

Projector-Guided Non-Holonomic Mobile 3D Printing

Xuchu Xu¹, Ziteng Wang², 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 $80\text{ cm} \times 30\text{ cm}$ 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 house-scale objects. 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.

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

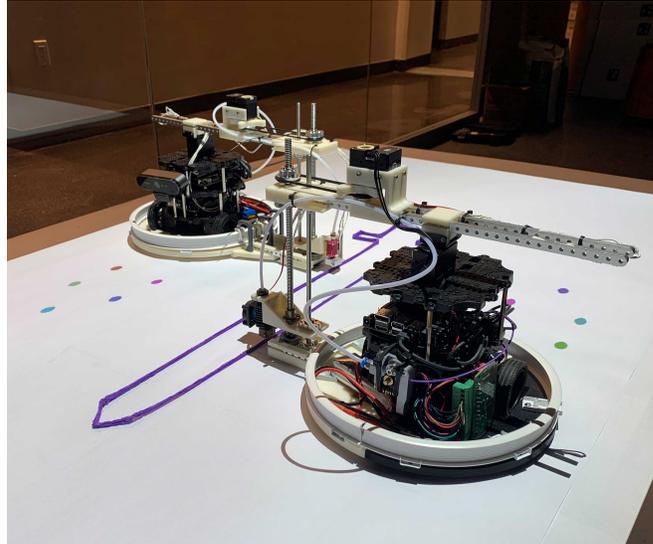


Fig. 1. Collaborative printing of two mobile 3D printers in our system.

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.

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. [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. [2] 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. Their algorithm could obtain localized feedback and adjust the motion error from the AprilTags [3] on the ground.

Holonomic vs. non-holonomic mobile platform. The main differences between the above mobile 3D printing

¹ New York University, Brooklyn, NY 11201, USA {xuchuxu, cfeng}@nyu.edu

² tonywangziteng@163.com

[†] Chen Feng is the corresponding author.

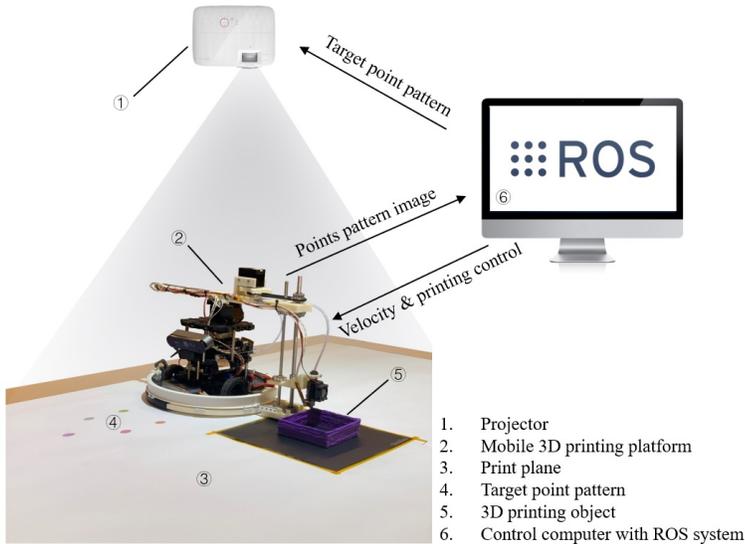


Fig. 2. System settings and operation workflow. Left: projector-guided mobile 3D printing diagram. Right: training pipeline (green blocks) & printing pipeline (blue blocks).

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. 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. Position-based visual servoing (PBVS) [4, 5] 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) [6–8] 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.

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.

Our mobile platform was developed on Robotis TurtleBot3 burger. A 360° rotating robot arm was installed on the top layer. 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.

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). 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. 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. 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

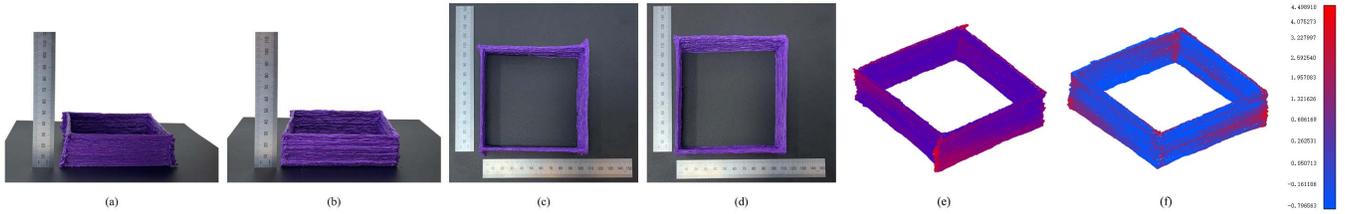


Fig. 3. Printing results of single robot printing: (a),(c),(e) fixed-arm printing; (b),(d),(f) rotation compensation printing; (e),(f) comparison of scanned point cloud and ground truth.

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., $L_e(\mathbf{u}) : \mathbb{R}^2 \rightarrow \mathbb{R}^{2 \times 2}$. If this matrix function is given, the control law of the mobile base is the same as in the classic IBVS: $\mathbf{v}_\theta = -\lambda L_e^+ \mathbf{e}$, where $L_e^+ = (L_e^T L_e)^{-1} L_e^T$, $\mathbf{e} = \mathbf{u} - \mathbf{u}^*$, and \mathbf{u}^* is the desired image location of a feature point. Note that in a dynamic trajectory following case, \mathbf{e} is the optical flow of the feature point from the current to the previous frame. Given M tracked feature points, $\mathbf{e} = [e_1^T, \dots, e_M^T]^T \in \mathbb{R}^{2M \times 1}$ is the stacked error vector (i.e., the optical flow), and $L_e = [L_e(\mathbf{u}_1)^T, \dots, L_e(\mathbf{u}_M)^T]^T \in \mathbb{R}^{2M \times 2}$ is the stacked image Jacobian evaluated at each feature point, and the resulting control \mathbf{v}_θ jointly regulates each feature point's error vector via the least squares principle.

The interaction matrix function $L_e(\mathbf{u})$ 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 $[v_\theta^1, \dots, v_\theta^T]$ while recording the color dots' image measurements at each frame as $[U^0, U^1, \dots, U^T]$, where $U^t = [u_1^t, \dots, u_N^t]$. Denoting the i -th point's optical flow vector at frame t as $\mathbf{f}_i^t = \mathbf{u}_i^{t+1} - \mathbf{u}_i^t$, we can train the MLP by minimizing the L2 loss over this dataset as follows:

$$\min_{L_e} \frac{1}{T} \sum_t \frac{1}{N} \sum_i \|L_e(\mathbf{u}_i^t) \mathbf{v}_\theta^t - \mathbf{f}_i^t\|^2. \quad (1)$$

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.

V. EXPERIMENT

We design three experiments to verify the feasibility of our mobile 3D printing system. 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.

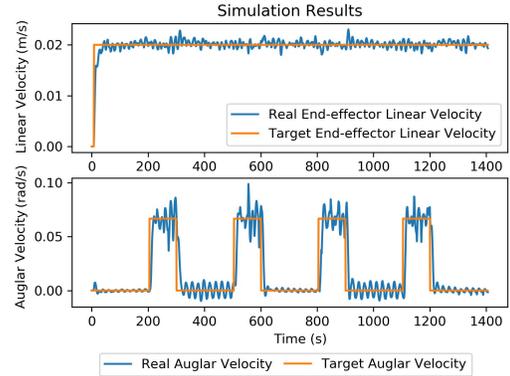


Fig. 4. Simulation results of real velocity vs. target velocity.

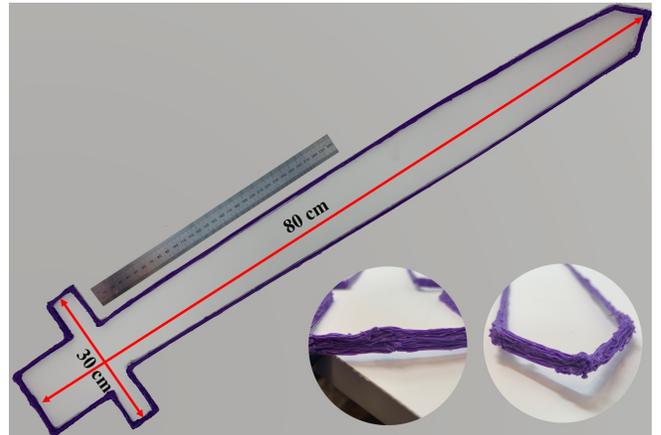


Fig. 5. Large-scale collaborative printing results. Bottom circle images show the connection points of two trajectories.

A. 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 4, both the linear and angular velocities of the mobile platform exhibited almost the same shapes as 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.

B. 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

TABLE I
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

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 3 were not completed all at once. Based on the previous printing results, we adjusted the extruder control and replaced the batteries for every 1-cm height. We measured the printed wall thickness in Table I in comparison with the designed thickness as ground truth (GT).

As Figure 3 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.

C. 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 5. 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 5 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 occurred 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 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.

ACKNOWLEDGMENTS

This research is supported by NSF CPS program under CMMI-1932187.

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. 1, 2
- [2] 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. 1
- [3] E. Olson, “Apriltag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3400–3407. 1
- [4] 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. 2
- [5] 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
- [6] 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
- [7] 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.
- [8] 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
- [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
- [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