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

CLIPSwarm: Generating Drone Shows from Text Prompts with Vision-Language Models

Pablo Pueyo, Eduardo Montijano, Ana C. Murillo and Mac Schwager

***Abstract*— This paper introduces CLIPSwarm, a new algorithm designed to automate the modeling of swarm drone formations based on natural language. The algorithm begins by enriching a provided word, to compose a text prompt that serves as input to an iterative approach to find the formation that best matches the provided word. The algorithm iteratively refines formations of robots to align with the textual description, employing different steps for "exploration" and "exploitation". Our framework is currently evaluated on simple formation targets, limited to contour shapes. A formation is visually represented through alpha-shape contours and the most representative color is automatically found for the input word. To measure the similarity between the description and the visual representation of the formation, we use CLIP [1], encoding text and images into vectors and assessing their similarity. Subsequently, the algorithm rearranges the formation to visually represent the word more effectively, within the given constraints of available drones. Control actions are then assigned to the drones, ensuring robotic behavior and collision-free movement. Experimental results demonstrate the system's efficacy in accurately modeling robot formations from natural language descriptions. The algorithm's versatility is showcased through the execution of drone shows in photorealistic simulation with varying shapes. We refer the reader to the supplementary video for a visual reference of the results.**

I. Introduction

Foundation models, including large language models, image and video generation models, and vision-language models, have revolutionized the technological landscape due to their ability to generate, manipulate, and translate complex natural language and image data. These models are pre-trained on vast amounts of diverse data and can perform a wide array of language-related tasks, including text generation [2], translation, and more. Their adaptive nature and applicability show their growing impact, redefining how we interact with technology. This impact can be observed in several key areas such as content creation, customer support, software development tasks, and scientific research. In this last realm, foundation models find application in robotics. Existing solutions propose different techniques to control actuators using natural language or giving commands to robots to perform different actions (e.g. move to the room on your left) [3]. Nevertheless, to the best of our knowledge, there are no prior works in the literature that use foundation models to control a swarm of robots. In this work, we propose to use a vision-language model, CLIP, to drive a swarm of robots to a target formation that matches a text description.

This work was supported by DGA project T45_23R, MCIN/AEI/ERDF/European Union NextGenerationEU/PRTR project PID2021-125514NB-I00 and by PID2022-139615OB-I00/MCIN/AEI/10.13039/501100011033/FEDER-UE.

P. Pueyo, E. Montijano and A. C. Murillo are associated with the Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain {ppueyor, emonti, acm}@unizar.es

M. Schwager is associated with Dept. of Aeronautics and Astronautics, Stanford University, USA {schwager}@stanford.edu

Text Prompt: *"Leaf"*

Fig. 1. **Drone formation automatically crafted to match a given text**. CLIPSwarm takes a single word describing a shape as input and automatically determines the color and positions of a robotic swarm formation that best fits the given text. The example illustrates the shape created by a formation of 30 robots. The drones move to positions that collectively form a shape corresponding to the word *"Leaf"*. Left: graphical representation of the shape formed by the robot formation. Right: formation of drones as part of a show in a photorealistic simulation representing the input word.

On the other hand, *Artistic robotics* has emerged as a promising field in recent years for both the general audience and the robotics community. The main focus of this trend is using robots to express or design art in various manners, such as the aim to transform robots into painters [4], sculptures [5], dancers, or even cinematographers [6].

One of the latest trends in robotic arts involves the use of a team of robots or drones that collaboratively create artistic shapes, which is the main focus of this work. Such "drone displays", where drones function as pixels and coordinate to create visually appealing shapes in the sky, have become common features of major public sporting and celebration events (e.g. the Olympics Opening Ceremonies, civic New Year's celebrations) with several companies focused on producing these artistic shows. Existing solutions in the literature need the interaction of a person to manually design the shape that the robots should form [7].

Complementing the existing literature on the aforementioned topics, the primary objective of this project is to utilize an existing *foundation model* for shaping a swarm of robot formations with *artistic intent*. CLIPSwarm paves the way and is the first step to creating robot formations autonomously. As shown in Fig.1, the user introduces a description in natural language of the desired shape of the formation, e.g. *"leaf"*, and CLIPSwarm automatically decides the position and color of the robots of the formation to form a shape and a drone formation that corresponds

with that description, so the users do not need to create the patterns beforehand. To do so, we use CLIP [1], a multimodal foundation model that is trained to encode texts and images, finding similarities between them, and calculating what is referred to as CLIP Similarity. The system takes as input a text prompt describing a desired shape, and the output comprises the positions of the robots to best represent that shape, along with the color that is best suited to the prompt.

To achieve this goal, our proposed method begins with an engineered prompt. The algorithm selects the most representative color for the input text and elaborates an enhanced prompt to provide additional information to CLIP. Then, we execute an iterative algorithm to choose the best formation describing the introduced text, improving the CLIP Similarity across iterations. As the last step, the system identifies the most representative robot positions that form the desired shape and their corresponding colors. As output, our method generates the robot locations and color, which are then fed to a lower-level distributed formation control system to drive the robots to their goal positions while avoiding collisions. This user-friendly approach facilitates the process for users, eliminating the need to pre-create patterns, and offers a dynamic and efficient means of translating artistic descriptions into robotic formations.

Our method is validated with extensive experiments. We first analyze how the proposed algorithm increases the CLIP similarity obtained through iterations, and many qualitative examples showcase how the shape achieved by the formations matches the input text. We also run a simulated drone show to represent the process of creating several figures in photorealistic simulation, employing a navigation algorithm to avoid collisions between robots and giving realistic robotic dynamics to the drones. The video is included as part of the supplementary material. The presented experiments demonstrate CLIPSwarm's applicability to real robotic systems, and its ability to create swarm formations of robots that correspond to given descriptions.

II. Related work

Foundation models, such as large-scale language models, have seen widespread adoption in various fields, including robotics. Researchers have been exploring ways to integrate natural language understanding and generation capabilities into robotic systems [8]–[10]. Some applications include the utilization of multimodal foundation models to control manipulators [3], direct actions [11], or design a plan of actions to a robot performing a particular task [12]. Certain works integrate foundation models with reinforcement or imitation learning techniques, achieving human-like behavior in robots [13], [14]. One of the latest solutions employs ChatGPT [2] to directly generate Python code capable of controlling heterogeneous robots to perform various actions as commanded in natural language [15]. While existing solutions represent a significant leap in the application of foundation models for robotics, CLIPSwarm stands out as the first solution utilizing a foundation model to control a cooperative swarm of robots with artistic purposes without the need for retraining or fine-tuning any model.

Artistic robotics is an emerging research area in robotics that is gaining attention from both the general audience and the robotics community. Various works explore the transformation of robots into painters, with techniques ranging from replicating existing styles of real painters [4], [16] to preprocessing input images for simplified robot input [17], or even autonomously creating graffiti [18]. Some works focus on robots sculpting without human interaction [5]. Artistic expressions extend beyond sculpture or painting; for example, some solutions involve robots acting as professional dancers [19] or serving as autonomous cinematographers. In this latter application, robots autonomously record cinematic scenes, satisfying artistic or technical details [6], [20], [21].

In the realm of multirobot artistic robotic formations, some works employ optimal control to move 2D formations of robots forming a set of given patterns, [22], [23]. The same authors incorporated an interface to draw the desired pattern in [24]. More recent approaches address how to perform multidrone formations shows, [7], [25]–[27]. All these works receive the desired pattern as input. Recent works generate 3D shapes using diffusion models [28] or GANs [29] but the behavior of the particles, which do not explicitly consider robotic behaviours and constraints, can make it difficult to adapt to a robotic extent.

In contrast to the aforementioned approaches, CLIPSwarm allows users to give simple instructions to the system in natural language, without the need to create patterns beforehand. This is achieved thanks to the use of Foundation Models, eliminating the need for extensive datasets or time-consuming retraining. The platform automatically determines the best positions and color for the team of robots to represent the formation described by the text input. It then provides an approach to performing a drone show, considering the dynamics and potential collisions inherent in a robotic swarm system.

III. Solution

Our solution is organized into three distinct modules, as illustrated in Fig. 2. The first module enriches the input word by incorporating additional details to create a more accurate text prompt. The second module employs an iterative algorithm to enhance the *CLIP Similarity* between the prompt and the images associated with a set of formations. This metric is formulated using CLIP [1], a foundational model trained on an extensive dataset of text-image pairs. We denote by $\text{CS} : t \times \mathbf{I} \rightarrow [0, 1]$ the CLIP similarity function, that returns a positive score measuring the similarity between an input text t and an image $\mathbf{I}$. This function achieves higher values when the text and image pair describe the same concept or idea. Finally, the third module adapts the output formation and is in charge of assigning goal positions to the drones and move them avoiding obstacles. The three modules are detailed in the following subsections.

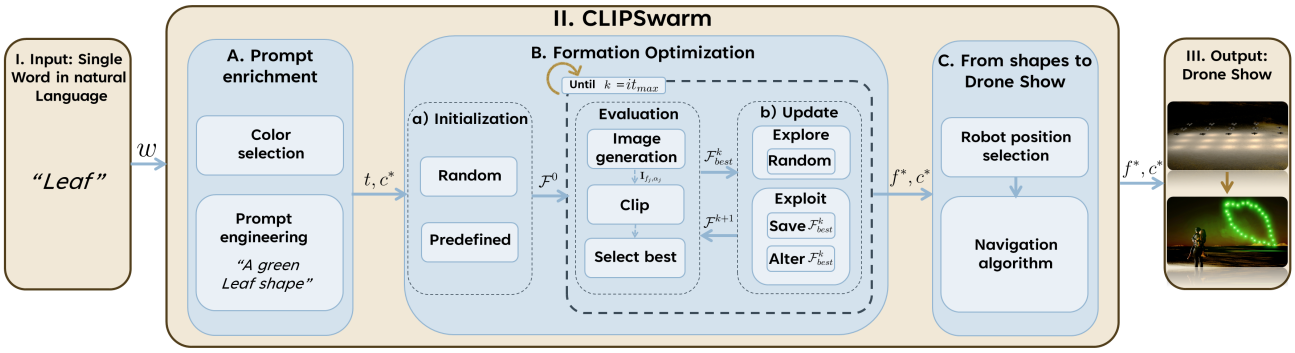

Fig. 2. **CLIPSwarm algorithm diagram**. A schematic summary of the platform.*I. Input*, a word describing the desired formation. *II. CLIPSwarm algorithm*, including the three modules of the system. *(A) Prompt enrichment*, involving color selection and prompt engineering to enrich the input word and form a text. *(B) Formation Optimization*, steps to select the formation that best describes the input text. *(a) Initialization*. A set of formations (consisting of robot positions) are randomly sampled from a uniform distribution. Some predefined shapes are added as part of a 'warm start'. *Evaluation*. The formations are converted to images. Then, CLIP extracts the similarity between the images and the input text, and the formations with the best similarities are selected. *(b) Update*. New formations are iteratively created employing an "exploration-exploitation" strategy, improving the CLIP similarity across iterations. *(C) From shapes to drone show*. The positions of the obtained formation are optimized through robot position selection and a navigation algorithm. *III. Output* is drone positions to perform a drone show by moving and selecting the color of the drones representing a shape described by the input word.

A. Prompt enrichment

The input of the system is a single word, denoted by w. This first module finds the most representative color c and composes an enriched text, t, for more detailed information on the desired shape. After conducting various tests and comparing different additional words, we observed that the CLIP Similarity is higher when certain words are added to the input text. Specifically, we perform prompt engineering by enriching the input text to describe both a shape and its corresponding color, such as "A green leaf shape".

To accomplish this, we start by generating a set of 10 images, $\mathcal{C} = \{\mathbf{I}_1, \ldots, \mathbf{I}_{10}\}$, with all their pixels of a single basic color[1]. We select the color with the highest CLIP Similarity when compared to the input word w,

$$c^* = \arg \max_{c \in \{1,\ldots,10\}} \text{CS}(w, \mathbf{I}_c), \tag{1}$$

considering it as the most representative for the input word. Subsequently, we add the remaining words to construct a text prompt t in the following manner: $t =$ "A c^* w shape". This text t and the color c^* are then inputted into the second module of our solution for further processing.

B. Formation optimization

Let M be the number of robots available to create the formation. A formation is defined by the positions of all the robots in the image plane, $f = \{\mathbf{p}_1, \ldots, \mathbf{p}_M\}$, where $\mathbf{p}_i \in \mathbb{R}^2$ is the position of the i-th robot of the formation. The goal is to find the formation, f^*, that maximizes the CLIP similarity of t and the image associated with the formation.

1) Image generation: Since a robot formation is a sparse collection of points, a fundamental element in our solution is how we generate the images to compute the CLIP similarity. In this work, images are crafted by drawing a concave contour over the robots positioned on the edge of the formation using the Alpha-Shape algorithm [30]. The algorithm receives a set of points, i.e., a formation, f, and a real number, α, as inputs and returns a graph, G, where the nodes are the points and the edges depend on α. Particularly, for $\alpha = 0$ the algorithm returns a graph with the convex hull of the formation. As α increases, the figure associated with the graph presents a higher concavity, by including more of the formation points in the contour. To obtain a single polygon, the parameter can increase until a limit value, α_f, depending on f, where the contour contains all the points in the formation (Figure 3). Bigger values of α than α_f will return a graph with more than one connected component.

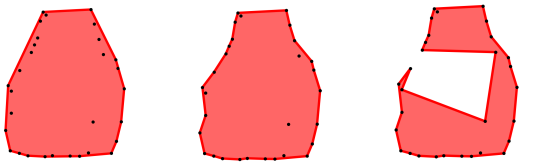

Fig. 3. **Influence of Alpha Value on Contour:** Alpha shapes representation of a robot formation with varying alpha values. Different alpha values produce distinct representations of the contour. When the alpha value is zero, the contour forms a convex hull. Larger alpha values result in a more finely detailed and intricate contour. Each formation has a maximum value of alpha (α_f) that ensures all points are inside a single polygon while the contour has the maximum concavity.

Our images are generated by filling the polygon associated with G of the color c^*. For simplicity in the notation, we denote by $\mathbf{I}_{f,\alpha}$ the image obtained using this procedure for the formation f and the parameter α. Figure 7 in the experiments show different examples of such images.

2) Iterative optimization: Once we are able to compute images from formations, we can use them to compute the best one in terms of the CLIP similarity. Particularly, we propose an optimization algorithm where we iteratively refine a set of N potential formations and α parameters, based on the CLIP similarity of their corresponding images with

[1]red, orange, yellow, green, cyan, blue, purple, magenta, pink, brown

respect to t. Let

$$\mathcal{F}^k = \{\mathcal{F}_1^k, \ldots, \mathcal{F}_N^k\}, \tag{2}$$

be the formation set at iteration k, where $\mathcal{F}_j^k = \{f_j^k, \alpha_j^k\}, j = 1, \ldots, N$, is a formation and α-parameter duple. Therefore, our process iteratively finds

$$\{f^*, \alpha^*\} = \arg \max_{\mathcal{F}_j = \{f_j, \alpha_j\} \in \mathcal{F}^k} CS(t, \mathbf{I}_{f_j, \alpha_j}). \tag{3}$$

a) Initialization: The formation set is initialized by a combination of multiple random formations (rnd) and predefined shaped formations (Δ). This initial set of formations is termed as *'Initialization Pool'*, or $\mathcal{F}^0$,

$$\mathcal{F}^0 = \mathcal{F}_{\text{rnd}} \cup \mathcal{F}_{\Delta}. \tag{4}$$

In the random formations, the position of all the robots is initialized using a uniform distribution within the boundaries of the image space. Besides, to facilitate a 'warm start' for the optimization algorithm, five predefined basic shapes (rhombus, triangle, inverted triangle, hexagon, square), along with p variations of these shapes (by adding some noise to the positions of the robots) are added to the set. The parameter α is constant and equal for all the formations during the initialization. Figure 4 shows some examples of these predefined formations along with some random shapes (column on the right) for clarification.

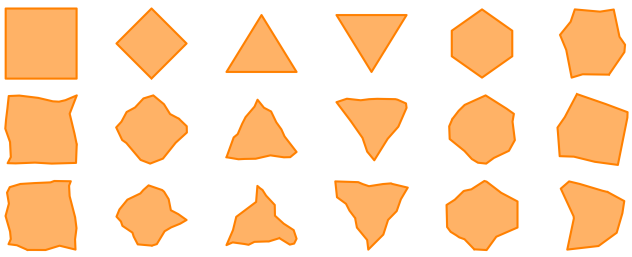

Fig. 4. **Predefined shapes**. Columns 1-5 display predefined shapes along with random variations of them, which are added to the initialization pool as a 'warm start' during the Initialization stage. Column 6 shows some random samples from the initialization pool for comparison.

b) Update: The formation set is refined at each iteration based on an 'exploration-exploitation' technique, discarding formations with low CLIP similarity and introducing new formations based on random locations (exploration) and different variations of those with good scores in the past (exploitation). In particular,

$$\mathcal{F}^{k+1} = \mathcal{F}_{\text{best}}^k \cup \tilde{\mathcal{F}}_{\text{best}}^k \cup \mathcal{F}_{\text{rnd}}^{k+1}, \tag{5}$$

where $\mathcal{F}_{\text{best}}^k$ are the best b formations at iteration k, according to the CLIP similarity of their images, $\mathcal{F}_{\text{rnd}}^{k+1}$ is a new subset of random formations, created analogously to the initialization step, and $\tilde{\mathcal{F}}_{\text{best}}^k$ are variations over the best set that we describe next.

We implement four different variations over $\mathcal{F}_{\text{best}}^k$, incorporating a mix of aggressive and smooth modifications. The resulting formations are incorporated into the new set of formations for the next iterations,

$$\tilde{\mathcal{F}}_{\text{best}}^k = \mathcal{F}_{\text{subd}} \cup \mathcal{F}_{\text{one}} \cup \mathcal{F}_{\text{contour}} \cup \mathcal{F}_{\alpha}. \tag{6}$$

The first set (subd) involves dividing the map into four halves (top, bottom, left, right). The positions of the robots within each subdivision are altered randomly, following a uniform distribution within small boundaries. The second set (one) considers smooth alterations, generated by moving a single robot for all the best formations. The third modification (contour) of the formations entails relocating equally all robots to the contour of the shape, introducing a slight amount of noise for further variation. Finally, in $\mathcal{F}_{\alpha}$, for each formation in $\mathcal{F}_{\text{best}}^k$, we include a new element where we replace the default value of α by the limit value α_f of that formation, leading to the creation of distinct contour drawings in the next iteration.

Once the set is complete, a new iteration begins. The process is repeated for a fixed number of iterations, finally returning the best formation, Eq. (3), along with the values of α and c^*.

C. From shapes to drone shows

As final step, we need to create the final 3D positions for the drones, decide which drones go where, and move them safely avoiding collisions.

At this stage, we focus only on the contour of the best formation, f^*, obtained using the associated value of alpha, α^*. The contour is divided into M equal and joint segments, placing one robot at the extremes of them (Figure 5). Since the formation is in 2D and the drones need to move in 3D, we reproject the formation to be at a fixed distance and height with respect to the point of view of the spectators.

We consider that initially the drones are equally spaced in the ground. To decide the position in the formation that each drones needs to reach, we use the Hungarian algorithm, minimizing the total distance traveled by the swarm to reach the target formation and obtaining an assignment with low probability of collision. Finally, to control the drones we use the well-known Optimal Reciprocal Collision Avoidance (ORCA) algorithm for 3D vehicles [31]. We note that this process can be repeated with additional formations without the need to start from the ground if needed.

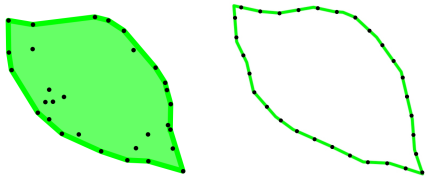

Fig. 5. **Postprocessing step to determine the position of the robots**. The postprocessing step determines the positions of the robots to meaningfully represent the given shape. On the left are the end positions of the 30 robots calculated by the second step of the algorithm. On the right, the postprocess step of the algorithm distributes equally the same number of robots to better represent the same shape.

IV. Experimental Validation

This section demonstrates CLIPSwarm's ability to generate formations that match a provided natural language word, highlighting the applicability of the system to a robotic environment. A first experiment analyzes the proper behavior of the approach, demonstrating how the algorithm improves CLIP similarity across iterations. Then, we present a set of experiments to run the whole formation generation pipeline given our defined *test-set* of words. Finally, we demonstrate several executions of the system in a realistic drone show simulation using AirSim, to showcase the system's applicability to a realistic robotic environment. The supplementary video provides a more detailed execution of this simulation.

A. Assessing the algorithm

This experiment investigates how our algorithm progressively increases the CLIP similarity obtained across iterations. ensuring the resulting robot formation accurately reflects the target word.

To this end, we define a *test-set* of 50 words [2] representing various shapes. Subsequently, we execute the first two modules of the solution (Sec. III-B) for the *test-set*. After conducting several tests, we noticed that the CLIP Similarity metric stops improving after a certain number of iterations. Therefore, for this experiment, we opted for an 'early-stopping' strategy and concluded the training at 15 iterations. Formations were generated using $M = 30$ robots, and parameters were configured to produce $N = 500$ formations in each iteration.

For all tests, we analyze how the CLIP Similarity keeps growing as our algorithm runs more iterations. In particular, we compute the average percentage of improvement AoI as

$$P_w = \frac{max(\mathcal{S}_w^{it_{max}}) - max(\mathcal{S}_w^1)}{max(\mathcal{S}_w^1)}, \tag{7}$$

$$AoI = avg(P_w) \quad \forall w \in \textit{test-set}, \tag{8}$$

where S_w^k represents the list of CLIP Similarities obtained for all formations corresponding to a particular word w at iteration k. P_w denotes the percentage of improvement, comparing the best CLIP Similarity achieved in the last iteration with that of the first iteration for the word w.

A graphical representation of the results of this test after executing the solution for the 50 words of the *test-set* is depicted in Fig.6. The solid line illustrates the average percentage of improvement across iterations relative to the initial maximum CLIP similarity. At the end, the AoI for all words is 10.15% with respect to the first iteration. The lighter-shaded area represents the standard deviation. The dashed lines at the top and bottom indicate the scenarios where CLIP similarity demonstrates the most improvement across iterations and the worst improvement (meaning that the initial shape is a good fit for the word), respectively.

[2]apple, avocado, balloon, banana, bear, bicycle, bird, boat, book, bottle, butterfly, car, cat, chair, cherry, cloud, coin, cup, diamond, drop, fish, flower, hat, heart, house, key, kite, leaf, lemon, lighthouse, lightning, moon, mushroom, orange, pear, plane, puzzle, raindrop, robot, rocket, shoe, spoon, star, strawberry, sun, tomato, train, tree, wave, watermelon

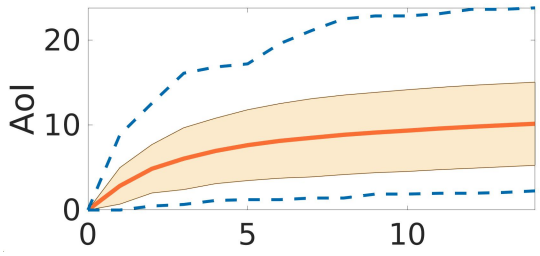

Fig. 6. **Average of improvement of CLIP Similarity** Solid orange line represents the average improvement of CLIP across iterations for our test set composed of 50 words. Light-shaded area indicates the standard deviation of the average improvement. Top dashed line represents a particular word where the average improvement increases the most across iterations. The bottom dashed line represents a particular word where the average improvement increases the least across iterations, possibly due to a good seed.

TABLE I
Best Similarity Metric/Iteration for Shapes in the Drone Show

	Cat	Lemon	Apple	Raindrop	Heart	Leaf
it=1	0.250	0.305	0.245	0.265	0.270	0.310
it=7	0.278	0.329	0.284	0.292	0.280	0.317
it=15	0.281	0.334	0.297	0.296	0.302	0.325

B. Modeling formations from a word

This section showcases formations generated by the algorithm in response to an input word and their evolution over successive iterations. For this experiment, we chose certain shapes from the outcomes of the previous experiment, after running it on the *test-set*. These shapes will be utilized for the drone show in the subsequent experiment. Figure 7 illustrates the results of this experiment. Each row corresponds to one test with one word. It depicts the representation of the formation with the highest score at various iterations of the algorithm, in response to the introduced word marked as "User input". The algorithm dynamically enriches the prompt, shown in the figure as "Enriched input". The initial figure in each row displays a representation of the initialization pool that is used as a seed in the Initialization step and just composed by random shapes.

For quantitative results, Table I shows the best CLIP Similarity for each case for the initial, intermediate, and final iteration. This metric increases as the iterations progress, indicating the solution's ability to obtain formations that match closer and closer the given description.

C. Drone Show in photorealistic simulation

This section emphasizes the adaptability of the system to a realistic robotic environment, highlighting the effect of the postprocessing stage of our algorithm (Sec.III-C). To demonstrate this, we conducted a drone show in the photorealistic simulator AirSim [32], [33], offering a scenario closer to reality. To specify the color of the drones, we equipped them with a light whose color can be controlled through the AirSim API. The connection between CLIPSwarm and AirSim is implemented in ROS [34], facilitating the potential transition of the solution to a setup with real drones.

We employed a Python adaptation [35] of ORCA for 3D formations to send control commands to the drones. This

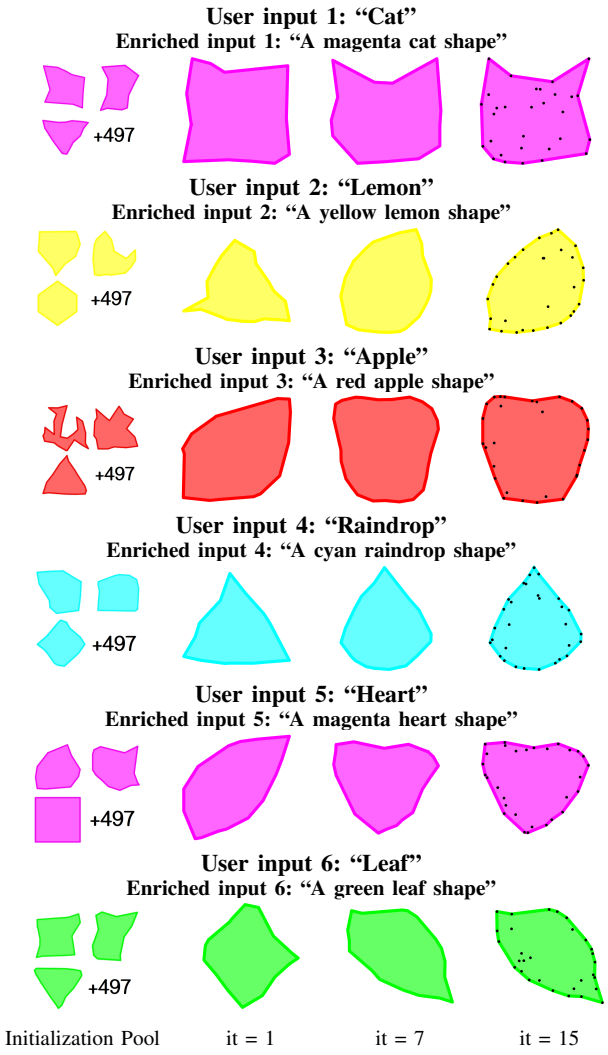

Fig. 7. **From a Single Word to Shapes.** Shapes crafted from formations of 30 robots. The first column displays the representation of the initialization pool used as the seed in the Initialization step, composed of random and predefined shapes for a ‘warm start’. Subsequent columns show shapes with the highest similarity in different iterations (1, 7, 15) of the algorithm, given the input text and enriched description detailed in the title of each row. The fourth column represents the formation with the highest similarity after 15 iterations, also chosen to be used in the drone show.In this case, the positions of the robots of the formations are depicted for comparison with their positions in Fig. 8, after the equal distribution of robots along the contour, as shown in Fig. 5.

algorithm ensures that the drones progress towards the goals in smaller steps to reach the final position, avoiding collisions between them.

The stages of the drone show in photorealistic simulation are depicted in Fig. 8. The shapes are automatically obtained using CLIPSwarm. The drone show replicates the shapes obtained in the last iteration of the algorithm, as shown in the fourth column of Fig. 7. In comparison with these formations, the postprocessing stage distributes the drones equally along the contour and reprojects the shapes from 2D to 3D, as described in Section III-C. The figure displays the initial positions of the drones (top left). An intermediate stage of the drone show is displayed (bottom left) to represent the transition from one shape to another. The postprocessing

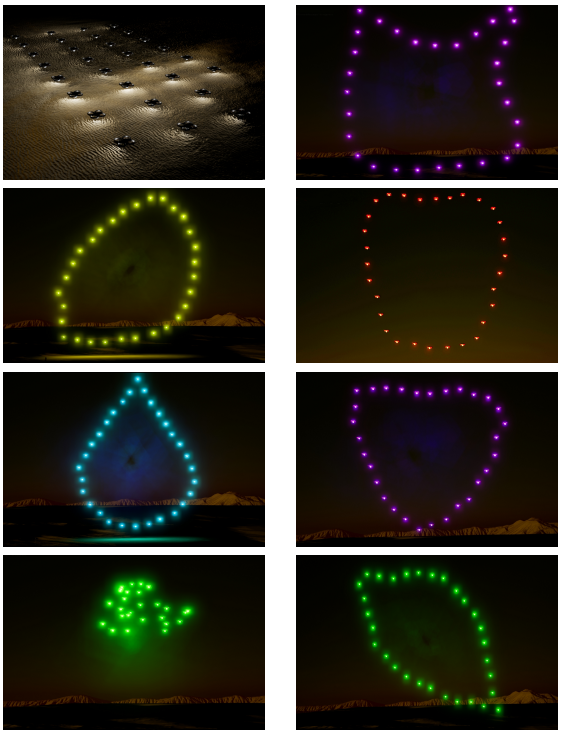

Fig. 8. **Drone Show in photorealistic simulation**. The shapes have been automatically created in earlier stages of the algorithm when the next words from the *test-set* are inputted: [‘cat’, ‘lemon’, ‘apple’, ‘raindrop’, ‘heart’, ‘leaf’]. The show is performed in a photorealistic simulation (AirSim) with 30 drones. The initial stage is depicted on the first row on the left. An intermediate step is depicted in the bottom row on the left to demonstrate the ability to respect dynamics and avoid collisions between drones during shape transitions. Shapes arranged chronologically from left to right.

algorithm ensures drone dynamics and collision avoidance. We direct the reader to the supplementary video for a complete visualization of the drone show, showcasing all the transitions and shapes.

V. Limitations

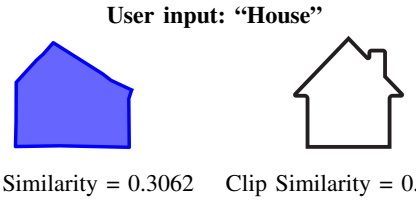

Fig. 9. **Limitations of the algorithm**. On the left, the shape obtained by our algorithm as a response to the input text “House”. On the right, a stock example of a contour of a house. The contour simplifies the evaluation process but may not capture the details of the shape. Additionally, relying on the CLIP Similarity metric may result in a lower score for formations that better correspond to the given text for an average user.

For the sake of simplicity, we have decided to evaluate the formations based on the CLIP Similarity of the Alpha-shape contour of the formations. This simplification allows for a faster evaluation process and works well with a relatively low number of robots. However, the variety of shapes that can be modeled with just a contour is limited. Moreover, our current algorithm relies significantly on CLIP, and increasing the similarity does not always mean that the shape is closer to what an average user would expect, as CLIP is not

specifically trained with contours.We illustrate this with an example in Fig. 9, where the algorithm fails to capture all the expected details that would represent a house. In this case, the CLIP Similarity of the picture on the left is higher than the one on the right, even though the latter may seem closer to a *house* for an average reader, showcasing a limitation of using this CLIP as a similarity metric. Future steps will include working with more complex inputs as well as using additional metrics that complement the CLIP Similarity.

VI. Conclusions

In this paper, we introduced CLIPSwarm, which is designed to create automatically drone formations that represent a given word in natural language. We have explained how we enriched the word provided and engineered a corresponding text prompt. This text is then used by an iterative algorithm that employs an 'exploration-exploitation' technique to derive a formation of robots that aligns with the description in the given text. We used CLIP to encode the text and the images into vectors to measure the similarity between the description and the image of the formations. The formation is then adapted to visually represent the word within the constraints of the available number of drones. Control actions are assigned to the drones to guide them to their positions, ensuring robotic behavior and collision-free movement. Our experiments have shown that the algorithm improves the clip similarity across iterations. Additionally, the system can model robot formations from natural language and be applied to performing drone shows featuring different shapes and colors. In the future, we will explore a wider variety of formations, including more complete shapes and the optimization of 3D formations.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Int. Conf. on Machine Learning*, 2021, pp. 8748–8763.

[2] "Chatgpt," https://openai.com/blog/chatgpt, accessed: 2024-02-29.

[3] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Conference on Robot Learning*, 2023, pp. 540–562.

[4] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, "Non-photorealistic rendering techniques for artistic robotic painting," *Robotics*, vol. 8, no. 1, p. 10, 2019.

[5] Z. Ma, S. Duenser, C. Schumacher, R. Rust, M. Baecher, F. Gramazio, M. Kohler, and S. Coros, "Stylized robotic clay sculpting," *Computers & Graphics*, vol. 98, pp. 150–164, 2021.

[6] P. Pueyo, J. Dendarieta, E. Montijano, A. C. Murillo, and M. Schwager, "Cinempc: A fully autonomous drone cinematography system incorporating zoom, focus, pose, and scene composition," *IEEE Transactions on Robotics*, 2024.

[7] D. Nar and R. Kotecha, "Optimal waypoint assignment for designing drone light show formations," *Results in Control and Optimization*, vol. 9, p. 100174, 2022.

[8] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[9] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, "Large language models for robotics: A survey," *preptint arXiv:2311.07226*, 2023.

[10] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human-robot interaction: A review," *Biomimetic Intelligence and Robotics*, p. 100131, 2023.

[11] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran, "Can foundation models perform zero-shot task specification for robot manipulation?" in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 893–905.

[12] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9118–9147.

[13] C. Kim, Y. Seo, H. Liu, L. Lee, J. Shin, H. Lee, and K. Lee, "Guide your agent with adaptive multimodal rewards," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration," *arXiv preprint arXiv:2311.12015*, 2023.

[15] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *Microsoft Auton. Syst. Robot. Res*, vol. 2, p. 20, 2023.

[16] A. Beltramello, L. Scalera, S. Seriani, and P. Gallina, "Artistic robotic painting using the palette knife technique," *Robotics*, vol. 9, 2020.

[17] A. Karimov, E. Kopets, G. Kolev, S. Leonov, L. Scalera, and D. Butusov, "Image preprocessing for artistic robotic painting," *Inventions*, vol. 6, no. 1, p. 19, 2021.

[18] 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 *Int. Conf. on Robotics and Automation*, 2022, pp. 4065–4072.

[19] H. Peng, C. Zhou, H. Hu, F. Chao, and J. Li, "Robotic dance in social robotics—a taxonomy," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 3, pp. 281–293, 2015.

[20] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *JFR*, vol. 37, no. 4, pp. 606–641, 2020.

[21] P. Pueyo, E. Montijano, A. C. Murillo, and M. Schwager, "Cinetransfer: Controlling a robot to imitate cinematographic style from a single example," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 044–10 049.

[22] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Multi-robot system for artistic pattern formation," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 4512–4517.

[23] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P.Beardsley, "Image and animation display with multiple mobile robots," *The International Journal of Robotics Research*, vol. 31, no. 6, pp. 753–773, 2012.

[24] S. Hauri, J. Alonso-Mora, A. Breitenmoser, R. Siegwart, and P. Beardsley, "Multi-robot formation control via a real-time drawing interface," in *8th Int. Conf. of Field and Service Robotics*, 2013, pp. 175–189.

[25] M. Waibel, B. Keays, and F. Augugliaro, "Drone shows: Creative potential and best practices," ETH Zurich, Tech. Rep., 2017.

[26] H.-J. Kim and H.-S. Ahn, "Realization of swarm formation flying and optimal trajectory generation for multi-drone performance show," in *IEEE/SICE Int. Symposium on System Integration*, 2016, pp. 850–855.

[27] H. Sun, J. Qi, C. Wu, and M. Wang, "Path planning for dense drone formation based on modified artificial potential fields," in *2020 39th Chinese Control Conference*, 2020, pp. 4658–4664.

[28] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2837–2845.

[29] L. Hui, R. Xu, J. Xie, J. Qian, and J. Yang, "Progressive point cloud deconvolution generation network," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer, 2020, pp. 397–413.

[30] H. Edelsbrunner, "Alpha shapes-a survey," in *Tessellations in the Sciences: Virtues, Techniques, Applications of Geometric Tilings*, 2011.

[31] J. Snape and D. Manocha, "Navigating multiple simple-airplanes in 3d workspace," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3974–3980.

[32] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*, 2018, pp. 621–635.

[33] P. Pueyo, E. Cristofalo, E. Montijano, and M. Schwager, "Cinemairsim: A camera-realistic robotics simulator for cinematographic purposes," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 1186–1191.

[34] A. Koubâa *et al.*, *Robot Operating System (ROS)*. Springer, 2017.

[35] "Orca 3d," https://github.com/mtreml/Python-RVO2-3D, accessed: 2023-03-2.