2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3503-7770-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IROS58592.2024.10802489

Architectural-Scale Artistic Brush Painting with a Hybrid Cable Robot

Gerry Chen[1], Tristan Al-Haddad[2], Frank Dellaert[1], and Seth Hutchinson[1]

***Abstract*— Robot art presents an opportunity to both showcase and advance state-of-the-art robotics through the challenging task of creating art. Creating large-scale artworks in particular engages the public in a way that small-scale works cannot, and the distinct qualities of brush strokes contribute to an organic and human-like quality. Combining the large scale of murals with the strokes of the brush medium presents an especially impactful result, but also introduces unique challenges in maintaining precise, dextrous motion control of the brush across such a large workspace. In this work, we present the first robot to our knowledge that can paint architectural-scale murals with a brush. We create a hybrid robot consisting of a cable-driven parallel robot and 4 degree of freedom (DoF) serial manipulator to paint a 27m by 3.7m mural on windows spanning 2-stories of a building. We discuss our approach to achieving both the scale and accuracy required for brush-painting a mural through a combination of novel mechanical design elements, coordinated planning and control, and on-site calibration algorithms with experimental validations.**

I. Introduction

Art, including visual arts, music, and dance, is a special medium of communication that can engage and inspire the public in a way that traditional robotics research often struggles to do, while the challenge of robot art simultaneously pushes forward the state-of-the-art in robotics. For example, the Boston Dynamics robot dance video reached 30 million viewers – likely more than any academic paper ever written. For this work, we seek to create a large-scale mural which draws-in audience members to witness live robot painting and inspire learning about robotics for years to come.

A number of prior works have explored robotic painting with brushes. The most common approach is to use painterly rendering algorithms such as [1], [2] to generate brush stroke placements which are then executed by a serial manipulator [3]. Some more recent approaches build on the traditional painterly rendering approach with learned brush models [3], optimization [4], differentiable rendering [5], [6], and semantic objective functions [7], [8] to generate higher quality renderings. Other works apply generative models to generate brush strokes directly [9], [10]. Robot-centric approaches close the loop between the physical artwork and rendering to produce better physical paintings [3], [11]. Nevertheless, both stroke generation algorithms and robot platforms are not conducive to large-scale paintings: the number of strokes generally increases proportionally to the canvas area and serial manipulators are difficult to scale to architectural sizes.

This work was supported by NSF Grant No. 2008302, the Georgia Tech Library's Artist-In-Residence program, and Formations Studio.

[1]Institute for Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA, USA, {gerry,fd27,seth}@gatech.edu
[2]Formations Studio, Atlanta, GA, USA, tristan@formations.works

Fig. 1. *Polycentric Truthes* [sic] is a 27m by 3.7m mural painted on the windows of the Price Gilbert Library in Atlanta for the Library's Artist-in-Residence program with an audience of thousands of visitors. The mural was designed by Tristan Al-Haddad and painted by our hybrid cable-driven robot over the course of 7 days across 3 weeks.

Meanwhile, architectural-scale murals have been painted by robots using other mediums. Graffiti spray-painted murals have been painted using cable robots [12]–[14], UAVs [15], [16], and gantry robots [17], but spray paint is not well-suited for indoor environments and existing works have limited capacities for varied stroke shapes, especially compared to brush paintings. Marker-based graffiti murals have been painted using (humanoid) mobile robots [18] and gantry robots [19], but markers too lack the stroke variation of brushes which give an organic element to the artwork.

In this work, we address the many challenges associated with creating an architectural-scale, brush-painting mural robot. Specifically, the flexibility of the brush tip requires considerably more precision (especially in the out-of-plane direction) and dexterity than existing large-scale graffiti painting robots, while the large size of the mural makes existing brush painting robot systems impractical. While many prior works have explored robotic painting of large-scale murals and brush paintings independently, to our knowledge, our work is the first to combine the two. Achieving this combination of scale and precision requires novel insights into mechanical design, control, and calibration.

Our novel contributions include an analysis of cable-driven parallel robot (CDPR) cable routing solutions, the design of a brush stabilizing mechanism, and a joint- and task-space calibration algorithm for registering to absolute on-site canvas locations. We also demonstrate our robot painting a large-scale mural and share a variety of practical information for robot mural painting.

APPROACH

We discuss our system in 3 parts: robot hardware design, software design, calibration/operation. In *Sec. II: Robot Hardware Design*, we discuss how the mechanical and electrical design of the robot was chosen to meet the requirements of the painting task. In *Sec. III: Software Design*, we discuss how we control and coordinate the robot to follow a painting trajectory. In *Sec. IV: Calibration, Artwork Design, and Operation*, we discuss how the robot is calibrated, painting trajectories are generated, and the robot is operated.

II. ROBOT HARDWARE DESIGN

Our choice of robot hardware design was motivated by our mural's requirements. The mural is large, at 27m x 3.7m, planar, and to be painted with a paintbrush in multiple colors. Furthermore, the mural must be painted without interrupting the library's normal operations, which means that the robot must be able to *safely* paint in a public space.

We chose a hybrid CDPR + serial manipulator design for our painting robot to combine the scalability of CDPRs with the adaptability and dexterity of serial manipulators. CDPRs, which consist of several winch-actuated cables that pull on a moving platform to control its motion, are ideal for large-scale painting applications since they are characterized by their scalability and high payload capacities. However, because cables have no ability to resist compression (or bending), the moving platform often suffers from relatively lower stiffness/rigidity. Fortunately, for the application of painting on a planar surface, contact forces are low and the painting surface can aid in stabilizing the moving platform, together with appropriate internal tensioning of the cables. Comparing to some alternative large-scale painting platforms studied, cable robots tend to have taller workspaces than mobile robots [18]; be cheaper and more portable than gantry systems [19]; and posses greater payloads, easier control, non-battery-limited mission durations, and superior safety than UAVs [16]. Serial manipulators, on the other hand, are often used in artistic painting robots for their ability to carefully control the brush or marker [3], [4], [20]. Although they do not scale to large sizes alone, they have been paired with cable robots in the past to enable dexterity across large workspaces [21].

A. CDPR Design

Our CDPR was designed around the mural's setting, but is flexible and extensible to other settings as well. We chose a 4-cable planar CDPR 5.8m wide and 3.7m tall (Fig. 2) and paint in sections, relocating the robot between each section. 4 pulleys are located at the 4 corners of the robot, rigidly mounted to the building's structure with a custom tetrahedral wooden clamp which clamps onto mullions without damaging them (Fig. 4). The pulleys are located at 4 corners of a rectangle, and the mounting locations on the moving platform are also at the 4 corners of a rectangle (Fig. 3). Although this results in many singularities which primarily manifest as an inability to control the orientation of the moving platform, we find that the configuration is more passively stable and therefore easier to manage than one which requires feedback on the moving base orientation.

Fig. 2. Our 4-cable planar CDPR is 5.8m wide by 3.7m tall.

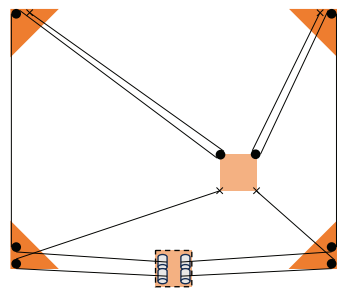

Fig. 3. The 4-cable planar CDPR uses platform-mounted pulleys on the top two cables to achieve a 2:1 mechanical force advantage. We perform an analysis showing we do not lose estimation accuracy with this approach compared to halving the winch radius.

To support the heavy wooden platform with serial manipulator, we seek to achieve more cable tension while minimizing cable length estimation error. We perform a theoretical analysis to compare how the accuracy changes if we add a pulley, half the winch radius, or add a motor gear ratio. Surprisingly, we have not found such prior analysis in the CDPR literature. Making the assumption that the majority of the cable length estimation error comes from variations in the effective winch radius, we model how the cable length estimate changes with changes in diameter. The cable length for the gearbox can be modeled as $\frac{1}{2}\theta d$, so the derivative with respect to winch diameter is $\frac{1}{2}\theta$, where θ is the output shaft angle and d is the winch diameter. If we instead half the winch radius, we have the derivative is still $\frac{1}{2}\theta$, but now θ must be twice as large to elicit the same change robot motion so accuracy halves. For the pulley, we have the same expression $\frac{1}{2}\theta$ and accuracy halves for the same reason. We experimentally validate this conclusion in Sec. V-A.

For our setup, we opted for a pulley configuration because the winch diameter was already very small at 10mm and motor gearing would increase cost and complexity while also introducing the potential for decreased backdrivability.

Fig. 4. The serial manipulator is mounted on the CDPR's moving platform (Left) controlled by 4 cables routed through pulleys rigidly mounted to the window mullions with wooden clamps (Right).

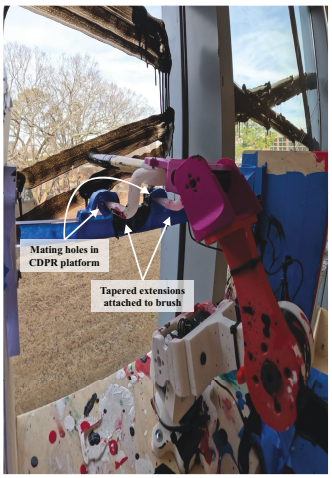

Fig. 5. The brush stabilizer's self-centering mates are engaged when the brush is in the painting position.

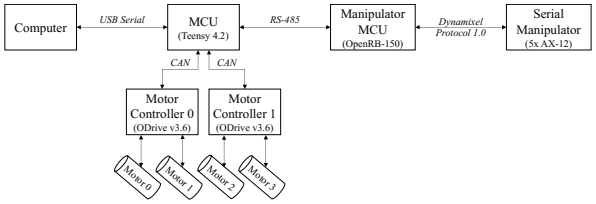

Fig. 6. The robot system components communicate with each other over robust interfaces to manage both the CDPR and serial manipulator.

Specifically, we employ a 2:1 pulley system for the top two cables of the cable robot. Instead of mounting a top cable directly to the moving platform, an additional pulley is instead mounted to the moving platform and the cable is routed through this pulley then affixed to the tetrahedral clamp near the fixed-based-pulley as shown in Fig. 3.

To drive the winches, we use BLDC motors with anti-cogging compensation, which enables smooth, precise control and accurate torque feedback.

B. Serial Manipulator Design

The serial manipulator is primarily designed around the choice of brush painting as the artistic medium. The brush is a versatile tool that can be used to create a wide variety of strokes, textures, and effects, while also being a tool that is familiar which we hope will make the mural more relatable to the public. Finally, as opposed to a more "industrial" design (e.g. with linear actuators and mechanical linkages), the serial manipulator is designed to be able to manipulate the brush in a way that is similar to how a human artist would manipulate a brush, thereby contributing to the performative aspect of the robot's artistic expression.

To achieve the requisite dexterity to paint, we designed a 4-DoF serial manipulator with a 0.38m reach and measured 150g payload capacity (at the tip) to hold the 110g brush. The manipulator is designed to be lightweight and affordable, leading to the choice to use AX-12 Dynamixel servos. Although for this mural we keep the brush perpendicular to the canvas when painting, the 4-DoF serial manipulator complements the CDPR's 2 translational degrees of freedom theoretically enabling SE(3) brush motions. We use a standard 2-DoF shoulder, 1-DoF elbow, 1-DoF wrist design for the manipulator due to its effectiveness and ubiquity as shown in Fig 5. The second degree of freedom in the shoulder uses two synchronized servos to improve the payload capability of the manipulator.

Due to the significant inaccuracy of the AX-12 servos, we designed a stabilizing stand to keep the manipulator rigid while painting (Fig. 5). During operation, we found that AX-12 servos experience error due to both mechanical backlash and controller degradation after prolonged continuous operation. As a result, brush positioning suffers from poor accuracy and, more importantly, deflection, which results in "cutting corners". To mitigate this, we designed a stabilizing mechanism whereby a pair of conical extensions on the brush/manipulator end-effector creates tapered fits with circular hole receptacles on the moving platform. The mechanism is designed such that it is engaged (locks the pose of the brush relative to the platform) when the brush is in the painting position (in contact with the canvas) and can be unengaged by simply pulling the brush back off the canvas with the manipulator. The mate is designed with a tapered, self-centering fit to be robust to positioning errors as the serial manipulator pushes the brush forward to engage the fit. The mechanism, engaged, is shown in Fig. 5.

Although the stabilizing stand means that the arm cannot move while the painting, we find that this is necessary to attain sufficient accuracy. We will further discuss in Sec. III the tradeoff between producing coordinated motions (i.e. CDPR and arm move together while drawing) vs fixing the arm in place while the CDPR moves to paint strokes.

III. Software Design

To coordinate our hybrid robot, we implement low-level controls for the CDPR and arm then tie them together with higher level coordination logic. For the purposes of this section, it is assumed that a time-stamped painting trajectory is already provided, using the procedures in Sec. IV.

Decoupled vs coordinated motion: We largely rely on the CDPR to follow the painting trajectory and reserve the serial manipulator just for moving the brush to make/break contact with the canvas and handle other brush management tasks (e.g. dipping to refill paint). The primary reason is the aforementioned inaccuracy of the hobby-grade serial manipulator which, when not held in place using our stabilizing mechanism, deviates far too much from the desired trajectory to outweigh the benefits of using it for trajectory following. Furthermore, as the CDPR can smoothly reach the entire canvas and we do not explicitly model how orientation affects brush strokes, there is limited added benefit to moving the manipulator together with the CDPR while painting aside from the performative aspect.

A. CDPR Control

We implement an iLQG tracking controller based on [13], which receives a desired trajectory and computes tensions to apply to the cables to track the trajectory given cable length and velocity measurements. The controller consists of an offline stage in which we pre-compute time-varying linear control gains and state estimation gains, and an online stage in which we execute the time-varying, locally-optimal linear state estimator and linear controller.

Offline, our algorithm takes as input a desired painting trajectory and first solves for the optimal nominal trajectory. The nominal trajectory is computed as the solution to a QP with quadratic state error and control cost objectives, a system model (equality constraints), and control limits (inequality constraints). The QP has a banded structure and is solved in linear time using iLQG, implemented using GTSAM [22]. Then, the QP problem is linearized around the nominal trajectory and the locally optimal, time-varying linear control gains and Kalman estimation gains are computed. In GTSAM, the time-varying LQR gains are computed automatically during the last iLQG iteration [23] so they can be directly retrieved, while the Kalman estimation gains are computed using a backward-forward algorithm.

The resulting time-varying gains are uploaded to the microcontroller (MCU) then executed in real-time as:

$$\delta\hat{x}_k = {}^{x}K_k \delta\hat{x}_{k-1} + {}^{z}K_k z_k + k_k \tag{1}$$

$$u_k = K_k \delta\hat{x}_k + u_k^* \tag{2}$$

where $\delta\hat{x}_k$ is the estimated deviation of the state from nominal; z_k is the measurement (cable lengths and velocities); u_k is the commanded cable tensions, ${}^{x}K_k$, ${}^{z}K_k$, and k_k are precomputed matrices/vectors for the time-varying linear state estimator, and K_k and u_k^* are precomputed feedback control gains and feedforward controls, respectively.

It may come as a surprise that the optimal state estimator does not appear to follow the standard form of a Kalman filter (while the control gains take the form of a typical time-varying LQR controller). This is because knowing the nominal trajectory in advance allows us to simplify the online computation by pre-multiplying matrices. Although full details of the derivation are provided in [13], the intuition is that the covariances depend only on the system model (which is fixed once we linearize around the nominal trajectory) and not on the real-time measurements so we can pre-compute and combine the predict and update steps into a single, affine function of the previous state estimate and current measurement per timestep.

Given the simplicity of the expressions in Eqs. (1) and (2), the control loop runs at 1 kHz on the MCU.

Occasionally, we must manually control the robot live (i.e. with a joystick) such as for setup and testing. In these cases, the trajectory is not known in advance so the dual-space PID controller from [24] is implemented. Although the accuracy and dynamic performance of the iLQG controller is superior, this manual mode is used primarily for visual alignment and testing so its reduced performance is acceptable.

B. Serial Manipulator (Arm) Control

The primary arm functions we seek to implement are:

- rest/off
- painting (touching the canvas)
- refilling (dip the brush into the paint).

Accordingly, we can define a set of only 3 configurations and the 3 (bidirectional) trajectories transitioning between them, as shown in Fig. 7, to cover all the required arm functions. The "prep" state is helpful since represents a configuration where it is relatively easy to find collision-free trajectories to any other configuration the arm needs to reach. Since the prep state does not make contact with the canvas, we also use it during travel motions.

To program the arm configurations, we take two approaches: manual "training" and analytical inverse kinematics. We find that a simple "training" procedure is often easier given that there are only a few configurations required (including collision-free intermediate waypoints) and the servo inaccuracy makes it such that task-space target positions would need to be tuned anyway. Specifically, the training procedure consists of disabling the servos and manually moving the arm to the desired configuration, then recording the joint angles which are later used as target configurations. We use this training procedure for the rest configuration and refilling (paint dipping) trajectory. However, we use the analytical inverse kinematics for the "prep" configuration and "start/stop painting" trajectories because programming the arm to successfully engage the stabilizing mechanism is easier when adjustments are made in task space (e.g. move 2cm higher). Furthermore, described in Sec. II-B, the servo accuracy degrades with prolonged operation so we compensate by commanding progressively higher (1cm/hr) target "paint" positions in task space as the painting progresses.

C. Coordination and State Management

The high-level coordination and state management between the CDPR and Arm is shown in Fig. 8, and we now also discuss a few non-obvious details. For the "*Paintbrush dry*" condition, we find that distance traveled along the canvas is a good heuristic and we automatically re-dip the brush for every 0.5m the brush has traveled along the canvas. We also pause slightly (1s and 2s, respectively) just after

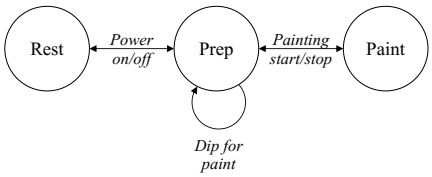

Fig. 7. Arm state machine consists of 3 states / configurations: rest (power-off configuration), prep ("central" configuration not yet touching the canvas but ready to paint), and painting (touching the canvas).

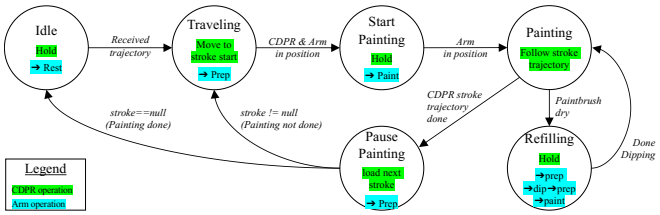

Fig. 8. The CDPR + arm pair is coordinated by this state machine to paint.

the transitions from "Traveling"→"Start Painting" and from "Start Painting"→"Painting" to allow the CDPR's motion to arrest, since arm motions often cause slight out-of-plane oscillations in the CDPR platform. We load each paint color as a separate trajectory so that the robot returns to the "Idle" state after each color, enabling us to switch paint colors.

For this work, we chose manual paint color switching over automation for three key reasons. First, cleaning brushes sufficiently to avoid color contamination is impractical – even humans require 10-15 minutes, so we simply purchase a separate brush for each color. Second, the drying time of the paint is long and variable, so we intentionally plan the trajectories to minimize color changes. Finally, the human operators can switch the paint color in less than a minute so the additional complexity of automatic color switching is difficult to justify. Future works which use different brushes and paints may find automatic paint switching more practical.

D. Safety Controls

In addition to standard safety measures such as E-Stop buttons, actuator limits, and safety interlocks, we implement a few additional safety features.

The serial manipulator's hobby-grade servos frequently overheat causing accuracy degradation and, if extreme, automatic shut-down of the servo which may affect the painting or drop objects onto passersby. In response, we continously monitor servo temperatures and automatically pause painting when the temperature reaches a threshold (65°C), wait for the temperature to cool to a safe level, then resume painting.

The CDPR's iLQR controller is also very reliable, but we nonetheless implement safety stops in the event of anomalies to prevent negatively affecting the painting and objects falling onto passersby. In these cases, a soft limit enacts when the CDPR is 10cm from its commanded position and a hard limit enacts at 20cm. The soft limit pauses the trajectory but holds the current position while the hard limit immediately stops the trajectory and gently lets the platform down by switching to a gravity compensation mode.

IV. Calibration, Artwork Design, and Operation

In this section, we discuss the procedure necessary to operate the robot on-site. Because the robot is designed to paint 5.8m at a time before needing to be repositioned, calibration is a critical step to ensure that adjacent sections of the mural are painted seamlessly. We also discuss the painting trajectory generation process and robot operation.

A. Calibration

A number of both proprioceptive and exteroceptive methods for calibration have been studied in the CDPR literature. Proprioceptive methods leverage the over-actuated quality of CDPRs together with kinematic and dynamic constraints to derive calibration parameters. Meanwhile, exteroceptive methods leverage external sensors to measure the robot's configuration which often beget more accurate calibration parameters. In this work, we use a combination of both methods to calibrate the robot, in particular leveraging the architecture of the site itself to provide exteroceptive information.

Our novel approach is to combine proprioceptive calibration for control parameters with exteroceptive calibration for task-space corrections. Specifically, our proprioceptive calibration involves measuring motor angles during an exploratory motion phase (which is relatively standard in the literature). Meanwhile, our exteroceptive calibration leverages *operator* feedback to manually position the robot at known locations on the canvas, based on the site's architecture, to provide absolute task space calibration (for which we apply a novel CDPR task-space warping algorithm).

1) Proprioceptive calibration: We perform the proprioceptive calibration in two stages. In the first stage, the operator manually moves the robot to obtain a rough initialization on parameters with which the robot can then use to move itself to a broader set of configurations in the second stage. This two-stage approach improves safety, since the robot is not required to control itself without a good parameter initialization, while also being quick, easy, and accurate. Compared to auto-calibration approaches which initialize using small perturbations around the power-on configuration of the robot [25], we find that our approach is more robust to the starting configuration of the robot since it is often powered on very close to the bottom edge of the workspace (due to the large height of the workspace) which results in potentially dangerous conditions for robot actuation. Other than the procedure used to collect data, the optimization problem used to solve for the calibration parameters is the same for both stages and standard in the literature [13], [26].

During the first proprioceptive calibration stage, the robot is placed in a compliant state by commanding an equal and relatively small tension in all cables allowing the operator to manually move the robot across various configurations reachable by the operator while recording motor angles. The operator moves the robot to a diverse set of configurations reachable by person for 3 minutes. The collected data is then used to formulate a nonlinear least squares optimization

problem to solve the geometric calibration parameters:

$$\min_{p,\boldsymbol{x}} \sum_{k=1}^{n} \left\| l(\theta_k; p) - \hat{l}(A - (x_k + B)) \right\|^2 \tag{3}$$

where $l(\theta_k; p)$ computes the cable lengths given measured winch rotations $\theta_k \in \mathbb{R}^4$ at time-step k and parameterized on winch calibration parameters p; $\hat{l}(\cdot) := \|\cdot\|_2$ is the euclidean distance function predicting the cable length; $A \in \mathbb{R}^{4\times 2}$ is the locations of the "base" pulleys (mounted to the window mullions as in Fig 4-right) in the world coordinate frame; $B_i \in \mathbb{R}^{4\times 2}$ is the location of the platform mounting/pulley location for cable i in the moving platform frame; and $x_k \in \mathbb{R}^2$ is the platform position at time-step k (out of n total time-steps) in the world frame. As discussed in Sec. II-A, we chose a configuration with a relatively passively stable platform orientation at the expense of not being able to control or sense orientation well. Therefore, we assume the orientation to always be the identity rotation so platform pose can be expresed as a position, $x_k \in \mathbb{R}^2$, and can be (broadcast) summed with B for the 4 cables to compute the platform mounting/pulley locations in the world frame.

We choose a quadratic cable length vs winch rotation angle model $l(\theta; p) = p_0 + p_1\theta + p_2\theta^2$ with $p \in \mathbb{R}^{3\times 4}$ for the 4 cables to account for the increasing winch radius as the cable winds back on itself on the winch. Although we have found the true function l to be almost perfectly piecewise linear (each layer of cable winding represents a discrete transition) on data collected with a motion capture system from a smaller-scale test system, we find the quadratic model to be much easier to optimize and still reasonably accurate.

When solving (3), we initialize p as $p_2 = 0$, p_1 by the nominal winch radius, and p_0 by measuring the initial cable lengths. Both A and B are assumed to be known and fixed according to the site architecture and platform design.

During the second proprioceptive calibration stage, the winch parameters p obtained from the first stage are used to actuate the robot in a grid pattern around the workspace. The measured motor angles during this motion are then used to solve (3) again to refine the calibration parameters.

2) Exteroceptive calibration: We perform the exteroceptive calibration by positioning the robot at known locations on the canvas and computing task-space corrections needed to reach these positions accurately. This is convenient in many applications since the locations for which accuracy is most critical are often also clearly marked. For example, in a pick-and-place task, CDPR accuracy is typically most critical at the pick and place locations (which often have clear landmarks) and least critical during the move. In our application, the window mullions represent the most important locations at which accuracy is required both to prevent damage to the building and to ensure proper alignment across different sections of the mural, which is painting in sections along mullion edges. Conveniently, the mullions also make for easily accessible locations for the operator to measure.

We select 16 mullion intersection locations across the canvas (see Fig. 9). Next, we have the operator use a joystick

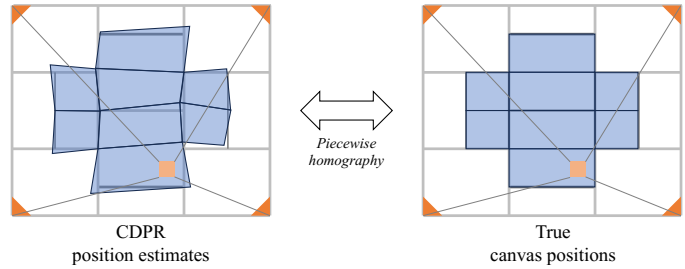

Fig. 9. We apply a piecewise homography in the task space to correct the proprioceptively calibrated cable robot's motions to align with the true canvas sections. Inaccuracy of CDPR estimates is exaggerated for effect.

to move the robot to each of these locations and record the robot's estimated task-space position. The true locations are taken from the architectural measurements and verified with a laser distance meter.

To perform calibration, we apply homography transforms[1] in the task space to each of the rectangular segments of the workspace defined by the calibrated positions. With the 16 points measured in our application, we form 8 rectangular sections of the painting portion of the canvas (Fig. 9). For each section, we compute a 3×3 homography matrix, H, mapping the true position to the estimated cable robot position using the 4 corner correspondences:

$$\begin{bmatrix} x_{cdpr} \\ y_{cdpr} \\ 1 \end{bmatrix} = \frac{1}{z} H \begin{bmatrix} x_{true} \\ y_{true} \\ 1 \end{bmatrix} \tag{4}$$

where (x_{cdpr}, y_{cdpr}) and (x_{true}, y_{true}) are the cable robot's estimated and the true canvas position, respectively; and z is a scaling factor to normalize the homogeneous vectors.

When painting, we simply apply the appropriate homography (based on which rectangular section we are in) to all the commanded trajectories to obtain corresponding trajectories in the CDPR's "erroneous" coordinate frame which is used in closed-loop control. Because adjacent rectangular sections share edges, transitions between homographies are guaranteed to be continuous. As compared to applying the inverse homography to the CDPR's internal forward kinematics state estimation inside the control loop, operating the control loop on the non-homography-corrected state estimation and applying the homography to the commanded trajectories is simpler to implement and does not exhibit reduced performance.

Our approach can generalize to arbitrary transformations (not just homographies). Triangular mesh warping, for example, may be more generalize-able to non-rectangular workspaces. However, because our canvas is naturally rectangular and we wish to reduce the number of segments where possible, we choose to use homographies.

B. Artwork Design and Trajectory Generation

The artwork designed for the mural in Fig. 1 was designed by Al-Haddad in Rhino, an architectural CAD software. It was generated parametrically to create sets of ellipses as a

[1]For details on homographies (projective transforms in computer vision) and how to compute them, see e.g. [27, Ch. 2.1.1 "Projective"].

reference to deconstructing the texts of Galileo, translating them to drawings then code before reconstructing them on the face of the library[2].

To convert the artwork to a painting trajectory, we create infill paths in Rhino from the vector graphics with 15cm stepover then retime them using [28] to ensure feasible dynamics before applying the iLQG algorithm described in Sec. III-A. In general, we find that converting artworks into toolpaths using off-the-shelf CAM software, retiming using [28], then running the trajectory with robot-specific trajectory tracking controllers is a robust, efficient workflow for painting tasks without complex brush models.

C. Operation

The overall painting process is as follows. Since, as described in Sec. II-A, our robot is narrower than our canvas and we paint in columnar sections, after each section we remove and reinstall the tetrahedral clamps supporting the pulleys in the next location. We then perform the calibration procedure described in Sec. IV-A and prepare the brush and paint for the robot to paint. Upon uploading the precomputed iLQR gains, the robot first navigates to its starting location using the "manual mode" dual-space PID controller before proceeding to follow the state machine in Fig. 8. Upon completion, the robot is place back into "manual mode" to return the platform to the operator.

V. Experimental Results

We evaluate our contributions of cable routing, brush stabilizer, and calibration algorithms separately in addition to evaluating the overall system through the completed mural.

A. Cable Routing

As discussed in Sec. II-A, we theoretically expect that cable length estimation error should not change with winch diameter for a fixed angle change θ. To experimentally validate this, we conduct a small-scale experiment with a single winch and a single cable. For various winch diameters, we measure the change in cable length for a fixed 25-rotation change in motor angle, repeated 10 times. We find that the repeatability in millimeters is consistent with our theoretical analysis, showing no clear trend with the winch diameter. The results are shown in Table I.

B. Brush Stabilizer

We evaluate the brush stabilizer's ability to maintain the brush's orientation by measuring the deflection applying a 500g mass to the tip of the brush with and without the stabilizer. We find that the stabilizer reduces the deflection significantly from 165mm to 9mm. Although 500g is a pessimistic estimate for the maximum lateral force the brush will experience, we find that the stabilizer is effective at reducing deflection.

TABLE I
Repeatability of Cable Length Estimation for Various Winch Diameters

Nominal winch diameter (mm)	Cable length change for 25 rotations		
	mean (m)	std (m)	std (%)
11.5	1.0220	0.0025	0.244
14.5	1.2214	0.0033	0.273
18.0	1.4854	0.0014	0.091
20.0	1.6163	0.0024	0.147

The repeatability, as measured by the standard deviation of the cable length measurement, is comparable in millimeters, but worse as a percentage of cable length dispensed for smaller winch diameters.

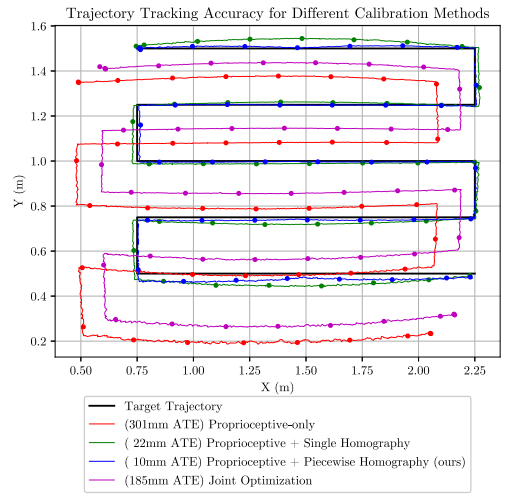

Fig. 10. Tracking accuracy is significantly better when using our homography task-space calibration, even compared to including the same exteroceptive task-space calibration data in a joint optimization routine. Dots added for visual clarity of near-overlapping trajectories.

C. Calibration

We evaluate the 2-part calibration algorithm by measuring the trajectory tracking accuracy with ablations on a smaller scale (3m×2.4m) testing CDPR. Specifically, we evaluate the tracking accuracy after applying only the proprioceptive calibration, after applying one single homography to the entire workspace, and after applying a 2×2 piecewise homography (4 sections, 9 grid points). Additionally, we also evaluate an approach using a single, joint optimization problem for both proprioceptive and exteroceptive measurements[3] but find that it performs worse than our approach and only slightly better than the proprioceptive-only calibration. Fig. 10 shows the trajectories for each of the controllers as measured using AprilTag markers on the platform and robot frame, with the piecewise homography improving average tracking error (ATE) over proprioceptive-only from 301mm to 10mm.

[2]Additional details available online at https://library.gatech.edu/AIR

[3]We formulate the optimization problem as a weighted nonlinear least squares problem with one term for the proprioceptive measurements as in (3) and the other term using the exteroceptive measurements in place of x_k in (3). Various choices of weights were tried, but the results shown in this paper use weights of 94% and 6% for the proprioceptive and exteroceptive terms, respectively, normalized by number of elements. Attempting higher weights for the exteroceptive term cause caused unsafe winch parameters.

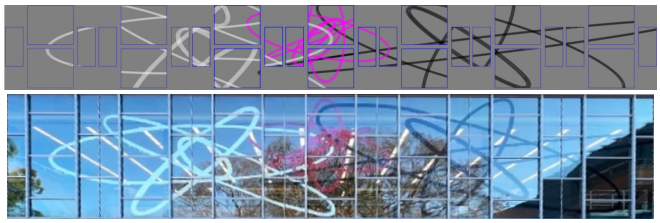

Fig. 11. Comparing the as-designed mural (top) to the completed, as-painted mural (bottom) demonstrates the proficiency of our robot.

D. Mural

We evaluate our overall robot system by painting the mural shown in Fig. 1. We find that the robot is able to paint the mural with high accuracy and precision, and that the mural is well-received by the public. Fig. 11 also compares the desired mural with the finished mural.

VI. Conclusions and Future Works

In this work, we created a robot system that can paint large-scale murals with a brush. We combined the large workspace of a CDPR with the precision and dexterity of a serial manipulator to create a robot that can achieve the precision needed for touching a brush to the canvas across a large area. We discussed a number of features that were required to attain sufficient accuracy, including novel ideas for cable routing, brush stabilization, and calibration, with experimental validation. Finally, we evaluated the robot by publicly painting a large-scale 27m by 3.3m mural with positive reception from the public. We believe that this work is a significant step towards more widespread and immersive robot art, and that it will inspire future work in this area.

Acknowledgment

We graciously thank Catherine Manci for coordinating the Library's Artist-In-Residence program and Georgia Tech Mechanical Engineering undergraduate capstone teams for their contributions constructing the serial manipulator.

References

[1] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998.

[2] ——, "A survey of stroke-based rendering," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 70–81, 2003.

[3] P. Schaldenbrand, J. McCann, and J. Oh, "FRIDA: A collaborative robot painter with a differentiable, Real2Sim2Real planning environment," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 712–11 718.

[4] S. Wang, J. Chen, X. Deng, S. Hutchinson, and F. Dellaert, "Robot calligraphy using pseudospectral optimal control in conjunction with a novel dynamic brush model," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[5] K. Frans, L. Soros, and O. Witkowski, "CLIPDraw: Exploring text-to-drawing synthesis through language-image encoders," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 5207–5218.

[6] Z. Zou, T. Shi, S. Qiu, Y. Yuan, and Z. Shi, "Stylized neural painting," 2020.

[7] P. Schaldenbrand, Z. Liu, and J. Oh, "StyleCLIPDraw: Coupling content and style in text-to-drawing translation," 2022.

[8] Y. Vinker, E. Pajouheshgar, J. Y. Bo, R. C. Bachmann, A. H. Bermano, D. Cohen-Or, A. Zamir, and A. Shamir, "CLIPasso: Semantically-aware object sketching," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530068

[9] D. Ha and D. Eck, "A neural representation of sketch drawings," in *ICLR 2018*, 2018, 2018.

[10] Q. Wang, C. Guo, H.-N. Dai, and P. Li, "Stroke-GAN painter: Learning to paint artworks using stroke-style generative adversarial networks," *Computational Visual Media*, vol. 9, no. 4, pp. 787–806, 2023. [Online]. Available: https://doi.org/10.1007/s41095-022-0287-3

[11] P. Schaldenbrand, G. Parmar, J.-Y. Zhu, J. McCann, and J. Oh, "CoFRIDA: Self-supervised fine-tuning for human-robot co-painting," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[12] G. Chen, S. Baek, J.-D. Florez, W. Qian, S.-W. Leigh, S. Hutchinson, and F. Dellaert, "GTGraffiti: Spray Painting Graffiti Art from Human Painting Motions with a Cable Driven Parallel Robot," in *2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 4065–4072.

[13] G. Chen, S. Hutchinson, and F. Dellaert, "Locally Optimal Estimation and Control of Cable Driven Parallel Robots using Time Varying Linear Quadratic Gaussian Control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Kyoto, Japan: IEEE, Oct. 2022, pp. 7367–7374.

[14] J. Lehni, "Soft monsters," *Perspecta*, vol. 40, pp. 22–27, 2008. [Online]. Available: http://www.jstor.org/stable/40482274

[15] A. Uryasheva, M. Kulbeda, N. Rodichenko, and D. Tsetserukou, "DroneGraffiti: Autonomous multi-UAV spray painting," in *ACM SIGGRAPH 2019 Studio*, ser. SIGGRAPH '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3306306.3328000

[16] C. R. Associati. UFO: Urban flying opera. [Online]. Available: https://carloratti.com/project/ufo/

[17] Aug 2021. [Online]. Available: https://youtu.be/zSIdvQsu27s

[18] Y. Jun, G. Jang, B.-K. Cho, J. Trubatch, I. Kim, S.-D. Seo, and P. Y. Oh, "A humanoid doing an artistic work - graffiti on the wall," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1538–1543, 2016.

[19] S. Wighton. My GIANT wall painting robot. [Online]. Available: https://youtu.be/osUTMnDFV30

[20] D. Berio, S. Calinon, and F. F. Leymarie, "Learning dynamic graffiti strokes with a compliant robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3981–3986.

[21] G. Chen, H. Muriki, A. Sharkey, C. Pradalier, Y. Chen, and F. Dellaert, "A hybrid cable-driven robot for non-destructive leafy plant monitoring and mass estimation using structure from motion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 809–11 816.

[22] F. Dellaert and M. Kaess, *Factor Graphs for Robot Perception*. Foundations and Trends in Robotics, Vol. 6, 2017. [Online]. Available: http://www.cs.cmu.edu/~kaess/pub/Dellaert17fnt.pdf

[23] S. Yang, G. Chen, Y. Zhang, H. Choset, and F. Dellaert, "Equality constrained linear optimal control with factor graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9717–9723.

[24] M. Gouttefarde, J. Lamaury, C. Reichert, and T. Bruckmann, "A versatile tension distribution algorithm for n -dof parallel robots driven by $n+2$ cables," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1444–1457, Dec 2015.

[25] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, "NIMS-PL: A cable-driven robot with self-calibration capabilities," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1005–1015, 2009.

[26] P. Miermeister, A. Pott, and A. Verl, "Auto-calibration method for overconstrained cable-driven parallel robots," in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.

[27] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd, Ed. Springer-Verlag, Berlin, Heidelberg, 2022.

[28] G. Chen, F. Dellaert, and S. Hutchinson, "Generalizing trajectory retiming to quadratic objective functions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.