities for single-piece objects are limited by the size of the printer's working volume. To overcome this issue, recent work like Song et al. [10] proposes to focus on printing 3D parts and making use of their interlocking property instead of using an adhesive material.

In 2016, Shih and Shen-Guan [5] introduced the SL-Block, a specific type of polycube, more precisely an octocube built up from an S-shaped and an L-shaped tetracube attached to each other. Figure 1 shows the structure of the SL-Block. They introduce a generative process (context-free string grammar) to provide a formalized language to describe possible structures that can be built using the interlocking SL-Blocks. It has been shown that it is possible to create various structures of different complexity just by combining identical SL-Blocks in different orientations [11]. Using this language, large and firm structures can be built in a top-down manner. Due to the interlocking property of the SL-Block, it is possible to build hierarchical structures without using any type of adhesive material such as mortise/tenon, glue, or nails.

B. Object Tracking

Tracking and detection of objects is an active area of research with many different approaches. Those approaches can be mainly categorized by the type of data and the resulting dimension of the data used to infer hypotheses about the object. The more dimensions the more information is available to form a sophisticated guess of the location and possible orientation of the inspected object. There are computer vision-based, as well as non-vision-based approaches. A non-vision-based approach was used by [12] to track the object pose just by evaluating the joint measurements of a robotic hand holding the object of interest. However, they realized that using just the joint measurements leads to significant offsets in the object pose estimation. Therefore they included a vision-based detection system to fuse it with the previously gained joint angles to form a good estimation of the object pose. This demonstrates that for precise predictions of manipulated object's poses, more than robot joint information is needed. A vision-based object tracking method is used by Pauwels et al. [13]. They use an RGB-D camera to extract depth information to update a 3D simulation of the scene. The simulation is then used to determine the pose estimate.

A simple yet robust alternative is to use fiducial markers on the objects to be captured. Because of their great detection rates even in bad lightning conditions, inbuilt pose estimation for the tags and error-resistant design fiducial markers such as AprilTag are popular methods for object tracking [14], [15], [16] or even Simultaneous Localization and Mapping [17] in controlled environments. Of the currently available flat rectangular tag variant designs, AprilTag seems to perform best [18] and is thus used for our project. Recently marker bundle-based object trackers have shown remarkable pose estimation accuracy. In Sarmadi et al. [19] a joint approach for camera calibration, estimation of the relative transformations of the markers, and reference perspective trajectory estimation of the markers were presented. They used a multi-camera setup with partially overlapping fields of view (FOVs), objects with applied markers bundles, and reprojection error minimization to achieve this. In [20] a similar technique is pursued. Instead of using multiple cameras and general multi-marker object tracking, they focus on tracking a single dodecahedronal manipulator attachment. Tags are placed on its surface ensuring that multiple are visible at the same time from the camera's FOV. They calibrate the cameras, then detect and optimize the transformations between the markers. During operation, the calibration and transformation estimates are used to track the pose of the chosen reference marker from a single camera. Both papers accomplish tracking markers even though they might not be visible at the time, by estimating other marker poses from the visible marker poses using their optimized pair-wise transformations. To refine a singular estimation of all desired marker poses, both works minimize a reprojection error - the mean squared error over the differences of estimated marker transformations to the detected marker transformations.

C. Compliant Control

In classical robotics, robots are commonly controlled using position controllers, which facilitate precise motion and predictable execution of tasks. Industrial robots deployed in manufacturing environments serve as an appropriate example of such robots. These robots typically operate in environments where all parameters are predetermined and frequently carry out repetitive tasks that necessitate precise manipulation. However, the environments in which they operate exhibit little to no variability, implying that the robot's objective can be accurately defined. In simpler terms, these robots do not need to account for environmental imperfections. For instance, in the context of industrial pick-and-place operations, if the object to be placed varies slightly

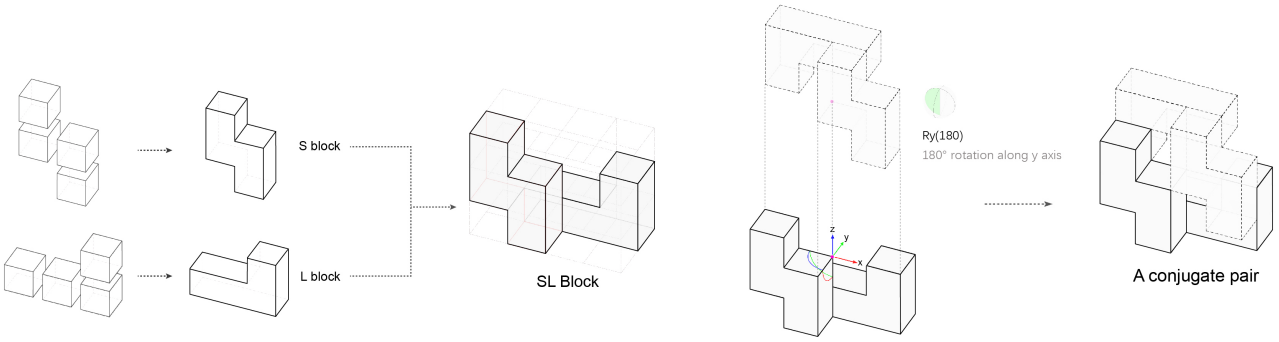

Fig. 1. Image of an SL-Block (Image by DDU). Introduced by [5] it consists of an S-shaped and an L-shaped tetracube attached to each other.

from the predetermined object location, the system may become unstable and fail. Thus, the conventional position-controlled robots may not be the best option for deployment in unstructured environments or for tasks that may entail variability. Compliant control techniques, such as impedance control, are an alternative to traditional position control for robots. These techniques enable robots to operate effectively in dynamic and uncertain environments by regulating their response to external forces and complying with the task requirements [21]. Impedance control, in particular, is a compliant control technique that has been widely used in robotics, allowing robots to respond appropriately to external forces or disturbances while interacting with the environment. This is commonly achieved by modeling the robot as a mass-spring-damper system. The spring component is used to regulate the resistance to displacement of the robotic arm and the damper limits the velocity.

$$F = M(\ddot{x}_d + B\dot{x}_d + K(x_d - x_a)) \tag{1}$$

F is the force applied by the robot's end-effector, M is the mass matrix of the robot, $\ddot{x_d}$ is the desired acceleration of the end-effector, $B\dot{x}_d$ is the desired damping term of the end-effector, K is the stiffness matrix of the end-effector, x_d is the desired position of the end-effector and x_a is the actual position of the end-effector. This equation represents the dynamics of the robot's end-effector in response to external forces and the desired impedance. The terms on the right-hand side of the equation represents the desired motion of the end-effector (the first two terms) and the desired impedance (the last term). By adjusting the values of M, B, and K, the characteristics of the robot can be tuned to match the required properties of the task.

III. Our Approach

In this section, we will describe the individual parts of our project pipeline. First, we describe the setup for the SL-Block. Followed by the real-world setup with three cameras and the Franka Emika robot [22]. Next, we describe the object pose estimation pipeline and our control scheme. Lastly, we describe how the Isaac-Sim Simulator is used to provide a simulation platform to evaluate our approach before applying it to the real robot.

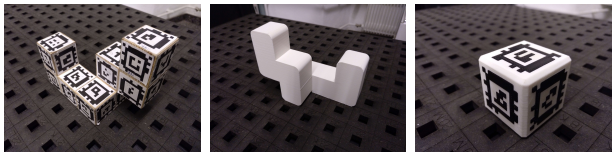

Fig. 2. Image of an SL-Block [5] with AprilTags (left). SL-Block with round corners (center). 3D printed april-tag prototype (right)

A. SL-Block

As described in section II-A this work makes use of the SL-Block, introduced by Shih et al. [5]. The SL-Block can be used due to its special topological interlocking property to build complex and firm structures by stacking them together without using any kind of adhesive material. To be able to detect and track the block, we use fiducial markers in the form of AprilTags [23]. A unique AprilTag is applied to each face of the block as can be seen in Figure 2. To get an exact mapping between the tag and position relative to the block, each placed tag is uniquely labeled by a number between 0 and 33. For each block, we use 34 different tags. To later distinguish different SL-Blocks, each marker id is only used once throughout the whole setup.

Although using many tags to label a block can be inconvenient practically and aesthetically, reducing the number of tags can lead to less accurate estimation since the pose is determined solely based on the visual-markers. When the majority of markers are obscured by the structure, having enough tags is essential for a robust estimate. Using only one camera, an average of less than 9 tags are used to determine the pose. We have considered limiting this number to use only the n best detections for future optimization. The detection pipeline itself is set up to scale with as many blocks as possible limited by the number of program instances your computer(s) can handle To address these issues, a promising solution is to use invisible tags. Recent research has explored the use of materials visible only in the infrared spectrum to create tags that are invisible to human vision

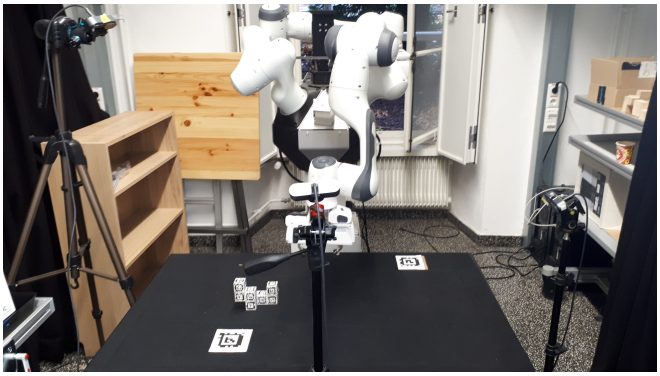

Fig. 3. Experiment setup containing the dual-arm robot and the three cameras positioned around the table.

[24]. By adopting this technique, structures can be labeled in a visually pleasing way while still maintaining accurate labeling. This approach not only enhances the structure's aesthetics but also its functionality by reducing visual clutter and improving the user experience.

We have multiple variations of SL-Blocks to work with. The oldest and still primarily used block in terms of experimentation is made of wood with tags attached to each face. This block however has multiple drawbacks which we will go into here. The first drawback is the fact, that the tags are glued to the faces of the block which results in imperfect placement and therefore induces error in the overall pose estimation. Additionally, being printed on paper, the tags are easily damaged, especially when being manipulated by a robot. To resolve this issue we found a way to use 3D-printed blocks with markers printed on them. This provides us not only with more precise placement of the tags but moreover, guarantees more robust tags which are less prone to being damaged during manipulation. The first version of the block consisted of cubes with sharp edges. This leads to the problem of tight trajectory tracking tolerances. Relaxing this constraint the combination of the new compliant block shape and an impedance controller for insertion improves the general performance. It also compensates for small errors in object pose estimation, funneling the trajectories into the desired placement location when they are close enough to the targeted one.

B. Real-World Setup

We use three ultra-high resolution (4K) webcams (Logitech Brio) placed around the scene to track the SL-Block. By using ultra-high resolution images we can place the cameras outside of the working environment of the robotic arm and are still able to detect the AprilTags with sufficiently high accuracy. We calibrated each camera individually using a 6x6 checkerboard method available through the OpenCV library [25]. The cameras have to be oriented in such a way, that the blocks, as well as the workspace, are visible from as many angles as possible. One reason for this is the relatively inaccurate distance estimation for the AprilTags [18]. Having at least one orthogonal view is therefore advantageous to get a better depth estimate for the respectively other cameras. The other reason is to handle occlusions from a single perspective. The derived camera configuration utilized in the end has the cameras placed around the table to the left, right, and front of the robot, facing it. They are mounted at different heights and angled downwards towards the same spot resulting in differing tilts.

C. Object Pose Estimation Pipeline

In this section, we first describe the common basis of our estimation approaches, then a baseline method we use for comparison and lastly our actual block pose estimation. The baseline is included for comparison on the same data. For this, we generate input image streams and truth position values to compare against in simulation. This isolates the algorithm performance from outside influences. The statistical evaluation can be found later on in chapter IV.

Our pose estimation for the SL-Block starts with tag detection. The continuous detection node from apriltag_ros2 library scans each camera stream for suitable tags. All detections are then published to the detection topic of the corresponding camera. The object locator node reads these and on each received detection array the estimations of all detected blocks are updated.

Due to our modeling, we know where on the block each tag is located and which orientation it has relative to our chosen reference tag. We identify the tags by their ids encoded in the marker.

In the following, i is a detected tag and o is another tag. We know the transformations ${}^{o}_{ref}T$ from the reference tag to the other tags from our modeling and the transformations ${}^{cam_c}_{i}T$ from our detections to the detecting camera. First, we calculate the transformation from the reference tag's frame to the cameras c:

$$ {}^{cam_c}_{ref}T = {}^{cam_c}_{i}T \cdot {}^{i}_{ref}T \tag{2} $$

Afterward, we calculate all transformations from the other tag frames to the cameras c, based on this estimated transformation:

$$ {}^{cam_c}_{o}T = {}^{cam_c}_{ref}T \cdot {}^{o}_{ref}T^{-1} \tag{3} $$

As a baseline, we decided to collect these transformations and average them tag-wise. For the translation, a mean is calculated while for the rotation the averaging is done following the maximum likelihood method for quaternion averaging [27]. The problem of orientation ambiguity in marker-based pose estimation, as analyzed by Springer and Kyas [28], can result in discontinuities in the orientation estimation. Specifically, it can lead to the flipping of signs in the orientation estimate of the z-axis. Therefore, such discontinuities are to be expected, and they can affect the accuracy and robustness of the pose estimation process. We discovered this ourselves and decided outlier detection was required for the baseline. We first considered implementing RANSAC [29] but due to the way we implement the estimation, we

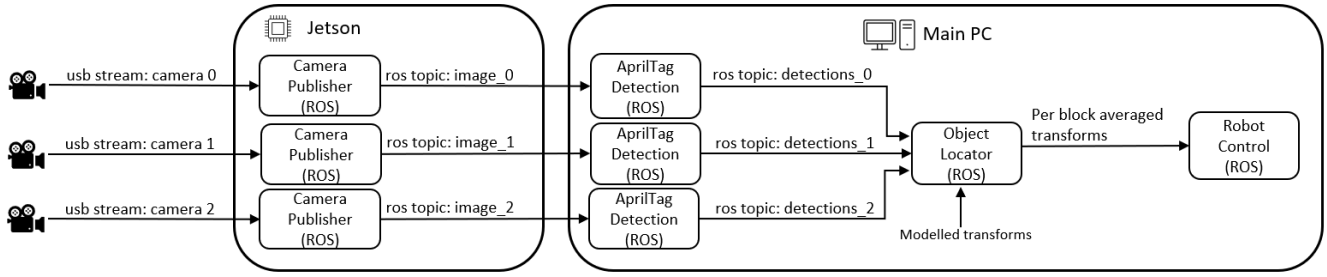

Fig. 4. This flowchart describes our object pose estimation pipeline. Three 4K cameras send images via USB to a Jetson Nano [26]. For each camera, a camera publisher node handles the incoming camera stream and forwards the image to a running AprilTag detection node. For communication between the different nodes, ROS is used, where each node publishes or subscribes to a topic (topics are marked as arrows connecting the different nodes). Running on the main PC, each AprilTag detection node then forwards the detected markers of the corresponding image. The Object Locator Node receives the detections from all three cameras and has access to the model transformations of each block. With that, it updates the transformations of all detected blocks. This estimation can then be used for robot control.

can determine by comparing their ids if a detection and an estimation of two tags are a true match. Thus we can count inliers without resampling. The error function we use to determine estimate quality is a reprojection error (described later). Giving us the best estimation as a byproduct.

In practice, we observed that the best estimation typically had lower errors when compared to the averages of all estimations. This observation aligns with our modeling and justifies our decision to adopt it as our model estimation at that point. However, the results are still not great, being at least marginally translated or rotated from the actual block in the image plane most of the time.

Multi-View Multi-Marker based pose estimation is the method we use. We apply reprojection error minimization building on the recent research of [19] and [20]. To obtain an accurate and reliable estimate of the pose, we optimize the estimated pose of the reference tag at a specific point in time. Specifically, we minimize the squared error between the detected tag poses in the images and the poses calculated using the known transformations from the reference tag to the other tags. This optimization process is performed across all available camera views, and the resulting errors are summed to obtain an overall estimate of the pose. By minimizing the error across all available camera views, we are able to obtain a more robust and accurate estimation of the pose. Different from their work we calculate the reprojection errors in the image plane of the cameras, while they compute the errors directly on the 3D estimates. Our error calculation method is the traditional ”bundle adjustment” error function used in photogrammetry. From an intuitive point of view, it has the advantage that changing the distance estimate for a given camera view has a lower impact on the error than correcting the in-plane translation in the other perpendicular views.

Formally: Given an initial estimate for the reference tag pose in world coordinates ${}^{w}_{ref}T$ as the initial estimate for minimization, we calculate the translations from the other tag frames to the world frame ${}^{w}t_o$. Given the detected translations of all tags in the respective camera frames ${}^{cam_c}t_i$ we transform them and their corresponding estimated other tag locations j,i into the world frame ${}^{w}t_{j,i}$:

$$ {}^{w}t_o = {}^{w}_{ref}T \cdot {}^{ref}t_o \tag{4} $$

$$ {}^{w}t_{j,i} = {}^{w}_{cam_c}T \cdot {}^{cam_c}t_{j,i} \tag{5} $$

And then all of them in all camera frames:

$$ {}^{cam_c}t_o = {}^{w}_{cam_c}T^{-1} \cdot {}^{w}t_o \tag{6} $$

$$ {}^{cam_c}t_{j,i} = {}^{w}_{cam_c}T^{-1} \cdot {}^{w}t_{j,i} \tag{7} $$

The reprojection error of all estimates j,i for detection i to the detected actual positions o of the other tags summed over all camera views in their respective image planes, c can now be calculated using the camera matrices K_c:

$$ \sum_{c=0} ||K_c \cdot {}^{cam_c}t_{j,i} - K_c \cdot {}^{cam_c}t_o||_2^2 \tag{8} $$

is then minimized to get the best estimate for the ${}^{w}_{ref}T$ which is supplied to the optimizer as a vector containing the translation and quaternion values. In the current iteration of the project, we are using scipy's optimization package. The optimizers we used, in the end, were Sequential Least Squares Programming (SLSQP) for constrained optimization using gradient estimation and Broyden–Fletcher–Goldfarb–Shanno (BFGS). The first as we wanted to ensure the boundary condition of the quaternions parameters staying equal to one in their total length. The second as it achieved lower errors, albeit at a higher computation time.

D. Robot Control

In our current setup, we are using a Franka Emika Panda Robot with integrated force sensors which enables the use of impedance control. It is mounted in a dual-arm setup which opens the option for parallel manipulation. However, for simplicity, we are currently only using one arm. For the communication with the robot we are using the provided ROS (Robot Operating System) interface for the libfranka control software in combination with MoveIt!, a ROS based high-level control framework. One of the key advantages of our setup is our use of the ROS communication framework. This allows us to easily integrate our pose estimation pipeline with

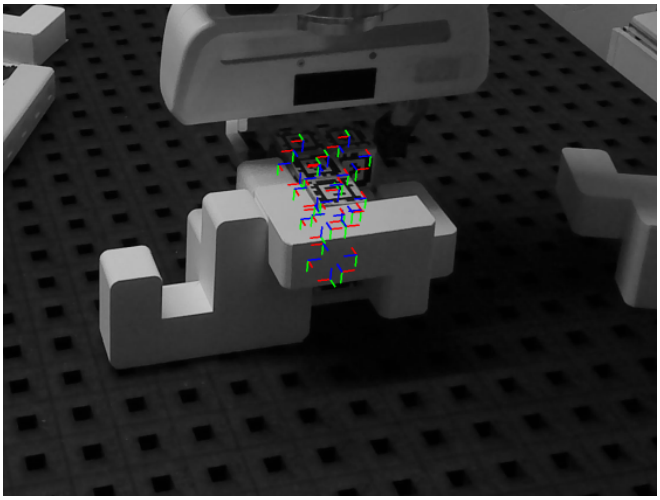

Fig. 5. Estimation of the orientation and position of each tag. To illustrate the algorithm, parts of the block were masked out demonstrating the estimation based on the visible tags.

the rest of our subsystems. By using ROS, we can seamlessly share data and commands between different parts of our system, improving our overall efficiency and effectiveness. To execute the construction plan and correctly stack the SL-Blocks, we plan to use pre-generated trajectories based on the block pickup location and insertion pose in a later stage of this project. The trajectories are provided by the grasshopper engine [30], which is a graphical algorithm editor allowing users to specify high-level design objectives. The trajectories from the grasshopper are communicated through the ROS interface by publishing the trajectories through a new topic. Subscribing to this topic, we can use the joint positions to make the robot execute the trajectory.

In this work, we employ an impedance controller from Franka's operating library libfranka to control the robot. Our pose estimator detects the current pose of the block to be manipulated and passes the information to the control software. Based on the estimated pose, we select a suitable grasp and proceed to grasp the block. The grasped block is then moved to a predefined location and the placing process is initialized. This process can be compared to the well-known peg-in-hole problem, for which there are numerous solutions available in the literature [31][32][33]. We abstract the interlocking of the cubes as a multi-peg insertion problem. Thanks to the low stiffness induced by the impedance controller and the compliant structure of the blocks, intuitive insertion strategies can be employed, resulting in the block sliding smoothly into its final position.

To be able to have seamless communication between all components, we have to make sure that the estimated pose of the block is in the right coordinate frame. Initially, the pose of the block will be relative to the camera which is used to estimate the location. This pose needs to be transferred in the corresponding frame of reference of the robot such that the robot can successfully approach the block in its frame. To accomplish this a method called hand-eye-calibration is used. The orientation of the robot and the camera to each other is calculated which then can be used to transform estimated block poses in the robotic frame of reference [34]. This calibration needs to be done with great accuracy to reduce the induced amount of uncertainty in the system.

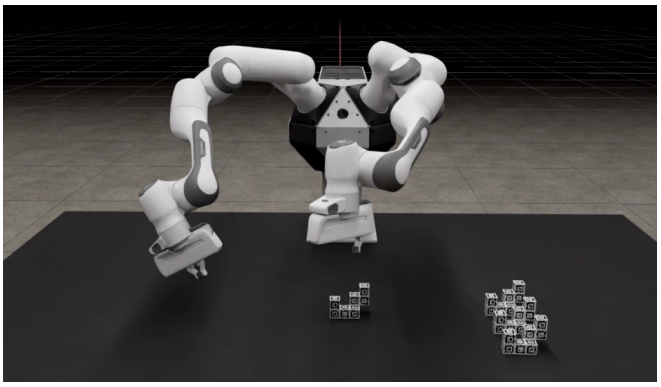

Fig. 6. Scene from inside the simulation with the dual-arm robot as well as the SL-Block.

E. Simulation

Parallel to the work on the real robot, we integrate a digital twin of our real-world setup. This enables us to test new approaches but also makes it possible to use it as a learning platform for RL algorithms. Conducting experiments on a real robot system not only takes time but also poses danger to the people around as well as the robot itself. Wrong configurations on a robot can lead to havoc and disaster which we want to prevent as much as possible. Even though the risk imposed through the robot arm that we are using is quite low, security aspects should always be kept in mind. By using a digital twin in a simulation we can reduce the risk of bad configurations on the real robot by first testing them in the simulated environment. In addition to this, we also gain greater flexibility when it comes to testing new ideas as we are not limited by the constraints of working with a real robot. In our case, we have two important requirements for the simulation to fulfill.

First, it has to be able to generate photo-realistic images of the SL-Block and its structures. This is needed so we can test and evaluate our detection pipeline with synthetic images and expect it to perform similarly well in a real-world environment. Therefore we implemented a digital twin of the SL-Block with the same AprilTag configuration as in the real setup.

Second, we need a physically accurate simulation of the entire environment. This especially refers to the physical properties of the SL-Block and its interaction with other blocks and manipulation through the robot.

1) NVIDIA IsaacSim: We are using NVIDIA Isaac Sim [35], a state-of-the-art robotics simulation platform. It allows us to generate photo-realistic images by using the latest advancements in real-time ray tracing and physically-based rendering. Additionally, we can work with physically-accurate simulation by leveraging the NVIDIA PhysX engine

Criterion \Method	Average Estimates Single-View	Best Estimate Single-View	SLSQP OPT Single-View	BFGS OPT Single-View
3D translation error ref. marker in mm	558	558	84	16.9
3D translation error z only ref. marker in mm	476	475	72	16.1
3D translation error xy only ref. marker in mm	288	288	34	4.1
Cumulative 2D error single-view in pixels squared	158	110	13.68	1.47
Computation time in ms	12	6	44	171

TABLE I

A COMPARISON OF THE METHODS OVER 3000 MEASUREMENTS

[36]. Regarding the ability to use the simulation as a training platform for reinforcement learning, NVIDIA IsaacSim provides a new way to speed up the training of such models by 2-3 orders of magnitude compared to traditional techniques. This is done using the recently published Isaac Gym [37] which removes the CPU bottleneck during training and directly passes the physics buffer via the GPU to the training network which also resides on the GPU. We use the provided Python interface as well as the ROS connector to interact with the simulation.

IV. EVALUATION

A. Object Pose Estimation

In this section, we compare the performance of the different approaches used through the different stages of our progress. The data for the evaluation was generated in simulation to focus on the method of object pose estimation instead of the setup as a whole. The 3D error in translation was calculated with the help of the simulation software. We published the translation of the reference tag from it and compared it to our estimates.

As one can see the BFGS single-view optimization is the clear winner in all categories but the computation time. The remaining 4.1 mm deviation in the x and y axis from the truth value shows the limitations our camera resolution imposes on the approach, as the error is almost at it's theoretically possible minimum 0. The 0 error might not be possible even given the detections of the tags. The other optimization algorithm SLSQP which we used in combination with a constraint ensuring our quaternion stays one, is 5 times faster but doesn't find as good a solution in this case.

As is evident in the table above all non-multiview systems are inaccurate in the depth aspect of the estimation, even when they are accurate in the 2d error. Unfortunately, we could not get our multi-view estimation implementation working in time for the deadline. Therefore, evaluation of it will be left for future work.

A rosbag containing the 4k image messages, true translation vector (tf) messages from the cameras to the reference tag, tf messages from the cameras to the world origin and lastly, the apriltag_ros detections messages, on which we did all of the evaluations will be shared with any interested party.

V. CONCLUSION AND OUTLOOK

In this work, we present our plans and progress towards developing a new approach to automating the construction of predefined structures assembled from SL-Blocks utilizing the Franka Emika robot. We tackle high-precision tracking of the SL-Block by modifying recent multi-marker multi-view-based object tracking algorithms. Instead of the transformations between the markers of our target object being unknown and determined in a calibration step as in previous methods, the blocks are modeled with transformations for the markers predefined. We cover all of the block's faces with fiducial markers (AprilTags). This reduces the probability that all markers are occluded from one of the viewing angles and improves the robustness of our pose estimation. Most of the setup concerning hardware, the experimental arrangement, and software integration via ROS both in simulation and in the lab has been completed, preparing the next stage of the sequential assembly project. Our tracking allows us to accurately determine the pose of the SL-Blocks. The hand-eye coordination for the robot and the cameras enable us to determine said pose in the robot frame. Based on this we are capable of executing pick and place operations independent of the blocks start pose, as long as it's in the working area and one of the programmed picking operations is kinematically feasible.

The current stage of the project can be seen as a proof of concept for the construction pipeline. In the next stage of the project the integration with the existing trajectory generation from grasshopper for precomputed construction plans and completing the Multi-View Pose Estimation integration will be the focus.

REFERENCES

[1] M. Ben-Ari and F. Mondada, Robots and Their Applications, 01 2018, pp. 1–20.

[2] M. Gharbia, A. Chang-Richards, Y. Lu, R. Y. Zhong, and H. Li, "Robotic technologies for on-site building construction: A systematic review," Journal of Building Engineering, vol. 32, p. 101584, Nov. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352710220313607

[3] Y. W. D. Tay, B. Panda, S. C. Paul, N. A. N. Mohamed, M. J. Tan, and K. F. Leong, "3d printing trends in building and construction industry: a review," Virtual and Physical Prototyping, vol. 12, no. 3, pp. 261–276, 2017. [Online]. Available: https://doi.org/10.1080/17452759.2017.1326724

[4] A. V. Dyskin, E. Pasternak, and Y. Estrin, "Mortarless structures based on topological interlocking," Frontiers of Structural and Civil Engineering, vol. 6, no. 2, pp. 188–197, Jun. 2012.

[5] S.-G. Shih, "On the hierarchical construction of sl blocks," Sigrid Adriaenssens, Fabio Gramazio, Matthias Kohler, 2016.

[6] B. Wibranek, Y. Liu, N. Funk, B. Belousov, J. Peters, and O. Tessmann, "Reinforcement learning for sequential assembly of sl-blocks," 09 2021.

[7] N. Funk, G. Chalvatzaki, B. Belousov, and J. Peters, "Learn2assemble with structured representations and search for robotic architectural construction," in Proceedings of the 5th Conference on Robot Learning, 2022, pp. 1401–1411.

[8] Y. Liu, B. Belousov, N. Funk, G. Chalvatzaki, J. Peters, and O. Tessman, "Auto(mated)nomous assembly," in International Conference on Trends on Construction in the Post-Digital Era, 2022, pp. 167–181.

[9] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[10] P. Song, Z. Fu, L. Liu, and C.-W. Fu, "Printing 3d objects with interlocking parts," Computer Aided Geometric Design, vol. 35-36, pp. 137–148, 2015, geometric Modeling and Processing 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167839615000436

[11] S.-G. Shih, "The art and mathematics of self-interlocking sl blocks," in Proceedings of Bridges 2018: Mathematics, Art, Music, Architecture, Education, Culture, 2018, pp. 107–114.

[12] M. Pfanne, M. Chalon, F. Stulp, and A. Albu-Schäffer, "Fusing Joint Measurements and Visual Features for In-Hand Object Pose Estimation," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3497–3504, Oct. 2018.

[13] K. Pauwels and D. Kragic, "SimTrack: A simulation-based framework for scalable real-time object pose detection and tracking," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2015, pp. 1300–1307.

[14] V. Abhijith and A. B. Raj, "Robot Operating System based Charging Pad Detection for Multirotors," in 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), May 2020, pp. 1151–1155.

[15] G. Yu, Y. Liu, X. Han, and C. Zhang, "Objects Grasping of Robotic Arm with Compliant Grasper Based on Vision," in Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering, ser. CACRE2019. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3351917.3351958

[16] N. Tian, A. K. Tanwani, J. Chen, M. Ma, R. Zhang, B. Huang, K. Goldberg, and S. Sojoudi, "A Fog Robotic System for Dynamic Visual Servoing," in 2019 International Conference on Robotics and Automation (ICRA), May 2019, pp. 1982–1988, iSSN: 2577-087X.

[17] S. Khattak, C. Papachristos, and K. Alexis, "Marker Based Thermal-Inertial Localization for Aerial Robots in Obscurant Filled Environments," in Advances in Visual Computing, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, M. Turek, S. Ramalingam, K. Xu, S. Lin, B. Alsallakh, J. Yang, E. Cuervo, and J. Ventura, Eds. Cham: Springer International Publishing, 2018, pp. 565–575.

[18] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial Markers for Pose Estimation," Journal of Intelligent & Robotic Systems, vol. 101, no. 4, p. 71, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s10846-020-01307-9

[19] H. Sarmadi, R. Muñoz-Salinas, M. A. Berbís, and R. Medina-Carnicer, "Simultaneous Multi-View Camera Pose Estimation and Object Tracking With Squared Planar Markers," IEEE Access, vol. 7, pp. 22 927–22 940, 2019.

[20] M. Trinh, J. Padhan, N. V. Navkar, and Z. Deng, "Preliminary Design and Evaluation of an Interfacing Mechanism for Maneuvering Virtual Minimally Invasive Surgical Instruments," in 2022 International Symposium on Medical Robotics (ISMR), Apr. 2022, pp. 1–7, iSSN: 2771-9049.

[21] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," IEEE Transactions on Robotics, vol. 31, no. 4, pp. 850–863, 2015.

[22] "Franka Panda," Nov. 2022. [Online]. Available: https://www.franka.de/research

[23] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). IEEE, May 2011, pp. 3400–3407.

[24] M. D. Dogan, A. Taka, M. Lu, Y. Zhu, A. Kumar, A. Gupta, and S. Mueller, "Infraredtags: Embedding invisible ar markers and barcodes using low-cost, infrared-based 3d printing and imaging tools," in Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–12.

[25] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[26] "NVIDIA Jetson Nano für KI-Anwendungen in der Peripherie und Bildung." [Online]. Available: https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-nano/

[27] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," Journal of Guidance, Control, and Dynamics, vol. 30, no. 4, pp. 1193–1197, 2007. [Online]. Available: https://doi.org/10.2514/1.28949

[28] J. Springer and M. Kyas, "Evaluation of orientation ambiguity and detection rate in april tag and whycode," 2022. [Online]. Available: https://arxiv.org/abs/2203.10180

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.

[30] S. D. c. t. N. Network, "Grasshopper." [Online]. Available: https://www.grasshopper3d.com/

[31] T. Tang, H.-C. Lin, and M. Tomizuka, "A learning-based framework for robot peg-hole-insertion," in Dynamic Systems and Control Conference, vol. 57250. American Society of Mechanical Engineers, 2015, p. V002T27A002.

[32] Y. Huang, X. Zhang, X. Chen, and J. Ota, "Vision-guided peg-in-hole assembly by baxter robot," Advances in Mechanical Engineering, vol. 9, no. 12, p. 1687814017748078, 2017.

[33] J. F. Broenink and M. L. Tiernego, "Peg-in-hole assembly using impedance control with a 6 dof robot," in Proceedings of the 8th European Simulation Symposium, 1996, pp. 504–508.

[34] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pp. 345–358, 1989.

[35] "Isaac Sim," Dec. 2019. [Online]. Available: https://developer.nvidia.com/isaac-sim

[36] "NVIDIA PhysX 4.5 and 5.0 SDK," Nov. 2018. [Online]. Available: https://developer.nvidia.com/physx-sdk

[37] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning," Tech. Rep., Aug. 2021, arXiv:2108.10470 [cs] type: article. [Online]. Available: http://arxiv.org/abs/2108.10470