Projector-Guided Non-Holonomic Mobile 3D Printing

Xuchu Xu[1], Ziteng Wang[2], Chen Feng[1†]

Abstract— Fused deposition modeling (FDM) using mobile robots instead of the gantry-based 3D printer enables additive manufacturing at a larger scale with higher speed. This introduces challenges including accurate localization, control of the printhead, and design of a stable mobile manipulator with low vibrations and proper degrees of freedom. We proposed and developed a low-cost non-holonomic mobile 3D printing system guided by a projector via learning-based visual servoing. It requires almost no manual calibration of the system parameters. Using a regular top-down projector without any expensive external localization device for pose feedback, this system enabled mobile robots to accurately follow pre-designed millimeter-level printing trajectories with speed control. We evaluate the system in terms of its trajectory accuracy and printing quality compared with original 3D designs. We further demonstrated the potential of this system using two such mobile robots to collaboratively print a 3D object with dimensions of 80 *cm* × 30 *cm* size, which exceeds the limitation of common desktop FDM 3D printers.

I. INTRODUCTION

Fused deposition modeling (FDM) is a common type of additive manufacturing (AM) method. A conventional FDM 3D printer is typically implemented as a gantry system enabling 3 degrees-of-freedom (DOFs) control of a printhead to emit fused materials, ceramic, or even concrete precisely to designed printing positions layer by layer. Such a gantry-based printer cannot print large-scale objects, with typically size limitations of $350 \times 350 \times 300$ mm^3 for desktop printers and $900 \times 600 \times 900$ mm^3 for industrial ones. A gantry system also makes it difficult for multiple printers to collaborate to achieve a faster printing speed. To remove these limitations, installing the printhead as the end-effector of a mobile manipulator is an appealing option.

However, a mobile 3D printer brings several challenges to the software and hardware design of such a robotic system. First, we can no longer benefit from stepper motors on the gantry to estimate the printhead position, which is needed for the computer numerical control of the printer. For a desktop FDM printer with a millimeter-level nozzle diameter, the localization and control of the printhead must achieve a millimeter or even sub-millimeter level to ensure successful printing. Otherwise, the layer-by-layer mechanism could easily fail before completing the printing.

Although millimeter-level positioning in a room-scale indoor environment can be achieved by camera-based precise localization systems, such as Vicon®, the issues of such a solution are the system cost and the non-trivial setup and

[1] New York University, Brooklyn, NY 11201, USA {`xuchuxu`, `cfeng`}`@nyu.edu`
[2] `tonywangziteng@163.com`
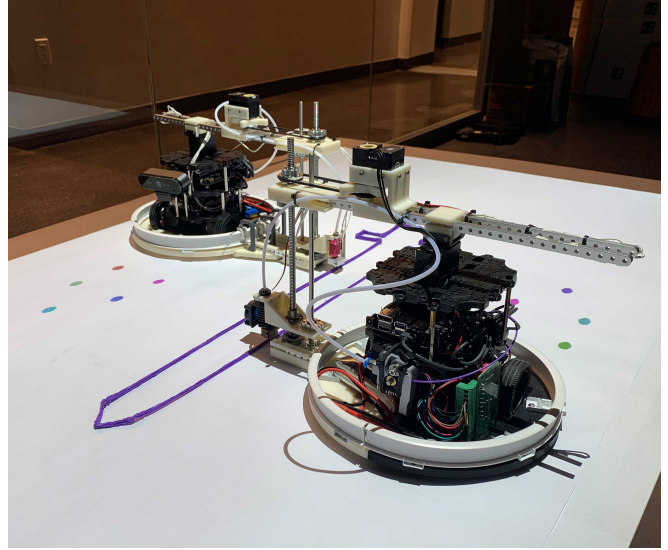[†] Chen Feng is the corresponding author.

Fig. 1. Collaborative printing of two mobile 3D printers in our system.

calibration. To support easy and friendly deployment, is it possible to develop a cost-efficient localization and control system for our mobile 3D printer without a complex manual system calibration process?

In addition to the software challenge, the hardware design of the robotic system is also non-trivial. Unlike regular mobile manipulators, which are often used for object gripping tasks, the aforementioned accuracy requirement means that the mobile platform must have good stability during its movement to minimize the vibration of the printhead. This could lead to a trade-off with the convenience of the mobile platform's planning and control: should we use a holonomic or a non-holonomic mobile robot?

To address these challenges, we propose and develop a low-cost non-holonomic mobile 3D printing system. The localization and control system is guided by a regular low-cost top-down projector equipped with learning-based visual servoing. The mobile platform is a non-holonomic two-wheel differential drive robot with a camera for visual servoing.

The following are our main contributions:

- We propose a low-cost, projector-guided, non-holonomic mobile 3D printing system with almost no manual calibration and easier ground settings.
- We present a learning-based visual servoing method for differential drive robot control with millimeter-level trajectory accuracy.
- We conduct both simulated and real-world experiments to validate our system prototype in terms of trajectory control accuracy and printing quality.

- We successfully printed a 3D object with dimensions of $800\ mm \times 300\ mm$ using two prototype mobile printers.

II. RELATED WORK

Mobile 3D printing. The accuracy of traditional gantry-based 3D printing relies on counting the steps of a stepper motor's output. For mobile 3D printing, the biggest challenge is how to localize a robot, because the accuracy and reliability of the wheel encoder and Inertial measurement unit (IMU) cannot provide satisfactory position feedback. Therefore, some pioneers have explored different types of localization methods for mobile 3D printing systems.

Marques et al. [1] proposed an omni-wheel 3D printing robot and a grid-based 3D printing system. Their mobile robot scans a grid on the ground through an optical sensor to perceive its position when printing. In addition, this system allows multiple robots to perform collaborative printing by operating on the same grid building plane. To avoid collisions in collaborative printing, they presented an automatically generated scheduling approach based on the dependency tree in [2]. To solve the power supply problem, Currence et al. [3] invented a floor power module with installed brushes on the robot that connected to the powered floor surface.

Zhang et al. [4] used a holonomic-based mobile manipulator for mobile 3D concrete printing. They cut the printing model into several parts and moved the mobile platform to the target workspace for printing execution. They later updated their control algorithm [5]. This algorithm could obtain localized feedback and adjust the motion error from the AprilTags [6] on the ground. Since concrete needs to be pumped through a pipe, Zhang and Pham [7] designed an algorithm that included four different modes to solve the motion planning problem for multiple tethered planar mobile robots. Similar to [4], Lv et al. [8] proposed a holonomic mobile printing method by switching the workspace and used the contour of the printed object to estimate the pose of the mobile platform and update the mobile platform control.

Holonomic vs. non-holonomic mobile platform. The main differences between the above mobile 3D printing systems and ours are twofold. First, we use a non-holonomic robot as our printing platform. According to previous printing results [1], although omni-wheel-based holonomic robots can be more convenient in terms of control and path planning, they still cannot avoid shifting between printed layers caused by a *slip on the orthogonal direction of the robot motion.* Our differential drive robot uses standard wheels and does not suffer from slippage in that direction. This also allows odometry based on wheel encoders, which is more difficult in omni-wheel robots. The second difference is the localization system. Their printing platforms operate in an absolute Cartesian coordinate frame. The accuracy of those systems relies on the grid, *which requires manual setup on the ground.* Our system uses projector-based visual servoing control, which requires no robot pose feedback in a Cartesian frame, nor do we need any non-trivial manual calibration or setup.

Visual servoing control. Prior work on visual servoing (VS) control generally fell into three categories: position-based, image-based, and fusion approaches [9, 10]. These three types of visual servoing methods control the robot movement by reducing the error between the observed image obtains error feedback from different spaces.

Position-based visual servoing (PBVS) [11–14] evaluates the error feedback from the observed object in the 3D Cartesian space. This is called 3D visual servoing. These types of methods require the camera intrinsic parameters to convert all the observed object pose to a 3D coordinate system. Therefore, the accuracy of the camera calibration and the robot model will directly affect the control output. In image-based visual servoing (IBVS) [15–17] control, the robot will minimize the error directly in the image space. This is also called 2D visual servoing. Because of the estimation error in image space, these types methods are insensitive to the calibration errors of the cameras on the robot. The disadvantages of IBVS are lost feature points during the rotation of the robot. The singularity of the Jacobian matrix could also cause control failures. Combining the advantages of both, Malis et al. [10] proposed a 2.5D visual servoing using the homography between planar targets. Our method is related to but different from all these methods. Similar to Lee et al. [18], *we use self-supervised deep learning to automatically discover the interactive matrix, thereby removing all manual calibration processes in the system.*

III. SYSTEM DESIGN

As shown in Figure 2, our mobile 3D printing system included a mobile platform, a top-down projector and a flat printing surface. Our mobile platform was composed of a TurtleBot and a robot arm with an FDM hotend kit. The TurtleBot provided continuous movement, and the robot arm helped the nozzle reach positions on the printing surfaces. We used an entry-level projector, BenQ MS535A SVGA, to project dynamic point pattern as the observation reference on the printing surface. The mobile platform's on-board computer computed the printing trajectory and velocity information from a projected image. The flat surface as a projection screen provided a borderless build plate for the entire mobile 3D printing system.

A. Mechanical Design

Our mobile platform was developed on Robotis TurtleBot3 burger. TurtleBot3 burger is a compact size differential drive mobile robot. It operates two dynamic wheels powered by Dynamixel XL430-W250. The original design of TurtleBot3 included stacked motors with a battery, driver board, and single-board computer(SBC) separated by four plastic plates. As shown in Figure 3, a 360° rotating robot arm was installed on the top layer. We expanded the bottom layer with an additional four parts parts such that the whole bottom became a circular shape. It is noteworthy that the bottom plates were composed of two symmetric semicircular 3D printing parts in our original design. However, due to the size limitations of the 3D printer, we finally divide it into four parts to meet the size limitation requirement. To avoid robot arm motion in the radial direction, a flexible curtain track was installed
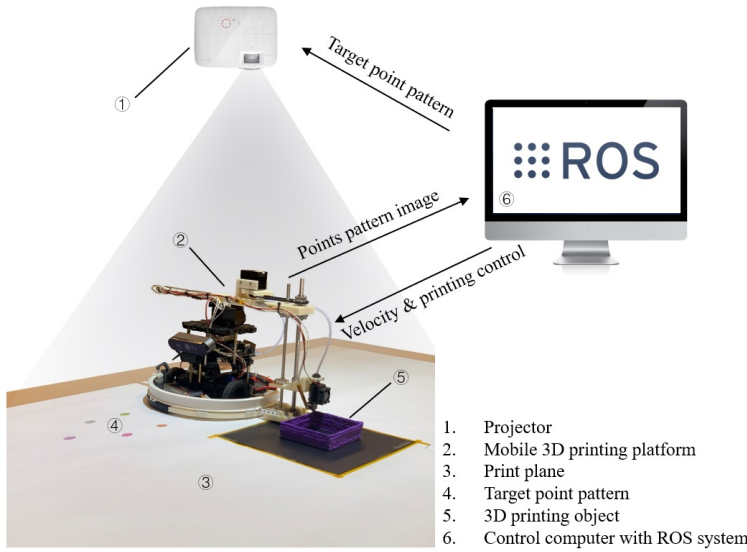
Fig. 2. System settings and operation workflow. Left: projector-guided mobile 3D printing diagram. Right: training pipeline (green blocks) & printing pipeline (blue blocks).
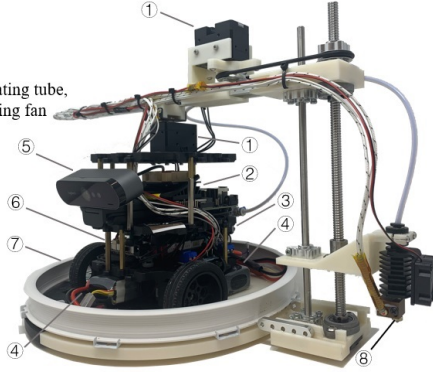
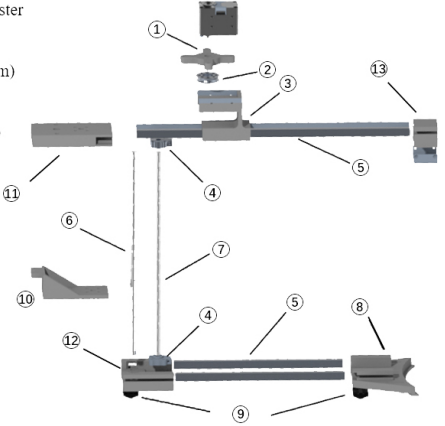Fig. 3. Hardware design of mobile 3D printing platform.

Fig. 4. Adjustable length robotic arm assembly exploded view.

on the bottom plate to ensure the whole robotic arm rotated simultaneously.

The robot arm assembly consisted of two actuated DoFs. The proximal DoF is a revolute joint whose rotation axis is perpendicular to the top surface. The distal DoF is a prismatic joint that is orthogonal to the proximal DoF. Each joint is driven by a Robotis Dynamixel XM430-W350 smart servo motor. Each servo motor integrated a motor controller, network communication and wheel encoder. The robot arm's main structure has three separate ABS 3D printed parts, linked by aluminum hole pattern beams at the top and bottom. An optional circle track cart helps reduce the horizontal vibration of the robot arm when the arm length is extended in the radial direction. The repeating 3.5-mm holes on the aluminum beam provide a flexible arm length in the radial direction. The structure of the vertical prismatic joint is designed with one 8-mm lead screw and one 6-mm round shafting support rail fixed by a pillow block flange

bearing. The printer head holder achieved vertical movement by connecting with the lead screw nut. A Hotend kit (Lerdge BP6 with 8-mm nozzle) is installed on the end of the printer head holder. To reduce sliding friction and better support the weight of the print head, an additional ball caster wheel is added to the bottom of the arm block.

B. Mechatronic and Software System

Our control system is modified from TurtleBot3's mechatronic system architecture. As we previously mentioned, our main control system consisted of OpenCR (driver board) and Jetson Nano (SBC), as shown in Figure 3. As a driver board, OpenCR is connected to four servo motors through serial communication, two for the wheels and the other two for the robot arm. OpenCR's GPIO pins are used to control the heating tube, cooling fan, and to send the signal output to the extruder's stepper motor. Due to the different operating voltage and current settings of the actuators, we controlled

the heating tube and the cooling fan through additional MOSFET breakout boards. A Thermistor (100KNTC3950) provided nozzle temperature feedback by connecting with a 4.7K Ohms pull-up resistor. Due to multiple microstepper settings, we use the TB6600 as a stepper motor driver. Two 11.1V Li-Po batteries are attached on the bottom layer, one for control circuits and the other for all actuators.

For the control software, Robot Operating System (ROS) is used to communicate with and synchronize the nodes and to handle all low-level device control. We created three additional nodes and customized messages, which worked with the original TurtleBot3 libraries. Two of the nodes are used to maintain the heating temperature of hotend and publishing the direction and speed topic for extruder control. The other node calculates the pixel errors between the desired and the captured pattern positions and sends the errors back to the main control program. On the right of Figure 2, we present our training and printing workflow.

IV. VISUAL SERVOING

A. Learning-Based Visual Servoing (LBVS)

To use visual servoing to control the 2D movement of our mobile base, we need to determine the interaction matrix $L_e \in \mathbb{R}^{2 \times 2}$ between the control input $v_\theta = [v, \omega]^T$ (i.e., linear and angular velocities) and the image pixel measurements $u = [u_x, u_y]^T$ of a target feature point such that $\dot{u} = L_e v_\theta$. To make our mobile robot system easy to use and automatically deployable, unlike the classic IBVS or 2.5D-VS methods [9, 10], *we want to avoid any intrinsic/extrinsic/hand-eye calibration of the camera, and any feature point depth estimation*. We achieved this using a machine-learning-base approach to find a dynamics model, as described below. Thus, we call our method LBVS.

Utilizing the prior knowledge that a homography exists between the ground plane and the camera plane, we model the interactive matrix as a function of the pixel location, i.e., $L_e(u) : \mathbb{R}^2 \to \mathbb{R}^{2 \times 2}$. If this matrix function is given, the control law of the mobile base is the same as in the classic IBVS: $v_\theta = -\lambda L_e^+ e$, where $L_e^+ = (L_e^T L_e)^{-1} L_e^T$, $e = u - u^*$, and u^* is the desired image location of a feature point. Note that in a dynamic trajectory following case, e is the optical flow of the feature point from the current to the previous frame. Given M tracked feature points, $e = [e_1^T, \cdots, e_M^T]^T \in \mathbb{R}^{2M \times 1}$ is the stacked error vector (i.e., the optical flow), and $L_e = [L_e(u_1)^T, \cdots, L_e(u_M)^T]^T \in \mathbb{R}^{2M \times 2}$ is the stacked image Jacobian evaluated at each feature point, and the resulting control v_θ jointly regulates each feature point's error vector via the least squares principle.

The interaction matrix function $L_e(u)$ can be modeled as a simple multi-layer perceptron (MLP), e.g., a ReLU MLP (2-64-64-64-4) with three 64-dimensional hidden layers. The challenge is determining how to estimate this MLP automatically. Fortunately, we can take advantage of our projector and the ground plane. During the automatic system calibration stage, we can project a set of N random colored dots on the ground such that these points cover enough area on the image plane. We can control the mobile base with a sequence

of T frames of random velocity commands $[v_\theta^1, \cdots, v_\theta^T]$ while recording the color dots' image measurements at each frame as $[U^0, U^1, \cdots, U^T]$, where $U^t = [u_1^t, \cdots, u_N^t]$. Denoting the i-th point's optical flow vector at frame t as $f_i^t = u_i^{t+1} - u_i^t$, we can train the MLP by minimizing the L2 loss over this dataset as follows:

$$\min_{L_e} \frac{1}{T} \sum_t \frac{1}{N} \sum_i ||L_e(u_i^t) v_\theta^t - f_i^t||^2. \qquad (1)$$

Our LBVS method has a desirable property in that it is generally applicable to a wide range of camera lenses (perspective or fisheye) and can be calibrated automatically.

B. Additional Control Constraints

By adding additional control constraints, our mobile platform can drive more smoothly and avoid speed overshoot. As presented above, the LBVS speed control equation is $v_\theta = -\lambda L_e^+ e$. We divide λ into two parts based on the distance to the target point pattern. As the robot approaches to the pattern, we lower λ to prevent speed overshoots. For large e, we adopt a larger λ to reduce the error e quickly. As shown in our experiments, LBVS sometimes can still lose track of the point pattern during rotation. Thus, we included an additional mode to optimize the rotation. In this mode, we can estimate L_e as long as we can obtain position information of any three points from the point pattern. We also constrain the maximum linear velocity to 0.025 m/s and maximum angular velocity to 0.05 rad/s.

V. EVALUATION

We design three experiments to verify the feasibility of our mobile 3D printing system. We first introduce our experimental environment and hardware setup. In the first experiment, we test our LBVS control in the simulation and real-world environment. We use our printing system to print a hand-sized model, which could also be printed by a traditional desktop 3D printer. The purpose of the second experiment is to explore and optimize our printing system by comparing with basic 3D printing models. We verified the large-scale collaborative 3D printing in the final experiment. For each experiment, we discuss all printing deviations and failure cases that occurred.

A. Experiment Setting

Simulation environment. In the simulation environment, our mobile platform will drove on a plane object on the ground. This object was used to simulate the pattern's projection by updating the surface image with moving points. We subscribe odometry topic and joint space topic for recording the position and speed of the mobile platform. We used Moveit to plan the trajectory and send the execution data to the robotic arm.

Real-World Environment. We mounted a projector on the ceiling and adjusted the keystone correction. To ensure that the printing surface was the flat and level, we used three $0.61\ m \times 1.22\ m$ medium-density fiberboard (MDF) to form a $1.82\ m \times 2.44\ m$ flat surface. We laid roller paper
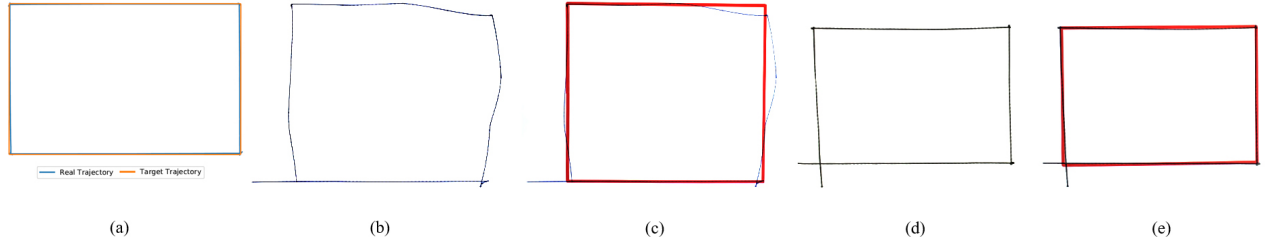
Fig. 5. The trajectory following experiment results: (a) Simulation results; (b) (c) Real-world results of LBVS without control constraints; (d) (e) Real-world results of LBVS control with control constraints.
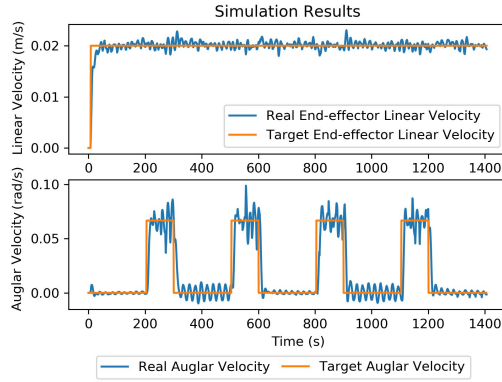

Fig. 6. Simulation results of real velocity vs. target velocity.

on the MDF to replace the yellow wood color. The white background allowed the camera to capture the feature points more easily. For easy peeling off of the printing model, we set the Buildtak printing board on the paper surface in single robot printing experiment.

Data Collection and Network Training. We used the well-known KLT algorithm [19] to calculate the optical flow $[\dot{u}_x, \dot{u}_y]^T$ between image frames. In our real world experiment, we projected 10,000 little triangles on the printing surface with a uniform distribution. We set a minimum requirement of 20 feature points that need to be capture. The real linear and angular velocity $[v, \omega]^T$ can be read directly from the joint space ROS topic. As explained in Section IV-A, we let the robot move straight forward and backward with no angular velocity while recording v and $[\dot{u}_x, \dot{u}_y]^T$ for different image points. We then let the robot spin without linear velocity while recording ω and $[\dot{u}_x, \dot{u}_y]^T$ for different image points. Finally, we combined the two datasets to train the model prediction $\boldsymbol{L}_e$ via equation (1). In the training process, we used 10^5 feature points and 3000 frames, as T in equation (1). The data collection and network training were completed in 30 minutes with $1e^{-4}$ learning rate by using an Nvidia TitanV graphics card.

B. Trajectory and Control Accuracy

In this experiment, we designed a rectangular motion trajectory. In Gazebo, the results showed that the LBVS could control the mobile platform to complete the trajectory well. In Figure 6, both the linear and angular velocities of the mobile platform exhibited almost the same shapes as

TABLE I
ATE FOR REAL-WORLD EXPERIMENT

ATE (mm)	RMS	Median	Max
Without Constraints	19.28	10.13	57.67
With Constraints	**1.67**	**3.12**	**7.30**

the pattern's speed. Meanwhile, as expected, there was a small delay between them. The reason was that the mobile platform needed to observe and process the point pattern before reacting.

For the real-world environment, we first tested the LBVS without additional control constraints. At the beginning of the trajectory, the mobile platform presented the same performance as that in simulation. After the pattern rotated, there were several moments when the robot maintained a constant angular velocity due to lost points. Therefore, deviations occurred in the upper right corner, as shown in Figure 5 (b). Furthermore, we found that the mobile platform moved back during its rotation process. These two factors could strongly impact the printing quality and cause printing failure during the process.

To optimize the LBVS control, we added the control constraints described in Section IV-B. As shown in Figure 5 (d) and (e), we only achieved a slight adjustment at the top edge. From Table I, the absolute trajectory error (ATE) [20] also indicated that the error of the LBVS with constraints was much smaller than that of the LBVS without constraints.

C. Single Robot Printing

We used a single mobile platform to print a cuboid model. Based on the previous experiments, we learned that the rectangle corners were the most critical positions affecting the printing results. To better handle four corners, we designed two different corner printing methods. The first method fixed the proximal joint position to $-\pi/2$ and completely relied on extruder control. The other method used the robotic arm to compensate for the rotation of the robot chassis. It kept the end effector stationary when the mobile platform turned. Additionally, we propose a new printing method that could optimize our printing results. In this method, the mobile platform prints front and back three times on every edge once before it turns to the next edge. The models shown in Figure 7 were not completed all at once. Based on the previous
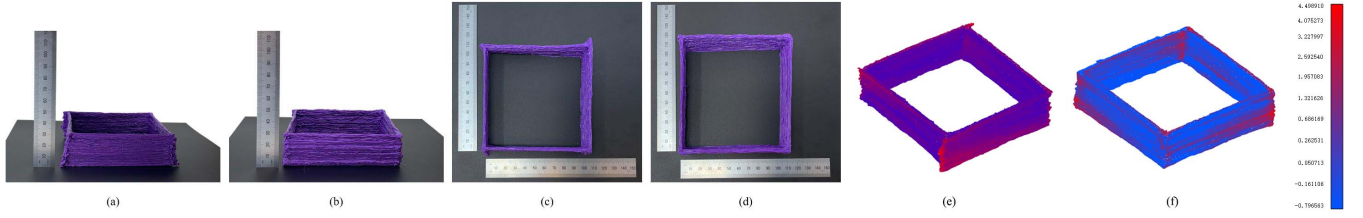
Fig. 7. Printing results of single robot printing: (a),(c),(e) fixed-arm printing; (b),(d),(f) rotation compensation printing; (e),(f) comparison of scanned point cloud and ground truth.
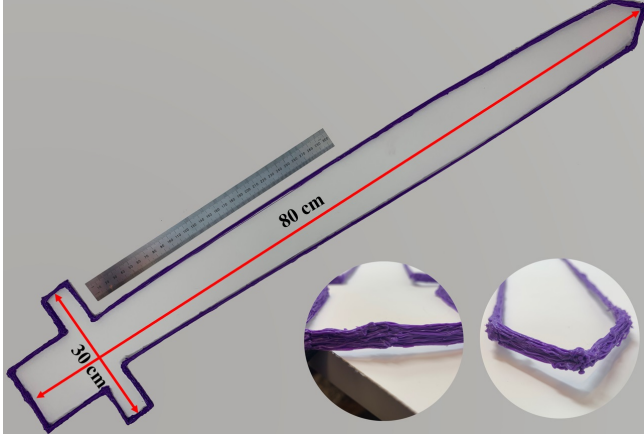

Fig. 8. Large-scale collaborative printing results. Bottom circle images show the connection points of two trajectories.

TABLE II
WALL THICKNESS FOR REAL-WORLD EXPERIMENT

Wall Thickness (mm)	GT	Mean	Maximum	Minimum	St.Dev
Without Constraints	2.00	3.72	4.92	**2.33**	0.40
With Constraints		**3.02**	**4.17**	2.74	**0.24**

the point of the sword. Avoiding collisions and control interference are two other issues that require concern. In this experiment, we manually designed the printing trajectory to avoid the collision. We also change the color of the point pattern to avoid mutual control interference.

VI. CONCLUSIONS

In this paper, we proposed a projector-guided non-holonomic mobile 3D printing system. Compared with a traditional 3D printer, we overcame the 3D printing size limitation due to the build plate and gantry structure in conventional 3D printing. Compared with prior works, our method does not require any manual calibration or any world coordinate for the mobile robot's pose feedback. From the experimental results, our LBVS worked effectively on the non-holonomic mobile platform.

Limitations and Discussions. Although our mobile 3D printing system has a great potential for future 3D printing and robotic construction, our current system prototype is still far from producing a high quality model with satisfactory surface smoothness, accurate size dimension and shape diversity. Our deviation analysis showed that the material overfill and layer shifting occurred on almost every layer. Currently, we only printed a 3D model's outer surface. When more degrees of freedom are added to the robot arm, our system should be able to print more diverse shapes with even internal structures. Also our current projector's illumination level is a limitation if we want to use this system in an outdoor environment, where laser projectors would be more helpful.

Future Work. In the future, we will optimize the LBVS control for more DOFs, update our hardware design, and print 3D models with competing qualities to those of desktop 3D printers. We believe that our system could also use different materials and operate in different application scenarios ranging from desktop-scale 3D FDM printing to room-scale 3D construction printing with multiple projectors.

printing results, we adjusted the extruder control and replaced the batteries for every 1-cm height. We measured the printed wall thickness in Table II in comparison with the designed thickness as ground truth (GT).

As Figure 7 shows, the model printed by the fixed robot arm method significantly over-printed at the four corners. When the mobile platform completed the turn, the end effector could not always reach the previous printing endpoint. We found that the model surface printed by the rotation compensation method was cleaner and smoother due to less over-stacking at the model's four corners.

D. Large-Scale Collaborative Printing

The purpose of the final experiment was to test our mobile 3D printing system, which could quickly set up multiple printing platforms and complete large-scale printing. Here, we used two mobile platforms to complete this experiment. We designed an asymmetric contour, a sword, which is shown in Figure 8. This sword model had a total length of 0.80 m and a width of 0.30 m, and multiple corners that need to be turned. Each mobile platform needed to print half of the entire shape at the same time. The printing model had stacked layers at the sword's point and grip, as shown in the Figure 8 circle. Furthermore, we printed directly on the laid roller paper instead of the traditional 3D printing surface.

During the experiment, we found that most of the mobile printing failures also occured with the desktop 3D printer. The most common failure is model shrinkage. Our model separated from the printing surface several times. To overcome this problem, we overprinted at every corner except

REFERENCES

[1] L. G. Marques, R. A. Williams, and W. Zhou, "A mobile 3d printer for cooperative 3d printing," in *Proceeding of the 28th International Solid Freeform Fabrication Symposium*, 2017, pp. 1645–1660. 2

[2] L. Poudel, W. Zhou, and Z. Sha, "A generative approach for scheduling multi-robot cooperative three-dimensional printing," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 6, 2020. 2

[3] J. Currence, R. Morales-Ortega, J. Steck, and W. Zhou, "A floor power module for cooperative 3d printing." 2

[4] X. Zhang, M. Li, J. H. Lim, Y. Weng, Y. W. D. Tay, H. Pham, and Q.-C. Pham, "Large-scale 3d printing by a team of mobile robots," *Automation in Construction*, vol. 95, pp. 98–106, 2018. 2

[5] M. E. Tiryaki, X. Zhang, and Q.-C. Pham, "Printing-while-moving: a new paradigm for large-scale robotic 3d printing," *arXiv preprint arXiv:1809.07940*, 2018. 2

[6] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3400–3407. 2

[7] X. Zhang and Q.-C. Pham, "Planning coordinated motions for tethered planar mobile robots," *Robotics and Autonomous Systems*, vol. 118, pp. 189–203, 2019. 2

[8] J. Lv, H. Shen, and J. Fu, "Large-scale 3d printing technology based on the visual stitching method," *Rapid Prototyping Journal*, 2019. 2

[9] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006. 2, 4

[10] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999. 2, 4

[11] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 404–417, 1987. 2

[12] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[13] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1624–1629.

[14] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30. 2

[15] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation (for robots)," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, pp. 691–700, 1989. 2

[16] J. Gao, A. A. Proctor, Y. Shi, and C. Bradley, "Hierarchical model predictive image-based visual servoing of underwater vehicles with adaptive neural network dynamic control," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2323–2334, 2015.

[17] D. Zheng, H. Wang, J. Wang, S. Chen, W. Chen, and X. Liang, "Image-based visual servoing of a quadrotor using virtual camera approach," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 972–982, 2016. 2

[18] A. X. Lee, S. Levine, and P. Abbeel, "Learning visual servoing with deep features and fitted q-iteration," *arXiv preprint arXiv:1703.11000*, 2017. 2

[19] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981. 5

[20] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251. 5