Fireflie: Generalized Multi-Agent Light Painting

Ryan Peng
School of Engineering, Brown University
Department of Computer Science, Brown University
Providence, USA
ryan_peng@brown.edu

***Abstract*–The application of drones in art and entertainment has become increasingly common. Here, in the application of light painting, the ability to generate and execute conflict-free complex fixed trajectories is demonstrated. Light painting differs from traditional multi-agent path finding (MAPF) problems, which often involve solving for paths from initial starting states to goal states, as trajectories cannot be changed. Instead, Fireflie resolves drone conflicts using a solver inspired by conflict-based search (CBS) which exploits the ability to reverse drone path orientations while achieving the same end result. Fireflie is able to successfully generate recognizable light paintings for two iconic works with few concessions, and provides a generalized multi-agent light painting pipeline with the ability to create real-world executable drone trajectories from any source imagery.**

I. Introduction

While applications for technology are often boundless, some of the most creative usages come, perhaps unsurprisingly, from the world of art. By approaching results-first, we can bend and experiment with technology until the expectation is achieved.

Drones have seen some of the most visible applications in drone shows, with each drone the spark of a technologically-advanced, controllable "firework." In 2017, Lady Gaga started her National Football League Super Bowl halftime show by jumping (in a pre-recording) into NRG Stadium with a backdrop of 300 drones which swirled in red, white, and blue before forming the American flag [1]. Recently, as many as 7,598 drones were used simultaneously to form aerial images in Yanji, China [2]. In these aerial images, every drone represents a "pixel" of the image in two or three dimensions, with the transition and movement of drones pre-planned to avoid drone collision.

Fundamentally, the drone shows consist of different states for different images, and the transitions between them. In light painting, which uses a light source to "paint" in long exposure photography, the transitions and pathing itself is what creates the image, not the individual keyframe positions of the drones. This application deviates from the traditional layout of multi-agent planning methods, which define an initial starting state and goal state. For instance, when planning for avoidance while painting, two agents cannot simply change the path between set states to avoid each other. Instead, the order and choice of pathing must be decided in a way to minimize the possibility of conflicts, while producing the same resulting set of paths.

Fig. 1. Crazyflie 2.1+ miniature quadcopter with a single motion-capture marker, battery, and LED expansion board (facing downwards). A white cardboard ring is fixed to the inside of the LED ring to allow for the LED to show when viewing from the sides.

Fireflie is a generalized pipeline for multi-agent light painting using the Crazyswarm platform [3] and Bitcraze Crazyflie 2.1 quadcopters as shown in Fig. 1. Using inspiration from previous work with PointillistQuad, a large quadcopter which recreated famous works of art by spraying acrylic paint onto a canvas [4], Fireflie also demonstrates the ability to form complex recreations starting from an image, replacing the raster paint "dots" with vector "brushstrokes." Fireflie differs from existing drone light painting implementations [5] by using multiple drones at a time, and planning based on the source image automatically, instead of using manually programmed waypoints.

II. Approach and Methods

A. Image Processing

In order for reasonable flight paths to be extracted, an image can be constrained using the simple framework in Fig. 2. Any image can be simplified to a set number of closed colored regions, with the boundaries of these regions acting as the drone paths. Each boundary is colored based on the fill in the original image. However, because the boundaries between regions are touching, resulting in overlapping paths, regions must be separated by "void" space. This is done by adding a black and white preprocessing step before forming the region, creating separate, colored paths. Each path is then converted into a piecewise parametric representation for flight.

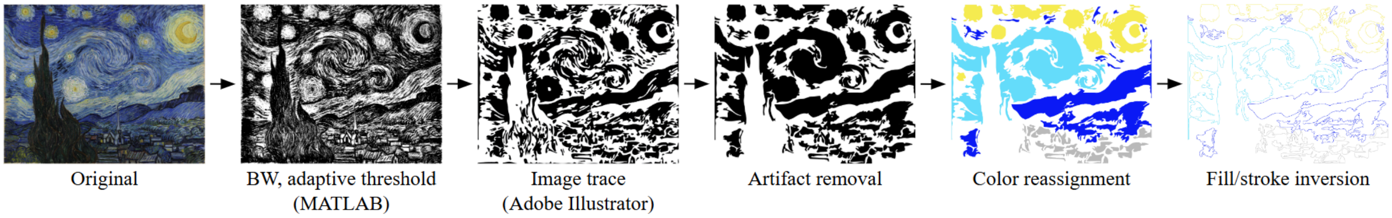

Fig. 2. Image processing pipeline demonstrated with *The Starry Night* [6]. Built-in image processing functions in MATLAB and Adobe Illustrator are initially used with manual inputs to isolate working regions. Further artifact removal, color reassignment, and fill/stroke inversion is done in Adobe Illustrator manually, with the intent of minimizing the number of regions and maximizing the size of regions, while retaining artistic detail and recognizability of key features.

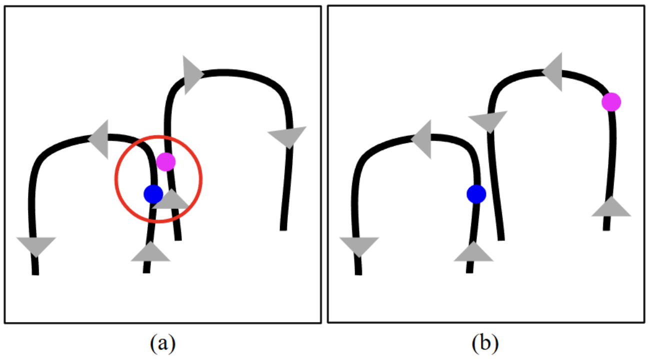

Fig. 3. Illustration of path-orientation-based conflict. (a) Conflict in drone path. (b) Conflict resolved by flipping the orientation of the second path.

B. Waypoint Creation

Using groups of parametric equations, each designated to one drone, waypoints are generated per-drone, separated by a step distance. Each waypoint is assigned the proper corresponding color and is grouped by path, defined as a set of continuous parametric equations. For discontinuities, interpolated points are added between the end of one path and the start of another. These points do not belong to a group.

C. Collision Avoidance Solver

Using a solver inspired by conflict-based search [7], all paths are given an orientation. The solver takes the waypoints and paths per drone as well as an allowable distance threshold as input. In a high-level search, a list of drone conflicts is generated based on the waypoints. A conflict is defined as any instance in time where two drones are separated by a distance less than the threshold. Then, the solver attempts low-level resolutions of each conflict by flipping the orientation of either drone path as demonstrated in Fig. 3. The solver succeeds when no conflicts remain after a set number of iterations, 100 for most pieces. For simple pieces, this implementation itself is sufficient to find a feasible solution.

For complex pieces, the method can be extended further by building a binary tree of individual conflict resolutions. This accounts for the case that flipping either path results in a valid solution. Then, both solutions are considered in further iterations before finding a valid path through the tree. The simpler one-layer, or non-tree implementation can be considered as simply defaulting to choosing the left node at all times there are two valid resolutions.

Unlike conflict-based search, there currently is no cost function which separates one conflict resolution from another–currently they are treated as equal in cost. As such, it is possible for the solver to reach multiple valid solutions, especially with high threshold values as input. Future implementations could integrate a cost function based on the cost of repositioning, correlated with the distance from the previous path to the start of the current path.

Optimization of the threshold value is done using a simple binary search algorithm. The highest threshold value that passes the collision avoidance solver is used to generate the final solution for flight.

D. Plane Assignment

In cases of low optimized threshold values (< 0.5 meters), additional spacing can be added by assigning drones to separate virtual planes spaced by a fixed distance, parallel to the image plane. This creates an additional plane assignment problem, where different combinations are created by assigning different drones to different planes. Assuming n drones and m planes, there are

$$\prod_{i=0}^{r-1}\binom{n-qi}{q+1}\cdot\prod_{i=r}^{m-1}\binom{n-qi}{q}\cdot\binom{m}{r} \tag{1}$$

ways to assign drones, where $q = \lfloor \frac{n}{m} \rfloor$ and $r = n \bmod m$. The optimization is redone with this plane assignment.

E. Camera Transform

If multiple planes are used, a backward projection is applied to convert the waypoints from film coordinates to world coordinates, given an optical center, camera position, and arbitrary focal length.

III. RELATED WORK

Drones have been used in various art applications, including choreographed performances and painting [8] [9] [10]. Less related work exists in the specific application of light painting. Light painting has been used as a method to demonstrate gesture control of single Crazyflie agents [11]. The ACT Lab has done previous work in light painting, including with multiple agents for a demo for South by Southwest (SXSW) [12]. Instead of vectorization, the demo fitted contours to a black and white image and extrapolated waypoints from the contours. Each drone also flew a set, manually curated path to account for drone-drone interference or collisions. Additionally, low-level commander control was used, detailed more in the following Section IV.C.

(a) (b) (c) (d)

Fig. 4. (a) *Composition II in Red, Blue, and Yellow*, Piet Mondrian. [13] (b) *Composition II* Fireflie adaptation. (c) *The Starry Night*, Vincent van Gogh. (d) *The Starry Night* Fireflie adaptation.

IV. IMPLEMENTATION

A. Image Processing

The base image is converted to black and white and processed using a manually set adaptive threshold in MATLAB, dependent on the colors in the original work. Image trace vectorization is done in Adobe Illustrator with manually set parameters. Optional color inversion, artifact removal, color reassignment, and fill/stroke inversion are done manually in Adobe Illustrator. Paths are grouped by color and exported to absolute-position SVG path elements with zero stroke width, then converted to parametric equations using a pre-existing library [14].

B. Drones and Flight Space

Individual agents are Bitcraze Crazyflie 2.1+, equipped with a Crazyflie 2.x LED deck. Four drones were used for producing the results in Fig. 4., but in testing up to eight agents were flown without issues. The system is scalable, only limited by the computational requirements of the solver and plane assignment. The practical flight space is roughly 4 m × 4 m × 2.15 m with tracking provided by VICON motion capture cameras, bounded by ground effect forlow-z positions and tracking field-of-view for high-z positions. Long exposures were recorded on a Google Pixel 8 smartphone with a 0.55x IMX386 sensor.

C. Waypoints and Collision Avoidance

The parametric equations are ingested into a Python script, and RGB colors reassigned based on the equations, still grouped by color. Colors should generally have greater than 50 lightness on the HSL scale to ensure a suitable LED brightness on the Crazyflie. Discontinuities in parametric equations are defined as jumps greater than 0.05 meters. The size of the canvas is defined by the size of the Illustrator artboard and is scaled to the flight space. When using drones in separate planes, a separation of 0.6 meters or greater is ideal to avoid potential downwash effects between two Crazyflies. Step distance is adjusted based on the piece and is closely related to the rate of commands sent to the Crazyflies.

D. Real-World Flight Parameters

Some previous light painting work, including the SXSW demo, was done with the low-level Commander framework [15] and using waypoints as setpoints communicated to the Crazyflies at a specific frequency. However, several disadvantages were observed when testing with the low-level framework at 20 Hz. In low-level, Crazyflies did not confirm reaching the waypoints, causing waviness in smooth curves or straight lines. Additionally, because a command must be sent to every drone in every instance, additional commands were required to hover drones after completing paths. When several drones are used, it is possible for commands to be lost to radio interference, leading to flight instability or crashes.

Instead, the high-level Commander framework, which generates intermediate setpoints for complex movements and handles command streaming, was used. While this sacrifices some of the finer-grain control provided by the low-level framework, it is traded for much smoother and consistent lines and more stable flight. Crazyflies were commanded between waypoints at 2Hz, the experimental maximum high-level frequency. High-level control also allows for a more variable setpoint distance, which gives control over the velocity. Flight points are smoothest when paths are all flown at a set theoretical velocity, though complex curves with many direction changes can cause lower actual velocities.

For discontinuities, which indicate repositioning between paths, drones will turn off their LED ring. Additionally, different drones have different total flight times. Drones turn off their LEDs and hover in place after completing their paths to avoid image artifacts from landing. After all drones are finished, a land command is issued to all drones.

V. RESULTS AND EVALUATION

Two pieces were flown using the described methods: *Composition II in Red, Blue, and Yellow* and *The Starry Night.* Both works were replicated successfully. Simpler works, such as *Composition II in Red, Blue, and Yellow*, require no additional accommodations. However, *The Starry Night* especially required additional downscaling to ensure successful flight with complex paths. Additionally, light paintings are constrained by the flight time of the Crazyflies themselves, around about three minutes. For *The Starry Night*, three flights of just under three minutes were needed.

VI. STRENGTHS AND WEAKNESSES

While the image processing retains critical details well, a downside to generalization is that the artistic style of the original work can be lost. In particular, solid colored regions are omitted in line work, and this presents a stylistic problem when the fill itself is perhaps the draw of the work. Much resolution is lost especially during the image trace and artifact removal stages in order to generate paths which characterize the overall work. However, textures such as the flowing lines in *The Starry Night*, which are within the image trace regions, are significantly reduced in the final product.

Along similar lines, the current implementation is artificially limited to one color per drone, as LED colors are reassigned per drone file. However, each waypoint can be assigned its own color, allowing a single drone to fly multiple single-color paths or even multi-color paths.

While the solver and plane assignment are incredibly comprehensive and can both theoretically present optimal solutions, the time complexity of both can increase significantly with an increase in agents.

The plane assignment, in particular, is a brute force method, testing every possible combination. With an increase in agents or planes, the computation becomes incredibly expensive, even though it is done in pre-planning. Possible methods for addressing this problem include accounting for symmetry of configurations and calculating and eliminating poorly performing combinations of planes and drones at a lower level, such as in pairs, before attempting to find an optimal configuration. Heuristics based on factors such as the area of the canvas covered could also prove useful.

There is currently no method in place for optimizing the order in which paths are assigned or flown by drones. Such a method could significantly reduce flight time and introduce another variable to consider in the solver. Optimization would be an extension of the traveling salesman problem, based on the start and end of paths.

VII. APPLICATIONS AND ETHICS

A. Other Applications

Fundamentally, the methods demonstrated here allow for the conflict-free execution of trajectories when constrained by the need to travel specific paths themselves, in various applications. One potential such application is surveying with heterogeneous drone swarms. Given different critical areas to be surveyed by certain drones with a certain sweeping pattern, a modified CBS-style solver such as the one presented here could be applied.

B. Ethical Implications

Any platform involving drones becomes a tool which can be used in a wide range of scenarios. The methods presented here could be adapted for surveillance as easily as for agricultural surveying. Light shows and painting in particular can be used to generate publicity for any company or brand [16]. Ultimately, the intent of Fireflie itself is in the application of light painting, and should be for artistic purposes.

Adaptation of existing works of art, especially in this research application, fall under fair use. However, contemporary works could be more protected. In the United States, *The Starry Night* is in the public domain, while *Composition II in Red, Blue, and Yellow* remains copyrighted as of 2024 by the Mondrian/Holtzman Trust [17]. Copyright protections of works should be considered depending on the application of the methods presented.

VIII. SUMMARY

Fireflie successfully generates recognizable light paintings for two iconic works, *Composition II in Red, Blue, and Yellow* and *The Starry Night*. The methods provided for image processing and swarm flight are generalizable to any source image, even if minor manual inputs are required to ensure preservation of key motifs. Successful collision avoidance is provided through the CBS-inspired solver and plane assignment, which generates optimal solutions. While there remain many directions for expansion, Fireflie provides a very robust foundation for future work in complex multi-agent light painting.

APPENDIX

Flight video for *Composition II in Red, Blue and Yellow* can be found at https://youtu.be/YRhMTGVuPsk. Compiled flight video using DaVinci Resolve for *The Starry Night* can be found at https://youtu.be/PkYptS-8j8k.

ACKNOWLEDGMENTS

I would like to acknowledge Eric Ewing for his guidance in program design and general Crazyswarm knowledge, Narek Harutyunyan for significant assistance in troubleshooting, Lucian Sharpe and Thalia Bonas for inspiring through their collaboration on PointillistQuad, and Professor Nora Ayanian for her tremendous support of my pursuits in robotics.

REFERENCES

[1] B. Barret. “Lady Gaga's Halftime Show Drones Have a Bright Future.” WIRED. https://www.wired.com/2017/02 /lady-gaga-halftime-show-drones/.

[2] E. Zhan. “7,598 drones set new world record with stunning aerial display.” Guinness World Records. https://www.guinnessworldrecords.com/news/commercial/2024/9 /7598-drones-set-new-world-record-with-stunning-aerial-display.

[3] J. A. Preiss, W. Honig, G. S. Sukhatme and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3299-3304, doi: 10.1109/ICRA.2017.7989376.

[4] R. Z. Peng. “PointillistQuad.” https://ryanzpeng.com/pointillist

[5] R. Klimas. “How to Light-Paint Giant Shapes in the Sky with a Drone.” PetaPixel. https://petapixel.com/2019/02/04/ how-to-light-paint-giant-shapes-in-the-sky-with-a-drone/

[6] Wikimedia Commons. File:Van Gogh - Starry Night - Google Art Project.jpg. https://commons.wikimedia.org/wiki/ File:Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg

[7] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, “Conflict-based search for optimal multi-agent pathfinding,” Artificial Intelligence, vol. 219. Elsevier BV, pp. 40–66, Feb. 2015. doi: 10.1016/j.artint.2014.11.006.

[8] C. Dubois, “Sparked,” ACM SIGGRAPH 2015 Computer Animation Festival. ACM, pp. 140–140, Jul. 31, 2015. doi: 10.1145/2745234.2792988.

[9] A. S. Vempati et al., “PaintCopter: An Autonomous UAV for Spray Painting on Three-Dimensional Surfaces,” IEEE Robotics and Automation Letters, vol. 3, no. 4. Institute of Electrical and Electronics Engineers (IEEE), pp. 2862–2869, Oct. 2018. doi: 10.1109/lra.2018.2846278.

[10] A. Uryasheva, M. Kulbeda, N. Rodichenko, and D. Tsetserukou, "DroneGraffiti," ACM SIGGRAPH 2019 Studio. ACM, Jul. 28, 2019. doi: 10.1145/3306306.3328000.

[11] V. Serpiva, E. Karmanova, A. Fedoseev, S. Perminov, and D. Tsetserukou, "DronePaint: Swarm Light Painting with DNN-based Gesture Recognition," Special Interest Group on Computer Graphics and Interactive Techniques Conference Emerging Technologies. ACM, pp. 1–4, Aug. 05, 2021. doi: 10.1145/3450550.3465349.

[12] E. Ewing, E. Blanco. swarms@sxsw. https://www.youtube.com/watch?v=NLoQcquP-DE

[13] Wikimedia Commons. File:Piet Mondriaan, 1930 - Mondrian Composition II in Red, Blue, and Yellow.jpg. https://commons.wikimedia.org/wiki/File:Piet_Mondriaan,_1930_-_Mondrian_Composition_II_in_Red,_Blue,_and_Yellow.jpg

[14] maltaisn. svg-equations (Version 1.1.0). https://github.com/maltaisn/svg-equations/releases

[15] Bitcraze. "The Commander Framework." https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/sensor-to-control/commanders_setpoints/

[16] Celestial Labs. "Elevate Your Brand with an Iconic Drone Show." https://celestial.show/brand-storytelling-drone-shows/

[17] Mondrian/Holtzman Trust. "Reproduction Rights." https://www.mondriantrust.com/reproduction-rights/