Tethered flight control of a small quadrotor robot for stippling

Brendan Galea and Paul G. Kry

***Abstract*— We investigate tethered flight of a small quadrotor robot in the context of creating stippled prints. At a low level, we use motion capture to measure the position of the robot and the canvas, and a robust control algorithm to command the robot to fly to different stipple positions to make contact with the canvas using an ink soaked sponge. With the objective of fully autonomous flight, we power our quadrotor using a wired tether. We compensate for the tether in our control of the robot by assuming a static catenary curve of fixed length between the robot and the power source, and model the forces and torques produced. We evaluate accuracy of hovering and flight on simple paths, and compare the results to untethered flight.**

I. INTRODUCTION

We are interested in the problem of applying ink to paper with aerial robots, ultimately for the creation of murals on large walls and on hard to reach surfaces. Our focus is exclusively on stipples, that is, drawing a picture using many small dots, because this lets us avoid the hard problem of controlling continuous contact between an aerial robot and a canvas. In our work, we use a Crazyflie quadrotor equipped with an ink soaked sponge mounted at the end of a small arm. Crazyflie quadrotors are an excellent platform for research and development because they have an open hardware and software design and the development environment is very well organized. They are also small and light, which makes them very safe in comparison to larger quadrotors.

Our previous work [1] provides details of how we control an untethered robot to produce stippled prints. This includes important low level details and challenges that must be addressed for successful control, such as robot model estimation, Kalman filtering for state estimation, latency between motion capture and control, radio communication interference, and control parameter tuning. At a high level, this involves computing a stipple pattern for an image, greedy path planning, a model for how ink is used up as stipples are placed, and a technique for dynamically adjusting future stipples based on past errors. Figure 1 shows a stippled print, in progress, but also highlights an important limitation. Because our controller can only draw a stipple once every five to six seconds, and with batteries that only last about six minutes, bigger prints require a large amount of user intervention to swap batteries and replenish ink on the sponge used for stippling.

In this paper we investigate the use of a tether to deliver power and an in-flight mechanism for refilling ink, with the goal of ultimately scaling up to tens of thousands of stipples. An important challenge is that the tether applies an additional load to the robot. With a larger robot, the additional force applied by a tether might be considered as negligible, but this is not the case for the Crazyflie, which has a mass of 29.4 g (the 240 mAh battery adds an additional 5.8 g). We observe that our untethered control degrades to unacceptable levels when we attach a very light tether. Thus, in this paper we augment our existing controller with a model of the tether based on the tension in a static catenary curve. Furthermore, we learn a simple model for the torque that the tether applies on the robot due to the attachment point and the current hovering location. We evaluate these models, and perform experiments that likewise dampen the dynamic motion of the tether by attaching at its midpoint a fine string that drags on the ground.

Fig. 1. The untethered quadrotor executing a stippled print of Grace Kelly with 2000 dots. With an untethered robot, approximately 40 separate flights are required to complete the print due to battery limitations.

This work was supported by NSERC.

Brendan Galea and Paul G. Kry are with the School of Computer Science, McGill University, Montreal, Canada.

E-mail: {brendan.galea,paul.kry}@mcgill.ca

II. RELATED WORK

Many control problems specific to quadrotors have been investigated, such as methods to produce aggressive maneuvers [2] and flips [3]. The robot control algorithms we use in this paper and our previous work [1] are largely inspired by that of Mellinger et al. [2], as well as the PhD thesis of Landry [4].

For continuous flight not bounded by battery capacity, we power the quadrotor using a wire tether. We model the shape of the hanging wire and include the forces it generates into our controller. While these forces may be negligible for larger robots or disregarded in other applications where absolute position accuracy is less important [5], [6], in our application it is critical because of the small robot size and our desire for the accurate placement of stipples.

A problem related to adding a tether to the robot is control and planning while taking into account the dynamics of a slung load [7], [8]. When the slung load is also allowed to become slack, the problem becomes even more challenging [9]. We take a simpler approach of assuming that the dynamics of the cable are small, given the damping effects on the very light weight wire we use, and we show that accounting for a static tether can make a significant improvement.

Tethers are often used for safety, and for testing new control algorithms, but they are also common for providing power or for communication. The tether modeling of Zikou et al. [6] is similar to ours, though they focus on controlling a spool to provide additional length or take up slack as necessary. Given that they use a larger robot, they let the existing quadrotor control handle the disturbance introduced by the weight of the tether.

Finally, we note that there are many examples of previous work using robots in the creation of art and drawings. Paul the robot produces sketches and portrait drawings of [10], [11]. Lin et al. [12] use a camera and a humanoid robot to create a line drawing portrait a person. Lu et al. [13] also use cameras and visual feedback to create images with hatching patterns that capture both texture and tone of the original image. Robots have been used to apply paint to canvas with feedback guided stroke placement [14], image stylization with semantic hints [15], and dynamic adjustment of layered strokes [16]. In this work, we address challenges of control of a small aerial robot with an attached tether with the goal of scaling up our existing technique of stippling with an aerial robot.

III. TETHERED FLIGHT

Our previous work [1] outlines our method for semi-autonomous stippling, with the limitations being short flight times requiring frequent battery changes and refilling the ink supply. We present here our methods for achieving fully autonomous stippling.

For fully autonomous stippling, an entire drawing containing thousands of stipples should require no user interaction following start to finish. Considering the duration of the quadrotor's battery is typically six minutes, this would require an impossibly fast rate of stippling. Wireless charging of the battery by having the quadrotor accurately land on a charging platform was considered as a possible solution. One disadvantage to this approach is that the brush could dry out during charging. Instead we remove the battery, and power the quadrotor through a wire connected to a power supply. Such a solution has accompanying problems that must be overcome.

A. Voltage drop

Stippling requires very accurate position control. In order for the wire not to affect flight control its weight must be small compared to that of the quadrotor but also sufficiently long to allow stippling on a large canvas. We use 30AWG wire 180 cm in length weighing 2.4 g. The Crazyflie requires up to 3 A of current in the range of 3.0 V to 4.2 V to run properly. The resistance of the wire is not negligible, roughly $0.6\ \Omega$ for the length we use, and there is a significant voltage drop across the wire. The resistance of the quadrotor is variable and dependent on the thrust, so increasing the supplied voltage is not a viable solution as the quadrotor would receive an irregular voltage as the thrust varies. To counteract this we attach a small buck converter to the quadrotor which drops the 12 V coming from the power supply to a consistent 3.3 V.

B. Tether model

Even at only 2.4 g, the force and torque from the wire acting on the quadrotor has a negative impact on the control. We account for this by modeling the wire and estimating the force and torque acting on the quadrotor. Similarly to how the flight controller accounts for gravity, tension is treated as an additional external force. However, unlike gravity which does not induce torque, connecting the wire directly to the center of mass of the quadrotor is not feasible and therefore the wire's tension induces torque.

We model the hanging wire using a catenary curve, which is the shape an idealized hanging cable assumes under its own weight when supported only at its ends. One endpoint of the cable is fixed to a stand (90 cm high) while the other is attached to the base of the quadrotor. The motion capture runs at a frequency of 100 Hz. In each time step, we numerically solve for the unknown parameters a, b, and c in

$$F(x) = a + \frac{1}{b}\cosh\left(b(x-c)\right), \tag{1}$$

obtaining the catenary curve that a cable of fixed length would assume by the two endpoints. We can find the direction of the tension forces by computing the derivative of the curve at each endpoint. We can then analytically solve for an approximation of the tension forces with the assumption that the system is at equilibrium.

For the case of accurate stippling, the ability to hover accurately at a point in preparation of stippling is paramount; it is only when our system detects that the position error is below a desired threshold that the quadrotor will perform the stipple. Therefore, the assumption that the system is at rest is

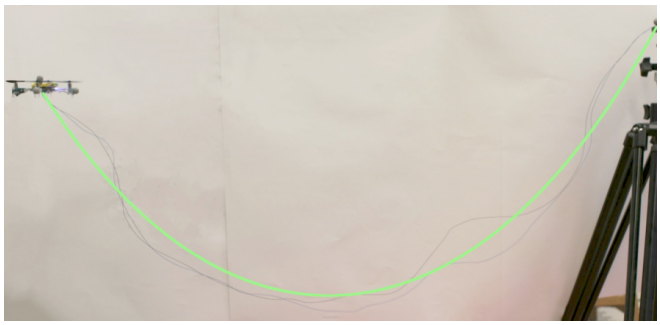

Fig. 2. The quadrotor in flight attached to the tether with the overlaid green line showing the curve our model predicts.

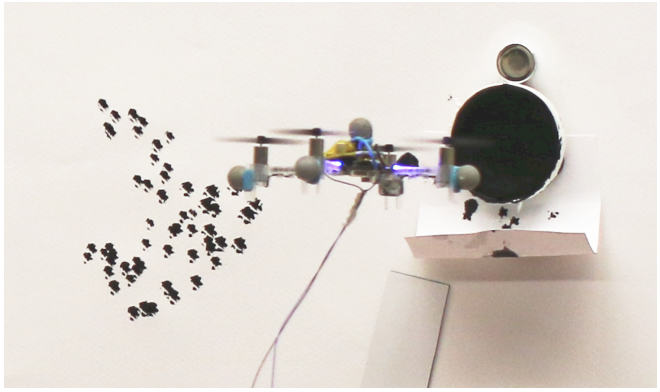

Fig. 3. Quadrotor is preparing to refill mid-flight using the ink well. To construct the ink well, we cut out the bottom of a coffee cup. We then cut a sponge to fit and glue it to the base of the cup. Before starting a drawing, we fill the sponge of the ink well to capacity, such that it is fully saturated but will not drip. We finally attach it to the canvas and record its location so the stipple server knows where to send the quadrotor for refilling.

not too restrictive in regards to stippling because the system will ideally be at rest when accuracy is most required.

Oscillations of the sagging tether is one problem we faced early on in this work. The quadrotor's ability to maintain a stable hover is reduced when the tether is swinging (we discuss and evaluate the effect in Section IV). The problem is most pronounced when the endpoints of the cable are relatively close together compared to the cable's length. We dampen these oscillations by tying a lightweight thread to the midpoint of the cable. This does not have a noticeable impact on the mass or shape of the cable, but frictional contact between the thread and the floor dampens the swinging motion.

C. Autonomous ink refill

The final hurdle for fully autonomous flight is the ability for the quadrotor to autonomously replenish its ink while stippling. Our solution is elegant in that it uses the existing controller with no need for modification. We make ink wells and attach them next to the canvas. When the ink model predicts that the ink is running low, we set the quadrotor's target stipple location to the center of the ink well. The quadrotor stipples the ink well repeatedly to refill its ink. All of this is controlled by the stipple server. As far as the quadrotor controller is aware, it is performing a stippling motion no different than it would on the canvas. We use an exponential decaying ink model constructed as described in our previous work [1]. Through experimentation we find that ten refill stipples result in a consistent maximum capacity regardless of the amount of ink already in the quadrotor's sponge. However, for longer drawings, to keep the refill consistent for drawing the desired range of stipple sizes, the ink well must be refilled roughly every 500 stipples.

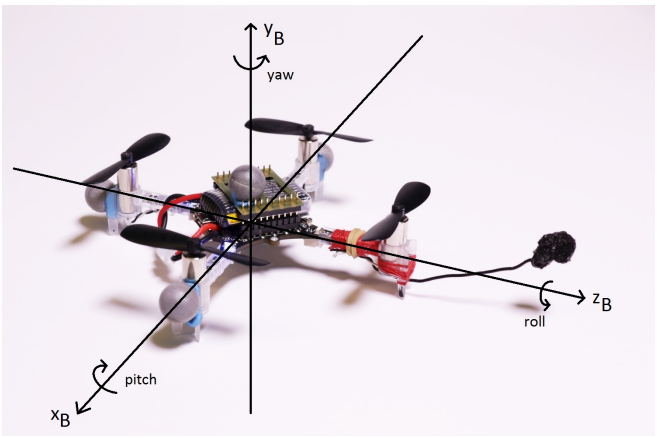

Fig. 4. The quadrotor with a diagram of its body frame overlaid.

D. Torque model

The catenary model provides us with an estimate of the force of tension caused by the wire, but not the torque acting on the quadrotor. Our simplified model of the quadrotor does not provide an inertia tensor. Also the lever arm is not known, since there is no estimate of the center of mass. For these reasons an experiment was performed to learn the torque induced on the quadrotor from the tension force using a simplified model.

For our model we assume that the inertia tensor can be approximated by a diagonal matrix with entries I_{xx}, I_{yy}, and I_{zz}, and that the principal axes run along the arms of the quadrotor as can be seen in Figure 4. The entries I_{xx}, I_{yy}, and I_{zz} correspond to the pitch, yaw, and roll of the quadrotor respectively. Let F be the force of tension exerted by the tether on the quadrotor and r be the lever arm from the quadrotor's center of mass to the attachment point of the tether. We only care about the torque acting on the pitch and roll of the quadrotor, which are given by

$$T_p = I_{xx} r_y F_z - I_{xx} r_z F_y \tag{2}$$

$$T_r = I_{zz} r_x F_y - I_{zz} r_y F_x \tag{3}$$

respectively. Both Equations 2 and 3 are linear with respect to the tension force. To construct the model to approximate the torque we identify the two unknown parameters for each equation using the experiment described below.

First, the quadrotor is flown to a variety of positions relative to the mounting point and allowed to stabilize. The integral term of the quadrotor's on-board PID captures the

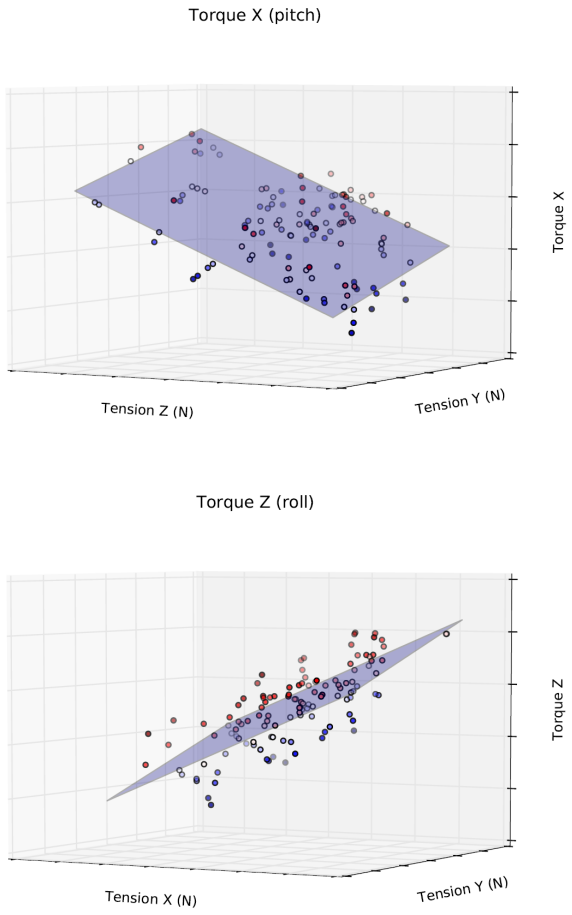

Fig. 5. Torque Model trained for one of the quadrotors. Prediction errors are colour coded, with darker red corresponding to larger positive errors, and darker blue larger negative errors.

torque induced by the tension force from the wire. The tension force as well as the integral term of the quadrotor is recorded. The tension force is mapped into the quadrotor's coordinate frame and a multi-linear model is used to fit the data. For one of our tethered quadrotors (we configured several Crazyfiles for tethered flight) we obtained

$$T_p = 2177.69F_z - 642.87F_y - 16.89 \quad (4)$$

$$T_r = 740.48F_y + 1963.97F_x + 6.51. \quad (5)$$

The intercepts for Equations 4 and 5 correspond to the trim of the quadrotor, a small constant offset applied to a control in order to make an aircraft fly correctly. The trim encapsulates many unknowns such as differences in thrust generated by the motors and asymmetries in the distribution of the mass. If it were possible to use a massless tether, the integral of the on-board PID should be constant and approach these values. The brush is responsible for the large negative value of -16.89 as the intercept in Equation 4.

With the multilinear torque model, the torque acting on the quadrotor can be predicted from the tension force and sent to the on-board controller. The multilinear model

TABLE I

HOVER TESTS USING TETHER TENSION ONLY, WHERE μ IS THE MEAN ERROR AND σ IS THE ERROR STANDARD DEVIATION IN CM, WITH MODEL INDICATED BY +M, AND WITH DAMPENING INDICATED BY +D.

Hover Tests	μ_x	μ_y	μ_z	σ_x	σ_y	σ_z
Control	-0.10	2.80	2.68	1.58	0.66	1.35
Near	-0.60	2.49	0.26	3.60	1.11	2.31
Near +M	0.05	-0.18	1.07	3.79	0.44	1.75
Near +D	-1.02	3.14	1.76	2.77	0.45	2.04
Near +M +D	-0.67	-0.04	1.00	1.27	0.42	1.56
Far	0.59	1.86	3.80	2.10	1.13	5.87
Far +M	0.41	-1.03	0.21	1.25	1.02	2.06
Far +D	0.48	3.91	3.50	2.51	1.13	4.19
Far +M +D	0.12	-0.86	0.51	1.56	1.71	3.19

TABLE II

COMPARISON OF HOVERING WITH BATTERY (CONTROL), DAMPENED TETHER WITH TENSION MODEL (TETHERED), AND DAMPENED TETHERED WITH TENSION AND TORQUE MODEL, WHERE μ IS THE MEAN ERROR AND σ IS THE ERROR STANDARD DEVIATION IN CM.

Torque Hover Test	μ_x	μ_y	μ_z	σ_x	σ_y	σ_z
Control	-0.10	2.80	2.68	1.58	0.66	1.35
Tethered	0.12	-0.86	0.51	1.56	1.71	3.19
Torque	0.10	1.84	1.40	1.48	0.78	1.49

does not account for all the variability. The coefficient of determination is only 0.613 and 0.764 for roll and pitch respectively. However, we use the model in combination with the on-board PID, which results in faster convergence for the integral term and overall better control. The improvement in control is most noticeable when quickly traveling between two far away points, since the torque acting on the quadrotor will vary drastically.

The main disadvantage of this method is that it is specific to a given robot, and there is certainly some variation across our robots. For different configurations, such as changing the brush length, marker placement, or small variations in the tether attachment across different quadrotors, the process of learning the linear torque model must be repeated. Nevertheless, it is worth the effort. Predicting the torque and including a feed-forward term to the on-board PID drastically improves the stabilization time of the on-board PID, which in turn provides improvements in overall control.

In addition to the usual control commands, the torque computed for the pitch and roll are sent to the on-board controller. The on-board controller is then modified to make use of these values when computing the commands to send to the motors. The integral term of the on-board PID controller for the pitch and roll is modified and computed as

$$k_I \left(T + \int_0^t e(t) \right), \quad (6)$$

where k_I is the integral coefficient, T is the torque, and $\int_0^t e(t)$ is the accumulated error in orientation over the course of the flight.

IV. EXPERIMENTS AND DISCUSSION

Flight tests were performed to measure the quadrotor's ability to maintain a fixed position in order to understand better the variability of the hover controller position when flying with the tether. Each hover test was performed twice, once close to the fixed endpoint at equal height and 40 cm away in the z direction, and once again farther away (140 cm) at the same height. The x direction measures the horizontal error in the direction orthogonal to the plane in which the catenary curve lies, while the y and z directions indicate the vertical and horizontal error within the plane of the curve.

Table I summarizes the results of experiments where the controller is only accounting for the tension force of the catenary model, and ignoring the torque caused by the tension. The dampened tests uses a single thread tied to the midpoint of the tether to dampen oscillations. It is clear that modeling the tension of the tether results in improvements in control. The standard deviation of the errors is smaller in almost all instances compared to the tests where tension is not modeled. For the results in Table I, the σ_x error is reflective of errors introduced by cable oscillation. The more the cable sags, the worse the oscillation, which can be seen by comparing the larger σ_x error of the non-dampened near tests to that of the far tests. However, dampening the tether mostly eliminates this discrepancy.

Finally, comparing the σ_z illustrates the necessity of providing an estimate of the torque to the controller. When hovering far away, σ_z errors are larger. This has to do with the torques being produced by the tension force. When the tension is pointing mostly down, little torque is produced since the cable is attached to the quadrotor at a point below its center of mass. When the robot is far away from the fixed endpoint, the horizontal tension component is larger and produces a greater torque on the quadrotor.

Table II shows a comparison of hover control tests, quantifying the improvements that the torque model provides. The control test is performed using battery powered flight rather than the tether. The tethered test is using a dampened tether and the controller is modeling the tension but not the torque. The torque test is also using a dampened tether, but is modeling torque as well as tension caused by the tether using the learned parameters as described in Section III-D. The hover test was repeated at the distance of 140 cm away in the z direction. The hovering results of the quadrotor when using the torque model are comparable to that of the untethered control; it even out-performs the control with regards to the σ_x. This may be due to the dampening thread providing a stabilization effect on the side to side motion of the quadrotor. Most notable is the improvement of the σ_z error for tethered hovering, comparing the error with and without the torque model. The large error produced by the tether is reduced to a level similar to that of the untethered hover control tests.

In general, the results of tethered stippling is comparable to that of untethered. For a small trade-off in the accuracy of stipple placement and rate of stipples, the onerous process of constantly refilling ink and swapping batteries becomes fully automated. Following the initial setup of the canvas and the ink well, we have observed our tethered quadrotor complete over 800 stipples with no interference. Tethered stippling also seems to be just as reliable as untethered. The only notable cause of failure throughout our experimentation is that the radio occasionally disconnects, and this occurs equally often in both tethered and untethered flights. Tethered flight works well for the cable length we use, but the operating volume is limited and only allows for stippling on moderately sized canvases. For other applications, tethered flight may be an important restriction. For example, the navigation around obstacles as demonstrated by Landry [4] would be impossible.

TABLE III

STIPPLING TEST RESULTS, WHERE t IS THE AVERAGE TIME IN SECONDS PER STIPPLE, μ_h AND μ_v ARE THE STIPPLE ERROR MEANS IN THE HORIZONTAL AND VERTICAL DIRECTIONS IN MM, AND σ_h AND σ_v ARE STIPPLE ERROR STANDARD DEVIATIONS IN THE HORIZONTAL AND VERTICAL DIRECTIONS.

Tethered Stippling	t	μ_h	μ_v	σ_h	σ_v
Control	3.8	0.97	-1.1	7.2	4.2
Tethered	4.5	1.05	0.3	7.2	4.5

As a final evaluation, we performed a stippling test to compare the tethered model to that of untethered control. The stippling test consists of the quadrotor repeatedly colliding with the canvas at set target locations a total of 120 times. Errors are computed as the difference between the target location and the projected position of the quadrotor's sponge onto the canvas at the time of collision. For the tethered test, the tether was dampened and we provided both tension and torque estimates to the quadrotor. Tethered tests without the model were also performed, but only a few stipples at most could be performed before the quadrotor would crash; for that reason these results are not included in Table III. As can be seen Table III, tethered stippling performs very similarly to untethered stippling, with the only prominent difference being a slightly slower stippling rate. This is possibly because the catenary model does not account for the dynamics of the tether as the quadrotor travels between points.

V. CONCLUSIONS

We present a method for fully autonomous stippling using a tether to provide power in order to not be limited by the capacity of batteries. We describe a model for the tether and how the existing controller can be easily extended to account for addition forces and torques acting on the quadrotor. Our experiments show that it is critical to account for the tether for our lightweight robots, and that that we can ultimately achieve a comparable level of accuracy in hover control when we fly these robots with a power tether.

There are a variety of other exciting related avenues for future work. Although we have eight robots in our fleet, we only use one at a time for stippling. When creating a larger print, it would be advantageous to coordinate multiple robots

to reduce the total printing time. While there is exciting work on the coordinated control of fleets of flying robots, we are not aware of any research on coordinating a collection of tethered quadrotors.

REFERENCES

[1] B. Galea, E. Kia, N. Aird, and P. G. Kry, "Stippling with aerial robots," in *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, ser. Expresive '16. Eurographics Association, 2016, pp. 125–134.

[2] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[3] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1642–1648.

[4] B. Landry, "Planning and control for quadrotor flight through cluttered environments," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[5] M. Srikanth, K. Bala, and F. Durand, "Computational rim illumination with aerial robots," in *Proceedings of the Workshop on Computational Aesthetics*, ser. CAe '14, 2014, pp. 57–66.

[6] L. Zikou, C. Papachristos, and A. Tzes, "The power-over-tether system for powering small UAVs: Tethering-line tension control synthesis," in *Control and Automation (MED), 2015 23th Mediterranean Conference on*, June 2015, pp. 681–687.

[7] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4888–4895.

[8] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for UAVs with a suspended load," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4902–4909.

[9] C. de Crousaz, F. Farshidian, M. Neunert, and J. Buchli, "Unified motion control for dynamic quadrotor maneuvers demonstrated on slung load and rotor failure tasks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2223–2229.

[10] P. A. Tresset and F. F. Leymarie, "Sketches by Paul the robot," in *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, ser. CAe '12, 2012, pp. 17–24.

[11] P. Tresset and F. F. Leymarie, "Portrait drawing by Paul the robot," *Computers & Graphics*, vol. 37, no. 5, pp. 348 – 363, 2013.

[12] C.-Y. Lin, L.-W. Chuang, and T. T. Mac, "Human portrait generation system for robot arm drawing," in *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, July 2009, pp. 1757–1762.

[13] Y. Lu, J. H. Lam, and Y. Yam, "Preliminary study on vision-based pen-and-ink drawing by a robotic manipulator," in *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*. IEEE, 2009, pp. 578–583.

[14] O. Deussen, T. Lindemeier, S. Pirk, and M. Tautzenberger, "Feedback-guided stroke placement for a painting machine," in *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, ser. CAe '12, 2012, pp. 25–33.

[15] T. Lindemeier, S. Pirk, and O. Deussen, "Image stylization with a painting machine using semantic hints," *Computers & Graphics*, vol. 37, no. 5, pp. 293 – 301, 2013.

[16] T. Lindemeier, J. Metzner, L. Pollak, and O. Deussen, "Hardware-based non-photorealistic rendering using a painting robot," *Computer Graphics Forum (Eurographics)*, vol. 34, no. 2, pp. 311–323, 2015.