

# Stippling with aerial robots

Brendan Galea, Ehsan Kia, Nicholas Aird, Paul G. Kry  
School of Computer Science, McGill University



## Overview

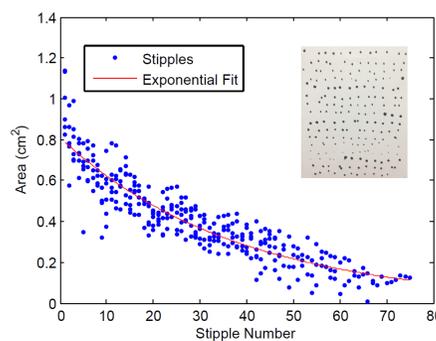
We create stippled prints using a quadrotor flying robot. At a low level, we use motion capture to measure the position of the robot and the canvas, and a robust control algorithm to command the robot to fly to different stipple positions to make contact with the canvas using an ink soaked sponge. Important details and challenges that must be addressed for successful control in our implementation include robot model estimation, Kalman filtering for state estimation, latency between motion capture and control, radio communication interference, and control parameter tuning. We use a centroidal Voronoi diagram to generate stipple drawings, and compute a greedy approximation of the traveling salesman problem to draw as many stipples per flight as possible, while accounting for desired stipple size and dynamically adjusting future stipples based on past errors. An exponential function models the natural decay of stipple sizes as ink is used in a flight. We evaluate our dynamic adjustment of stipple locations with synthetic experiments. Stipples per second and variance of stipple placement are presented to evaluate our physical prints and robot control performance.

## Flight Control

We use a simplified model that represents the quadrotor as a point mass that can align its pitch and roll instantly. Using position information from the motion capture system that is processed with a predictive Kalman filter, a PID controller computes the desired steering force. The desired pitch and roll are computed so that the  $y_b$  (i.e., robot body vertical) will be pointing in the same direction as the steering force. We control the yaw of the quadrotor with a separate P controller that calculates the yaw-rate. All of these values are sent to the quadrotor, which uses an onboard PID controller to achieve the desired orientation.

## Stipple Generation

Stipple generation is the process that takes a selected image and converts it into a set of stipples given a set of constraints such as the number of stipples and the range of stipple sizes. This set of stipples updates dynamically throughout a flight, adjusting the remaining stipple locations using feedback from the small errors the quadrotor makes when placing stipples. Starting with a Voronoi diagram from random points, we compute centroids for each region by integrating over the pixels of the target image using the brightness as weights. We then shift each point to the centroid of its region and repeat these steps until we reach a stable configuration.

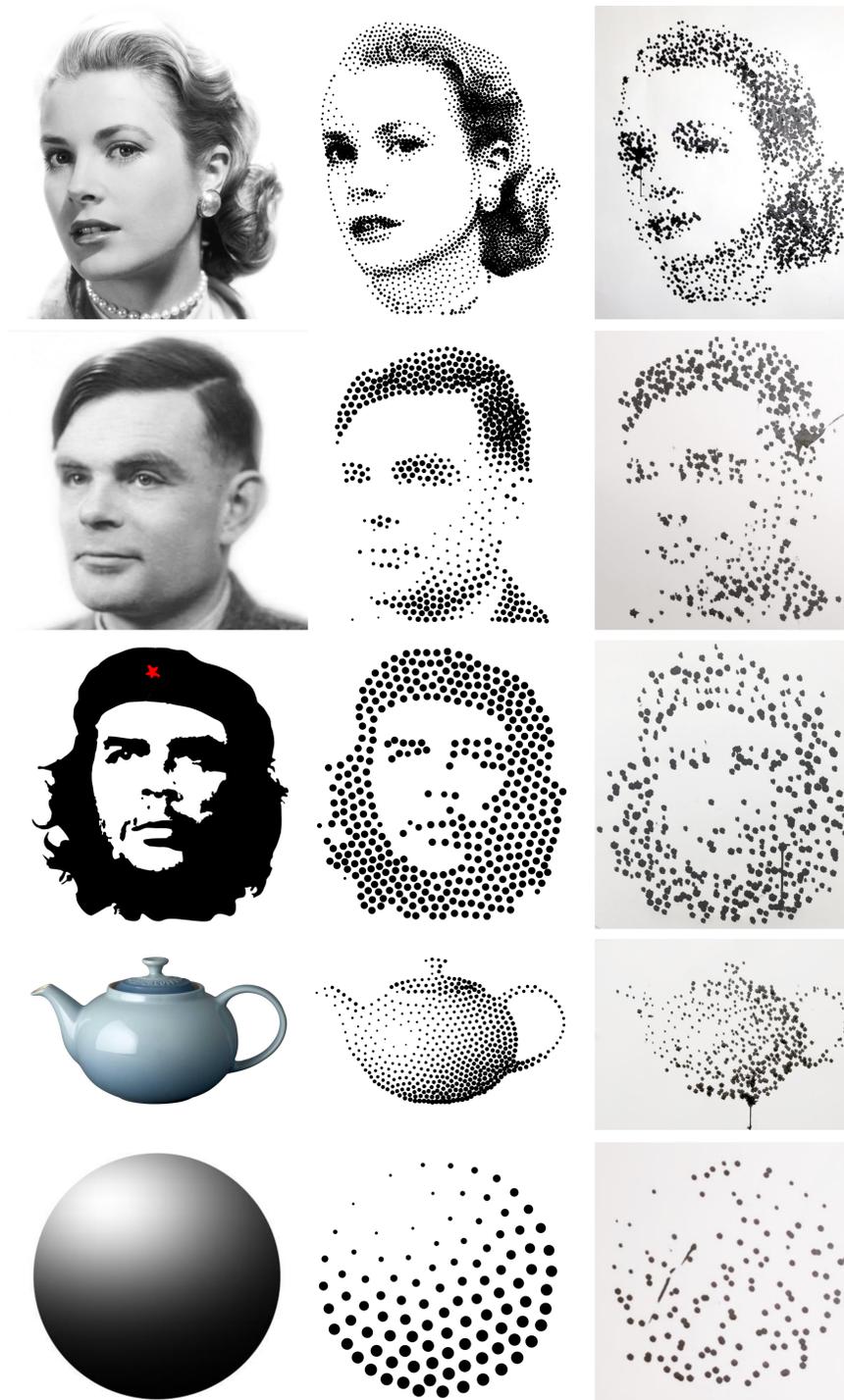


Using a set of six sequences, we measure the area of the stipples and build an approximate ink decay model by fitting an exponential function. This allows us to predict the size of the next dot the robot will draw and thus helps to determine the ordering of stipple placement. Once a stable configuration is reached, a stipple is generated within each Voronoi region where we linearly map the average pixel brightness to a stipple size in the available range.

The stipple size range is given by a maximum and minimum that we experimentally determined for the ink decay model. A greedy strategy is used to determine stippling order. At any given time, the next optimal stipple is selected as closest using a metric that combines distance and stipple area.

## Physical prints

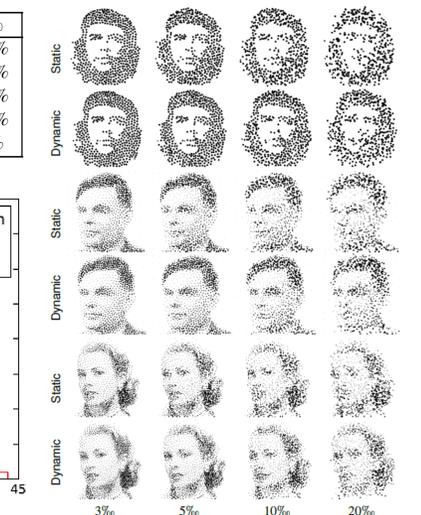
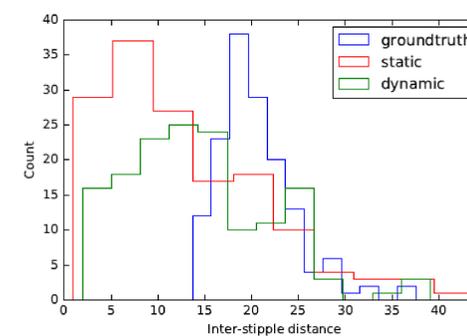
Comparison of real images, planned stipple positions, and results of stippling with the flying robot. Stipple counts for the sphere, teapot, Che, Turing, and Grace are 100, 500, 400, 500, and 2000 respectively.



## Dynamic stipple adjustment evaluation

Below right shows synthetic results to compare stipples drawn with and without dynamic stipple placement correction for different amounts of error. Error standard deviation of stipple locations in different columns is shown as per-mil of the canvas size. Below left shows a summary of percent difference between the mean inter-stipple distance of the static versus dynamic case compared to ground truth, for various images of different stipple counts, and for different per-mil error sizes. The histogram at bottom left shows the distribution of inter-stipple distances for the teapot image with 200 points and a 1% standard deviation Gaussian error. Comparing the original ground truth output without errors (blue), with the static stippling with error (red), and the dynamic error adjusted stippling (green).

Image	$n$	3‰	5‰	10‰	20‰
Sphere	100	18.2%	20.7%	13.3%	23.7%
Teapot	250	22.2%	33.3%	38.0%	31.4%
Che	500	26.7%	28.6%	31.9%	16.6%
Turing	750	54.5%	31.3%	33.8%	26.8%
Grace	1000	45.2%	33.6%	26.8%	4.1%



## Evaluation of stippling process and prints

Print information from selected images, where  $t$  is the average time in seconds per stipple,  $\mu_h$  and  $\mu_v$  are the stipple error means in the horizontal and vertical directions in mm,  $\sigma_h$  and  $\sigma_v$  are stipple error standard deviations in the horizontal and vertical directions in mm,  $n_f$  is the number of flights, and  $\bar{n}_s$  is the average number of stipples per flight.

Image	$t$	$\mu_h$	$\mu_v$	$\sigma_h$	$\sigma_v$	$n_f$	$\bar{n}_s$
Che	6.7	1.70	-2.1	6.9	4.6	10	40
Turing	6.7	0.99	-4.1	6.6	3.8	8	62
Teapot	6.4	0.23	-3.1	7.9	4.2	10	50

## References and video

Galea, B., Kia, E., Aird, N., Kry, P.G.: Stippling with aerial robots. In: Computational Aesthetics (Expressive 2016). 10 pages (2016)

<https://goo.gl/PCF78w>

