Task and Motion Planning for Sequential Assembly - Towards Multi-View Multi-Marker Object Pose Estimation

Paul Hallmann *[1] , Patrick Siebke *[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 are automatically constructed 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. They are modified with the additional assumption of known relative transformations between markers on the blocks. Our proof of concept setup nearing completion is described alongside our vision for improvements 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 by a robot. 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 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 nowadays 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 solved 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. To the best of our knowledge no similar works have been published thus far.

As joining multiple blocks into a structure requires sub-millimeter 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 placing fiducial markers on 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 that follows input trajectories.

This is where our work picks up. In the first stage of this project our focus lies on the development of the marker based tracking of the 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)[6].

Secondary to this we describe our plans to build on this first stage to arrive at a more sophisticated solution. This addresses potential limitations and issues that may arise with the first version.

*All authors contributed equally

[1]TU Darmstadt

Lastly, to prepare for future work a digital twin of the system is set up and maintained. It will enable us to freely experiment without safety concerns, help with evaluation, and possibly, later on, provide a platform for reinforcement learning (RL) to learn 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 printers working volume. To overcome this issue, recent work like Song et al. [7] 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 [8]. 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 [9] 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. [10]. 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 [11], [12], [13] or even Simultaneous Localization and Mapping [14] in controlled environments. Of the currently available flat rectangular tag variant designs, AprilTag seems to perform best [15] and is thus used for our project. Recently marker bundle-based object trackers have shown remarkable pose estimation accuracy. In Sarmadi et al. [16] 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 get all of this. In [17] 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, both are used while tracking 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 all visible marker poses using their 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.

III. Our Approach: Setup, Plans and Progress

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 [18]. Next, we describe the object pose estimation pipeline and how task and motion planning will be done. Lastly, we describe how the Isaac-Sim Simulator is used to provide a simulation platform to first test our approach before applying it to the real robot.

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

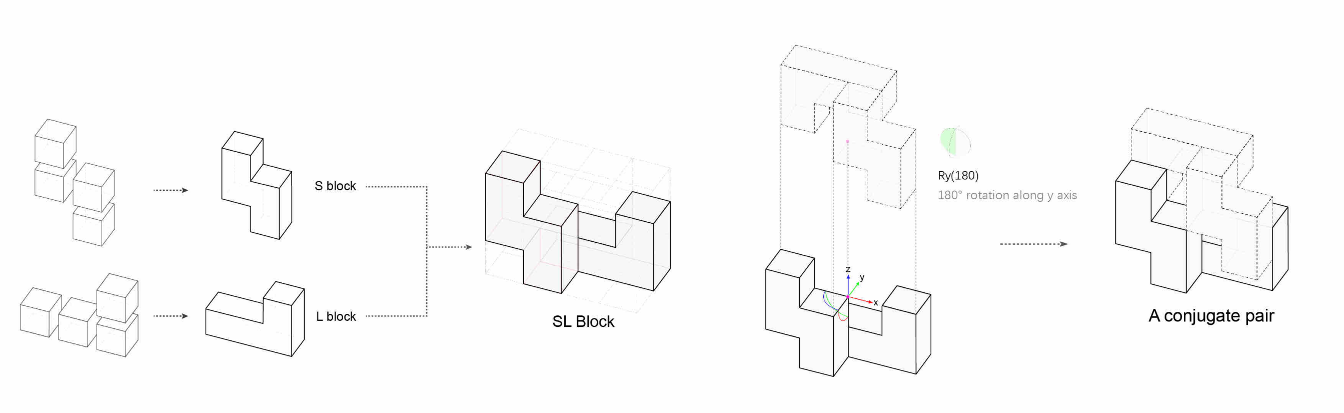

Fig. 1. Image of an SL-Block (Image by DDU). Introduced by [5] it consists of an S-shaped and an L-shaped tetracube attached to each other.

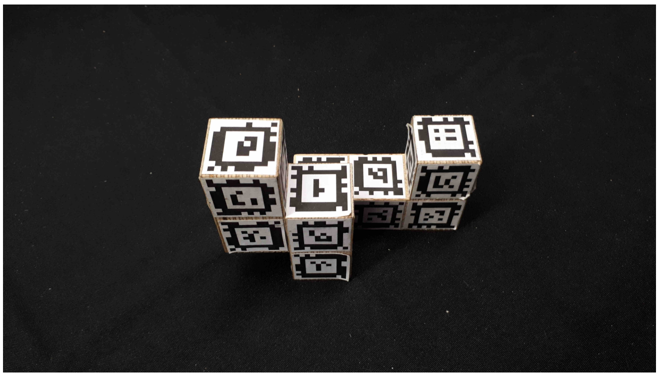

Fig. 2. Image of an SL-Block [5] with AprilTags.

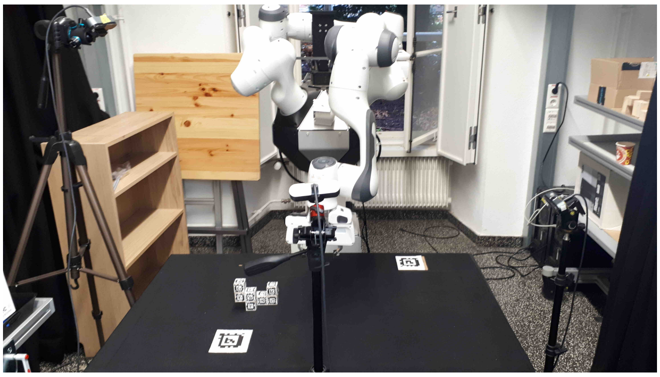

Fig. 3. Experiment setup containing the dual arm robot and the three cameras positioned around the table.

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 [19]. A unique AprilTag is applied to each face of the block as can be seen in Figure 2. To get an exact mapping between 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.

Currently, the SL-Block is still manufactured using wooden cubes and the tags are glued to it manually. Applying the tags manually to the SL-Block leads to not equally placed tag positions which lead to inaccuracies. To eliminate this inaccuracy, a 3D printed version of the SL-Block will be used in the future which is currently in development.

B. Real World Setup

To detect the SL-Block we use three ultra-high resolution (4K) webcams (Logitech Brio) placed around the scene. By using ultra-high resolution images we are able to 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 [20]. 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 [15]. 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 single perspectives. 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. To process the incoming data stream from the cameras we use a Nvidia Jetson Nano [21]. The resulting image streams are then published via ROS. The main pc handles the image streams performing tag detection. The detections are used for object pose estimation and tracking of the SL-Block. A control pc with an installed real-time kernel is connected to the Franka Emika robot in a dual arm setup to then manipulate the position of the SL-Block.

C. Object Pose Estimation Pipeline

The pose estimation for the SL-Block starts with the tag detection. The continuous detection node from the 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 is

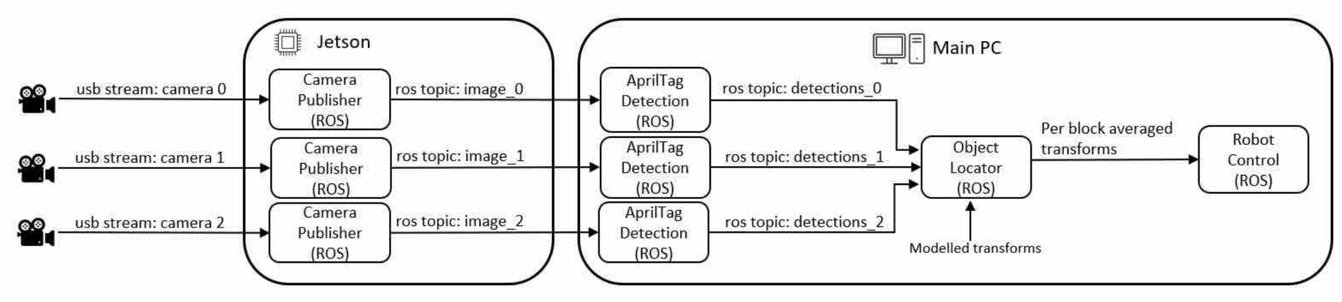

Fig. 4. This flowchart describes our object pose estimation pipeline. Three 4K cameras send images via usb to a Jetson Nano [21]. For each camera a camera publisher node handles the incoming camera stream and forwards the image to a running AprilTag detection node. For communication between the different 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 block. This estimation can then be used for robot control.

subscribing to this topic 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 base tag. We identify the tags by their ids encoded in the marker. For each tag detection the known transformations to the reference tag are used to obtain an estimation regarding all other tag poses:

In the following we will treat the tags $id = id \mod 34$ (id_o for other tags) where id is the id of a detected tag. We know the transformations ${}^{id}_{ref}T$ from the reference tag to the other tags from our modeling and the transformations ${}^{cam_i}_{id}T$ from our detections to the detecting camera. First, we calculate the transformation from the reference tag's frame to the cameras i:

$$ {}^{cam_i}_{ref}T = {}^{cam_i}_{id}T \cdot {}^{id}_{ref}T \tag{1} $$

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

$$ {}^{cam_i}_{id_o}T = {}^{cam_i}_{ref}T \cdot {}^{id_o}_{ref}T^T \tag{2} $$

Currently these transformations are collected and averaged 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 as described in [22]. As a next step, we plan to average all our camera perspectives transformed into the table frame, which will be used as the global frame. Given the transformations to each camera i from tags with id o ${}^{cam_i}_{id_o}T$ and a frame transformation from the camera's to the table's frame ${}^{tab}_{cam_i}T$, we can now calculate the tag pose estimations in the table frame:

$$ {}^{tab}_{id_{o,i}}T = {}^{tab}_{cam_i}T \cdot {}^{cam_i}_{id_o}T \tag{3} $$

The averaging will follow the same approach as for single cameras before resulting in sufficiently accurate pose estimation for testing of the full construction pipeline.

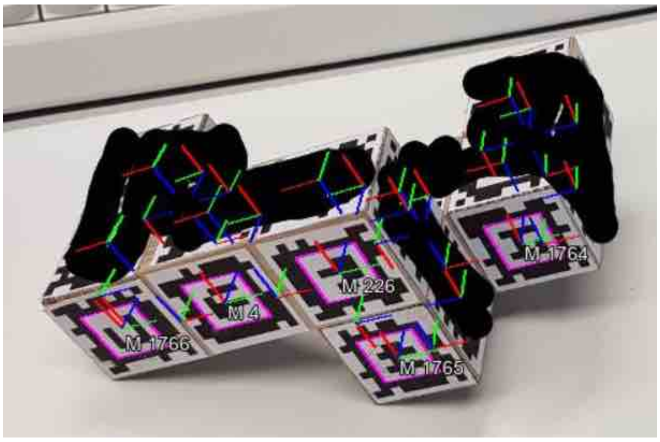

Fig. 5. Estimation of the orientation and position of each tag. To illustrate the algorithm, parts of the block were masked out demonstrating the estimation based on the visible tags.

Future Work for Object Pose Estimation: Even with view merging we expect the accuracy of the current pose estimation approach to be less accurate than the recent methods utilizing optimization as their results are impressive. Therefore we plan to apply reprojection error minimization following [16] and [17]. Different from those approaches, we will not use the Levenberg-Marquardt optimization algorithm but rather Dogleg as it was shown to produce almost equivalent results at a significant speedup for this kind of error function [23]. In other words, we plan to optimize over the estimated pose of the reference tag for one point in time, so that the squared error between each detected tag pose in the images and the poses calculated via the known transformations from the reference tag to the other tags are minimized, summed over all available camera views transformed to the table frame.

Formally: Given the average estimate for the reference tag frame in table coordinates ${}^{tab}_{ref}T$ as the initial estimate for minimization, we can calculate all other transformations from the tag frames into the table frame. Given the detected translations of all tags in the respective camera frames ${}^{cam_i}t_{id_o}$

we can calculate them in the table frame ${}^{tab}t_{id_{o,i}}$:

$$
{}^{tab}t_{id_o} = {}^{tab}_{ref}T \cdot {}^{ref}t_{id_o} \quad (4)
$$

$$
{}^{tab}t_{id_{o,i}} = {}^{tab}_{cam_i}T \cdot {}^{cam_i}t_{id_o} \quad (5)
$$

The reprojection error over the tag translations in the table frame summed over all camera views i:

$$
\sum_{i=0} ||{}^{tab}t_{id_{o,i}} - {}^{tab}t_{id_o}||_2^2 \quad (6)
$$

is then minimized to get the best estimate for the ${}^{tab}_{ref}T$ which is supplied to the optimizer as a vector containing the translation and quaternion values.

Furthermore, we plan to compare this method with a variant where we first reproject the 3D estimates of the tag poses back into the images and optimize over the summed error between points in all three images. This is the traditional "bundle adjustment" error used in photogrammetry, the science of gathering reliable information about physical objects and the environment through for example imagestream analyzation, to join multiple observations of a scene. From an intuitive point of view, it has the advantage that bad depth estimates from one camera should show low resistance to small changes as image coordinate wise it will have little impact, less the farther away the detection is from the camera due to the scaling of depth.

One advantage of optimizing over the reference tag pose and calculating the remaining tag poses is that the resulting transformations will be consistent with our model, which is not the case in general when averaging over all estimates for all tags. Lastly appending an extended Kalman filter as a final stage of our object tracking pipeline as it was done in [17] to smooth our outputs is intended.

D. Task and Motion Planning & Robot Control

To execute the construction plan and correctly stack the SL-Blocks, we use pre-generated trajectories based on the block pickup location and insertion pose. The trajectories are provided by the grasshopper engine [24], 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.

The robot will be controlled using a supplied controller from Franka's operating library libfranka. Later on, we will utilize our block tracking beyond the pre-pickup block pose estimation for better robot control. We will use the detected block trajectory to correct the trajectory of the gripper, so that the block gets exactly where it should, instead of focusing on the gripper movement. The task and motion planning approach presented by Braun et al. [25] combines search-based planning over high-level discrete actions and low-level trajectory optimization, utilizing geometric heuristics for the search and applying the receding horizon paradigm to gain performance. It should be a good fit for our problem as we have to combine high-level decisions such as "When do we grab which block?" with low-level planning for "How do we move the block for insertion?". This would enable the robot system to no longer be reliant on a pre-generated assembly plan and instead plan effectively given the actual situation on the assembly surface. In combination with the block tracking, it will also enable us to avoid collisions with the partially assembled structure.

We further plan to extend this idea with a separate controller just for inserting blocks into the partial structure. Here we are looking to apply RL, based on dynamic motion primitives (DMPs). For block insertion there are multiple insertion techniques suited to different scenarios. They are quite different to each other. Reinforcement learning algorithms have already been successfully applied to efficiently and robustly optimize the parameters of DMPs for example the domain of autonomous driving [26] or for everyday pick-and-place tasks [27] and recent work shows (Pro-) DMPs are applicable to block insertion tasks [28],[29]. Combined this leaves us confident that an efficient and robust stacking algorithm can be created by decomposing the different insertion techniques into DMPs.

E. Simulation

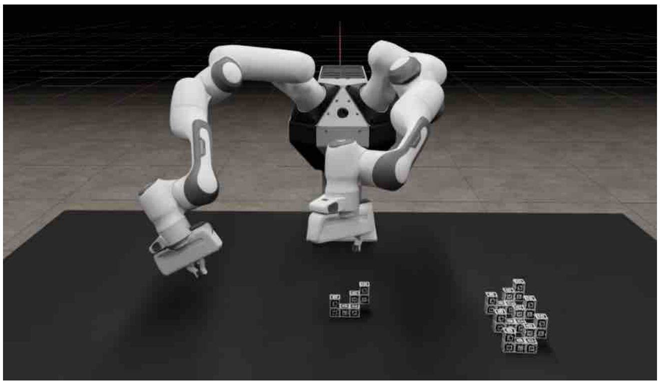

Fig. 6. Scene from inside the simulation with the dual arm robot as well as the SL-Block.

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 does not only take 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 [30], 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 [31]. 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 [32] 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. 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, experimental arrangement, and software integration via ROS both in simulation and in the lab has been completed, preparing for the future implementation of our planned solutions. Intermediate results of the partially implemented object tracking encourage us to extend our work to further improve the pose estimation and combine it with motion planning.

Due to the incomplete implementation state of our object tracking, we can't yet compare it with related work.

Object tracking is only the first necessary step towards our planned proof of concept version of the construction pipeline. Once it's fully operational we can proceed with improving its performance through the planned measures described throughout this work.

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

[7] 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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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

[13] 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.

[14] 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.

[15] 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

[16] 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.

[17] 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.

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

[19] 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.

[20] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[21] “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/

[22] 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

[23] M. Lourakis and A. Argyros, “Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment?” in *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, vol. 2, 2005, pp. 1526–1531 Vol. 2.

[24] S. D. c. t. N. Network, “Grasshopper.” [Online]. Available: https://www.grasshopper3d.com/

[25] C. V. Braun, J. Ortiz-Haro, M. Toussaint, and O. S. Oguz, “Rhh-lgp: Receding horizon and heuristics-based logic-geometric programming for task and motion planning,” 2021. [Online]. Available: https://arxiv.org/abs/2110.03420

[26] B. Wang, J. Gong, and H. Chen, “Motion primitives representation, extraction and connection for automated vehicle motion planning applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3931–3945, 2020.

[27] F. Stulp, E. Theodorou, and S. Schaal, “Reinforcement learning with sequences of motion primitives for robust manipulation,” *Robotics, IEEE Transactions on*, vol. 28, pp. 1360–1370, 12 2012.

[28] J. Carvalho, D. Koert, M. Daniv, and J. Peters, “Residual robot learning for object-centric probabilistic movement primitives,” 2022. [Online]. Available: https://arxiv.org/abs/2203.03918

[29] J. Su, Y. Meng, L. Wang, and X. Yang, “Learning to assemble noncylindrical parts using trajectory learning and force tracking,” *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2021.

[30] “Isaac Sim,” Dec. 2019. [Online]. Available: https://developer.nvidia.com/isaac-sim

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

[32] 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