Semantic-Syntactic Integration for Generative Drone Light Show Design

Nico Posner

Richard Huang

Sarthak Dhanke

Huanlin Dai

Binting Xia

Advisor: Stephen Barry

University of Chicago

Master of Science in Applied Data Science

Research Capstone Project

December 1, 2025

Abstract

This work presents a prototype-level pipeline for co-creative drone light show design that integrates large language models (LLMs) with analytical validation and deployment systems. Rather than treating LLMs as autonomous choreographers, the study reframes them as semantic front ends that generate interpretable creative intent, which is then translated into physically valid and executable drone formations. The proposed framework couples language-based generative interfaces with analytic solvers and the **Skybrush Studio API**, which performs safety verification, trajectory optimization, and compilation into deployable show formats. Through this integration, we demonstrate how semantic creativity and syntactic rigor can coexist within a single workflow, enabling intuitive yet verifiable design of drone swarm performances. The findings clarify where language-driven generation adds value, where it requires analytic reinforcement, and how modular architectures can support collaboration between human designers, generative models, and production-grade control software.

Keywords: drone light shows, large language models, semantic-syntactic integration, Skybrush Studio API, generative design pipeline, creative robotics, human-AI collaboration, formation sampling, trajectory optimization

Executive Summary

Designing drone light shows requires the reconciliation of two fundamentally different modes of reasoning: artistic ideation and analytical precision. While commercial tools such as SPH Engineering's Drone Show Software, Verge Aero's Design Studio, and Vimdrones Designer provide robust production environments for professionals, they remain heavily procedural. Designers must manually specify formations, transitions, and safety constraints within rigid graphical interfaces. These systems guarantee operational safety but offer limited support for the exploratory, iterative, and semantically rich stages of creative design.

At the same time, advances in large language models (LLMs) have made it possible to express complex spatial and temporal ideas in natural language. Recent academic work — including CLIPSwarm, SwarmGPT-Primitive, Swarm-GPT, FlockGPT, and LLM-Flock — has attempted to harness this capability to generate drone swarm behaviors directly from prompts. Yet these approaches largely treat LLMs as translators rather than collaborators. They produce symbolic or geometric intermediates that must still be processed by traditional solvers, without engaging with the workflows, constraints, or interpretive needs of professional show designers.

This project addresses that disconnect by proposing and implementing an integrated design pipeline that unites the semantic flexibility of LLMs with the syntactic rigor of analytical systems. Instead of attempting to have a language model produce executable trajectories directly, we introduce a structured framework in which generative models act as semantic front ends. They produce interpretable spatial representations—images, meshes, or symbolic formations—that are subsequently sampled, optimized, validated, and compiled through analytical methods. The final stage of this process integrates with the **Skybrush Studio API**, which performs automated safety checks, trajectory optimization, and binary compilation into deployable show files (`.csv` or `.skyc` formats).

Through this architecture, the project reframes LLM-assisted design not as an automation problem, but as a problem of *semantic-syntactic integration*. The pipeline demonstrates that human

creativity, machine interpretation, and analytical enforcement can coexist in a modular, interoperable workflow. The language model proposes; the analytical system verifies; the human designer iterates.

The research proceeded through four principal phases:

1. **Workflow Analysis:** Mapping the tools, cognitive processes, and constraints of existing professional pipelines through documentation review and, where possible, practitioner input.

2. **Pipeline Development:** Implementing a modular end-to-end system encompassing language interpretation, coordinate generation, temporal optimization, and validation through the Skybrush Studio API.

3. **Evaluation:** Assessing the semantic coherence, syntactic validity, and iterative usability of generated formations across multiple model architectures and prompting strategies.

4. **Design Synthesis:** Deriving practical guidelines for structuring prompts, incorporating analytic feedback, and supporting humanAI co-creation within production workflows.

The findings suggest that LLMs are most valuable not as autonomous generators of executable drone paths, but as accelerators of creative ideation. When coupled with analytical solvers and validation APIs, they enable rapid exploration of conceptual designs that can be safely transformed into physical performances. In this way, the project contributes both a working prototype and a conceptual model for integrating semantic and syntactic intelligence in creative robotics.

Ultimately, this work demonstrates that the path toward accessible and interpretable drone show design lies not in replacing human expertise, but in structuring collaboration—between human designers, generative models, and the analytical infrastructures that ensure safe and executable outcomes.

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

List of Appendices

Introduction

Problem Statement

Drone light shows represent a unique synthesis of creative design and algorithmic precision, transforming abstract artistic concepts into tightly coordinated aerial performances. While their visual and cultural impact has grown rapidly, the process of designing such shows remains complex, highly specialized, and constrained by strict physical and regulatory boundaries. Translating human imagination into executable drone trajectories requires both aesthetic sensitivity and formal guarantees of safety, spacing, and feasibility.

Recent proposals have explored using Large Language Models (LLMs) to bridge the gap between natural language expression and swarm behavior generation. In principle, this approach offers an intuitive design interface: a user describes the desired effect — "form a spiral that blossoms into a sphere" — and an LLM translates that prompt into flight paths or formation parameters. However, existing systems such as CLIPSwarm, SwarmGPT, and LLM-Flock have largely remained proof-of-concept demonstrations. These architectures use language models as semantic translators, producing intermediate representations such as geometric outlines or waypoints, which are then refined through separate analytic solvers. As a result, the LLM occupies a peripheral role: it initiates the design process but does not ensure executable, safe, or optimized outcomes.

The practical consequence of this separation is that current research prototypes have not achieved the level of reliability or workflow integration required for real-world deployment. The artistic and semantic potential of language-based co-design remains underutilized, while the technical constraints that define feasible drone motion are handled independently through manual or traditional optimization methods. This disconnect mirrors broader challenges in LLM-assisted content generation, where models excel at producing semantically rich ideas but struggle when constrained by formal syntactic or physical rules.

Our work begins from this recognition and seeks to explore whether these two modalities — the semantic flexibility of language models and the syntactic precision of analytical systems —

can be combined more effectively. Specifically, we propose an integrated pipeline in which the LLM functions not as a direct controller but as a generative front-end for a structured, verifiable production system. The final stages of this pipeline, including trajectory validation and compilation into deployable show files, are handled through the **Skybrush Studio API**, a professional toolchain for drone show design and execution. This framework grounds the generative capabilities of LLMs within the constraints of industry-standard safety and performance requirements.

Analysis Goals

The goal of this research is to evaluate how generative systems can contribute to the early design stages of drone light shows by integrating semantic input with deterministic analytical workflows. Rather than demonstrating fully autonomous choreography, the project examines how image-based generation and lightweight prompting can be coupled to existing validation tools to produce formations that are both expressive and executable.

Our analysis focuses on four questions:

1. What constraints shape the current workflow of professional drone show design, and where does early-stage creative exploration slow the process?

2. How can generative tools—either through images or lightweight language prompting—provide meaningful starting points for formation design?

3. How can sampling and analytical validation ensure that these generated formations are transformed into physically feasible, drone-ready outputs?

4. What role can production systems such as Skybrush Studio play in grounding generative content within real operational constraints?

This work treats generative models not as autonomous designers but as one component in a broader pipeline. The study examines how their outputs can be mediated, constrained, and ultimately rendered useful when paired with established analytical solvers.

Scope

This project is limited to the design phase of drone light show development. The focus is on transforming conceptual visual inputs—user-provided images or lightweight generative prompts—into validated drone formations. The work does not involve flight testing, real-time control, or new control algorithms. Instead, it concentrates on bridging creative generation with analytic feasibility.

The scope includes:

- Implementing a modular pipeline: image acquisition, 2D point sampling, analytic validation, and Skybrush-based compilation.
- Evaluating sampling strategies for converting images into drone-ready formations.
- Integrating the pipeline with the Skybrush Studio API for validation and deployment compatibility.
- Conducting conceptual exploration of how such a pipeline could be extended toward MCP-style interaction or 3D formation generation.

The project does not aim to create a standalone generative model for drone trajectories or a fully automated end-to-end design system. Instead, it formalizes and demonstrates a constrained but functional workflow that connects generative creativity with established, safety-pragmatic analytic tools.

Background

Commercial Tools

The design and execution of professional drone light shows are supported by a small ecosystem of proprietary software suites, including SPH Engineering's Drone Show Software,

Verge Aero's Design Studio, and Vimdrones Designer. These platforms provide integrated environments for defining 3D formations, scripting transitions, synchronizing lighting, and validating physical constraints. Their primary users are production companies and technical directors who work directly with proprietary drone fleets, often under hardware-specific licenses.

While these systems are robust and production-ready, they are not designed for exploratory or co-creative use. Their interfaces are graphical and timeline-based, emphasizing precision editing over conceptual ideation. Designers must specify individual formations, transitions, and timing parameters manually, with little algorithmic assistance during the early creative phase. Moreover, these environments assume substantial domain expertise: familiarity with aeronautical constraints, spatial reasoning in three dimensions, and the structure of executable show files. This combination of technical and artistic knowledge creates a high barrier to entry and limits the accessibility of drone choreography as a creative medium.

In recent years, **Skybrush Studio** has emerged as an influential hybrid system, bridging production-grade deployment with modular, API-accessible design tools. Its architecture separates front-end interfaces such as Blender extensions from a back-end engine that performs safety validation, trajectory optimization, and compilation into binary formats (e.g., `.skyc`). The **Skybrush Studio API** allows external systems to interact with this back-end directly through HTTP requests, enabling procedural or automated generation of drone flight plans. This capacity positions Skybrush not only as a professional editing suite but also as a potential integration layer for research pipelines that combine semantic generation with analytical verification.

From the standpoint of creative ideation, however, all these tools — including Skybrush in its commercial form — remain fundamentally syntactic. They excel at enforcing constraints and ensuring safety but offer no mechanisms for generating or interpreting creative intent expressed in natural language or visual form. The "semantic front end" of drone choreography remains largely undeveloped.

Academic Work

Academic research into LLM-assisted swarm design has sought to fill this semantic gap by introducing generative, language-driven approaches to formation and trajectory planning. Across systems such as *CLIPSwarm*, *SwarmGPT-Primitive*, *Swarm-GPT*, *LLM-Flock*, and *FlockGPT*, the central objective is consistent: to make swarm design more intuitive by allowing human operators to describe desired behaviors in natural language.

While conceptually aligned, these systems differ in scope and architecture. **CLIPSwarm** uses CLIP embeddings to associate textual prompts with 2D geometric outlines, producing silhouettes through iterative refinement of contour shapes. Its outputs are visually legible but limited to planar designs without dynamic motion or 3D extension. **SwarmGPT-Primitive** adopts a library-based approach, mapping language to predefined motion primitives that are synchronized with musical beats; it provides consistency but constrains expressiveness. **Swarm-GPT** attempts a more flexible waypoint generation process, where LLMs output raw coordinates subsequently filtered for safety, but the method suffers from verbosity, limited editability, and fragile temporal coherence.

LLM-Flock departs from centralized control by embedding LLMs directly on each agent, coordinating flight patterns through decentralized consensus. Though theoretically elegant, the approach introduces unpredictability and remains confined to simple geometric formations. Finally, **FlockGPT** introduces a dialogue-based interaction model, using signed distance functions to represent target shapes and allowing the user to iteratively refine geometry through conversation. Yet even this interaction model operates on static spatial data and does not address the temporal or safety constraints inherent in executable drone shows.

Across this literature, the primary role of the language model remains interpretive or symbolic: it translates natural language into an intermediate representation but defers all physical validation, optimization, and compilation to analytic solvers. The separation between semantic expression and syntactic enforcement persists, mirroring the same division seen in professional production tools.

Identified Gaps

Taken together, commercial and academic approaches outline two complementary but disconnected strengths. Commercial software provides the formal infrastructure — robust safety guarantees, validated trajectories, and executable output formats — but offers little support for creative ideation. Academic systems, conversely, provide generative flexibility through natural language or image-based interfaces, yet lack integration with the analytical frameworks that ensure real-world feasibility.

This research seeks to reconcile these domains through a structured, end-to-end pipeline that unites semantic generation and syntactic validation. The core insight is that LLMs should not attempt to directly produce executable flight paths, but rather to generate interpretable intermediate representations — such as 2D or 3D spatial configurations — which are then transformed, optimized, and verified through established analytical systems. The **Skybrush Studio API** provides a natural anchor for this integration, offering a programmatic interface where creative outputs from generative models can be converted into validated trajectories and compiled into deployable show formats.

In summary, the key methodological gaps motivating this work are:

- The absence of an integrated pipeline linking semantic generation to formal verification.
- The lack of human-centered design frameworks that combine creative exploration with operational feasibility.
- The need for modular interfaces — such as those provided by Skybrush — that can mediate between generative AI systems and existing production infrastructures.

This project positions itself at that intersection: developing a structured, hybrid workflow that unifies language-driven ideation with professional-grade safety and execution frameworks.

Methodology

This work implements a structured, prototype-level pipeline for the early-stage design of drone light show formations. The methodology emphasizes the integration of generative visual inputs with deterministic analytical validation, reflecting the separation between semantic intent and syntactic execution that characterizes professional workflows. The pipeline proceeds from image acquisition to point sampling, Skybrush-based validation, and final compilation into deployable formats.

Workflow Analysis

The development of the pipeline begins with an examination of existing drone show design workflows. Although the project does not replicate full production practices, it draws on publicly available documentation and demonstrations from commercial platforms such as SPH Engineering's Drone Show Software, Verge Aero's Design Studio, and Skybrush Studio. These materials clarify the structure of professional pipelines, the constraints imposed during formation design, and the requirements for producing validated flight plans. This analysis ensures that the implemented system remains aligned with real operational expectations, even in a constrained prototype form.

Pipeline Architecture

The implemented pipeline consists of four modular stages that transform a visual input into a validated drone formation compatible with Skybrush Studio. An overview of this workflow is shown in Figure 1.

1. **Image Acquisition:** The pipeline begins with a visual reference supplied either directly by the user or generated via lightweight prompting through an external model. This image represents the semantic intent of the designer and provides the basis for subsequent formation extraction.

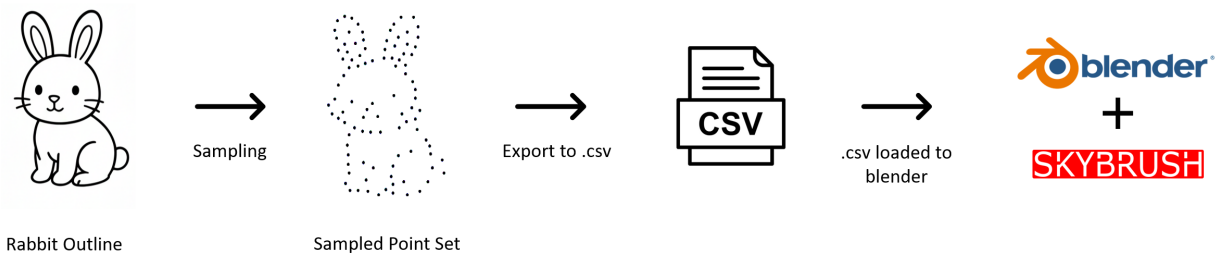


Figure 1: *Overview of the implemented pipeline, from image-based input to sampled formation extraction and Skybrush-based analytical validation.*

2. **Point Sampling:** The reference image is converted into a binary silhouette, from which a fixed number of drone positions are sampled. Several sampling strategies were evaluated, including grid sampling, blue-noise sampling, and farthest-point sampling. The latter proved most effective at preserving recognizable structure while maintaining minimum spacing between points. Figure 2 illustrates the transformation from silhouette to sampled points.

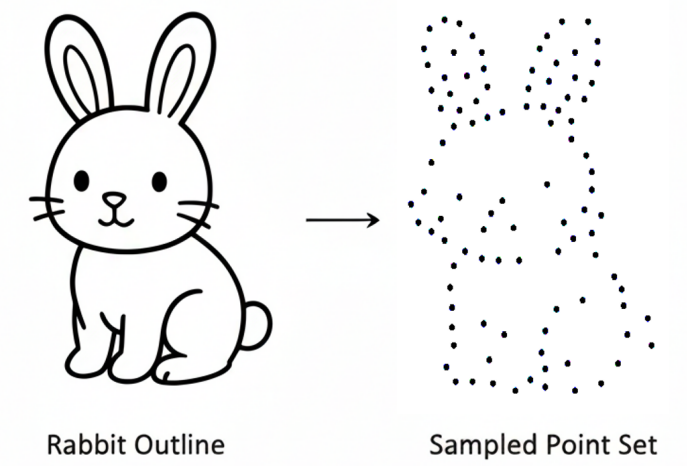
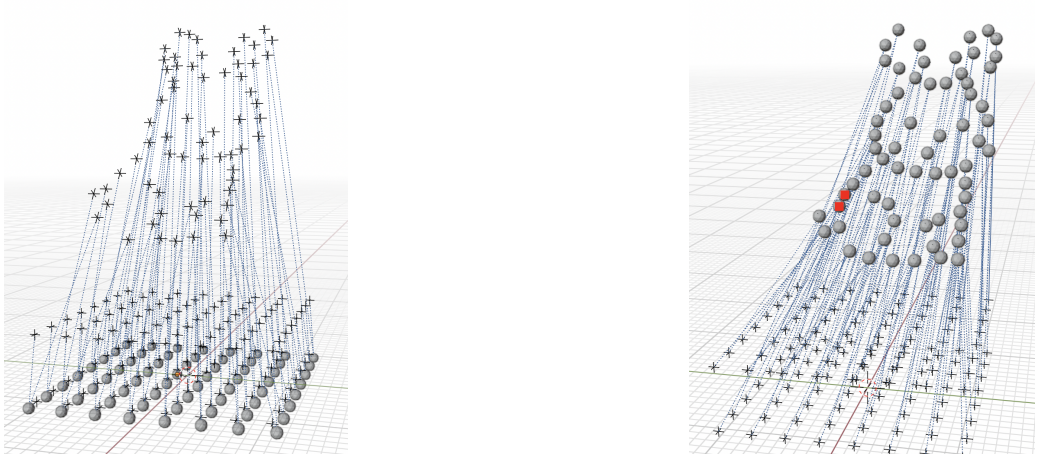


Figure 2: *Example of a binary silhouette (left) and the resulting sampled point set (right) produced through farthest-point sampling.*

3. **Skybrush Integration:** Sampled coordinates are exported as `.csv` files and imported into Skybrush Studio via its Blender extension. Within this environment, the sampled formation can be visualized and inspected using the same interface employed in professional drone show design.



(a) *Imported formation visualized in Blender.* **(b)** *Generated trajectories and temporal linking.*

Figure 3: *Visualization of sampled formations and trajectory planning using the Skybrush Studio extension within Blender.*

4. **Validation and Export:** Skybrush Studio performs analytical checks on spacing and formation feasibility. Once validated, the formation can be compiled into deployable formats such as `.skyc`, enabling downstream integration with compatible drone control software.

This architecture demonstrates how generative or user-driven visual concepts can be translated into syntactically valid formation data through established analytical tools.

Integration with Skybrush Studio API

Although the pipeline primarily uses the Blender-integrated interface of Skybrush Studio, its design aligns with the capabilities of the Skybrush Studio API, which exposes programmatic endpoints for safety verification and binary compilation. The system's reliance on industry-grade validation tools illustrates how generative front ends can be coupled with existing production infrastructures without modifying their internal logic. This highlights the feasibility of modular semantic-syntactic integration in a realistic workflow.

Evaluation

The evaluation focuses on qualitative and structural criteria rather than numerical performance metrics. Specifically, the pipeline is assessed on:

- **Sampling Fidelity:** The extent to which each sampling method preserves the recognizable structure of the input silhouette.
- **Feasibility in Skybrush:** Whether the exported formations import cleanly and satisfy spacing constraints as enforced by Skybrush Studio.
- **Workflow Coherence:** The clarity of the transformation from image input to validated formation, emphasizing interpretability and modularity.

Evaluation is supported by visual inspection of sampled point sets and their rendered counterparts in Skybrush Studio.

Limitations

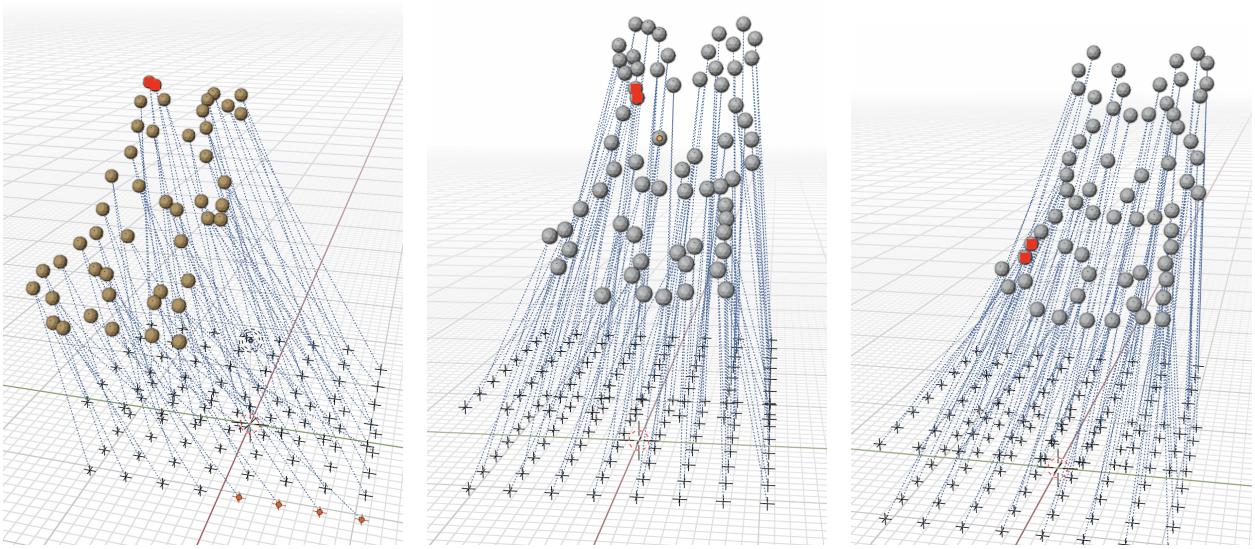
The methodology is constrained by its focus on static 2D formations. It does not generate or optimize temporal transitions, and it does not incorporate full 3D formation sampling or trajectory planning. The use of external image-generation tools captures only a narrow slice of semantic input, and the analytical validation depends on the capabilities provided by Skybrush Studio. While these constraints reflect the prototype nature of the system, they do not undermine its central aim: to demonstrate a coherent workflow linking conceptual imagery to analytically validated drone formations.

Despite these limitations, the methodology establishes a reproducible framework that illustrates how generative inputs and analytic solvers can be coupled in practice. This provides a foundation for future work in extending the pipeline to dynamic, 3D, or fully integrated generative-analytic systems.

Findings

This project does not aim to demonstrate autonomous LLM-driven choreography, but rather to assess the feasibility of a constrained pipeline that links generative visual input to analytically validated drone formations. The findings presented here reflect the empirical performance of the implemented system and the practical constraints observed during its construction.

1. Sampling Quality and Formation Fidelity



Grid Sampling *Blue-Noise Sampling* *Farthest-Point Sampling*

Figure 4: ***Comparison of sampling strategies used to convert binary silhouettes into drone formations.*** *Grid sampling runs efficiently but produces uneven density; blue-noise sampling yields scattered and harder-to-recognize forms; farthest-point sampling preserves global structure while maintaining minimum inter-drone spacing.*

Evaluation of the image-to-formation stage showed that the choice of sampling strategy has a decisive impact on the recognizability and structural coherence of the resulting drone layouts. As illustrated in Figure 2, grid sampling runs efficiently but produces uneven point density that distorts fine details. Blue-noise sampling generates more uniformly spaced points, yet the result appears visually scattered and often loses the silhouette's global structure. Farthest-point sampling consistently produced the most stable and interpretable formations by preserving overall

shape while maintaining minimum spacing between drones, making it the most suitable method for converting binary silhouettes into deployable formations.valuation of the image-to-formation stage demonstrated that sampling strategy plays a decisive role in the recognizability and structural coherence of the resulting drone layouts. Grid sampling proved efficient but produced uneven density. Blue-noise sampling yielded visually scattered formations. Farthest-point sampling consistently preserved global structure while maintaining minimum separation constraints, making it the most suitable method for converting silhouettes into stable drone formations.

2. Integration Feasibility with Skybrush Studio

(a) *Takeoff grid* **(b)** *Transition phase* **(c)** *Final sampled formation*

Figure 5: *Rendered frames from the compiled* `.skyc` *file inside Skybrush Studio, showing the takeoff sequence, transition motion, and final assembled formation.*

The pipeline successfully exported the sampled coordinates as a `.csv` file and compiled them into a `.skyc` show file using Skybrush Studio. Rendering the resulting show in the Skybrush simulator confirmed that the generated formation data were structurally consistent and interpreted correctly by the production-grade toolchain. The animation demonstrated that the drones lifted from a standard takeoff grid, followed interpolated transition paths, and assembled into the sampled formation without inconsistencies in spacing or trajectory alignment.

3. Constraints of 2D Static Formations

Because the implemented system focuses on static 2D silhouettes, it does not address volumetric formations, temporal transitions, or dynamic choreography. This constraint clarified the

limits of using images as semantic inputs: while effective for capturing shapes, they do not encode motion, timing, or 3D structure. These limitations reinforce the need for additional stages if such a pipeline were extended toward full performance design.

4. Practical Insights for Pipeline Design

The development process yielded several practical insights regarding modular system design. First, separating semantic input from analytical validation reduces failure modes when integrating heterogeneous tools. Second, using established solvers such as Skybrush Studio for constraint enforcement ensures that prototype systems remain compatible with real production infrastructures. Third, even limited generative input—such as silhouettes—can meaningfully accelerate early-stage formation design when paired with deterministic post-processing.

5. Summary

Overall, the findings indicate that a constrained, image-driven pipeline can produce analytically valid drone formations and interface cleanly with existing professional tooling. While limited in scope, the prototype demonstrates a practical method for linking generative visual concepts to executable formation data, and provides a foundation for future extensions into 3D sampling, temporal planning, or deeper integration with language-driven interfaces.

Discussion

The development of this prototype clarifies both the possibilities and the limits of linking generative inputs with analytical validation in the context of drone light show design. Although the system is intentionally constrained to 2D silhouettes and static formations, its construction reveals several broader themes relevant to hybrid creative-analytic workflows.

Separation of Creative Input and Analytical Enforcement

A central observation is that separating semantic input from syntactic enforcement simplifies integration with production-grade tools. By treating the input image as a high-level expression of intent and delegating feasibility checks entirely to Skybrush Studio, the pipeline avoids the brittleness typically associated with end-to-end generative systems. This division of labor proved practical: transformations remained interpretable, and failures were easier to diagnose because each stage preserved a clear functional boundary.

Role of Sampling in Bridging Representation Layers

The sampling stage emerged as the central intermediary between conceptual imagery and executable formation data. Converting a silhouette into a discrete set of drone positions requires balancing visual fidelity with spatial constraints, and the behavior of different sampling strategies directly influenced that balance. Grid sampling maintained structural alignment but produced uneven point density; blue-noise sampling created visually dispersed patterns with reduced recognizability; and farthest-point sampling provided the most consistent preservation of overall form while maintaining minimum separation between points. These observations indicate that sampling strategy, rather than the generative origin of the image, plays a determining role in the quality and feasibility of early-stage formation design.

Practical Interoperability with Production Tools

Importing sampled coordinates into Skybrush Studio validated the practical feasibility of the pipeline. The tool's existing infrastructure for visualization, scaling, and collision checking absorbed the upstream variability of the sampling process without modification. This reinforces the value of designing systems that complement, rather than replace, established workflows. The pipeline demonstrates that lightweight generative front ends can integrate meaningfully with industrial solvers even when the generative component is minimal.

Limitations of the Current Approach

The prototype exposes clear limitations. The exclusive focus on 2D silhouettes constrains the expressive range of formations and precludes volumetric or dynamic choreography. The absence of temporal modeling means that the system cannot produce transitions or full sequences. Furthermore, the reliance on external image generation limits semantic nuance: an image captures shape but not narrative or motion. These constraints reflect scope rather than failure, but they delimit the system's usefulness for full production pipelines.

Implications for Future Work

Despite its limitations, the prototype illustrates how a generative-analytic pipeline can be constructed incrementally. Future extensions may incorporate 3D mesh sampling, temporal optimization, or deeper integration with language-driven interfaces. More broadly, the project suggests that hybrid systems in creative robotics benefit from modular architectures in which generative tools provide flexible inputs while analytical solvers ensure feasibility. Such an approach may offer a viable template for other domains where expressive design must coexist with formal operational constraints.

Conclusion

This project demonstrates the feasibility of a constrained, image-based pipeline for generating analytically validated drone formations. Although deliberately limited in scope, the prototype shows that simple visual inputs—whether user-provided or generated externally—can be transformed into executable formation data when paired with appropriate sampling algorithms and established validation tools. In doing so, the system provides a concrete example of how creative intent can be translated into operationally feasible structures without requiring end-to-end generative control.

The implementation highlights the central role of the sampling stage as the intermediary

between conceptual imagery and formal analytic constraints. Farthest-point sampling proved most effective at preserving recognizable structure while satisfying minimum-spacing requirements, and the resulting formations imported cleanly into Skybrush Studio for inspection and compilation. This interoperability indicates that lightweight generative inputs can be coupled with production-grade analytical infrastructures without altering their internal logic.

While the current system is restricted to static 2D silhouettes, it also reveals a more general architectural pattern. By separating semantic input from syntactic verification, the pipeline illustrates how hybrid creative-analytic workflows can be structured in modular layers. This pattern is not confined to the present prototype: it suggests a broader framework in which generative components—whether image models, language interfaces, or interactive sketching tools—provide flexible starting points, and deterministic solvers enforce feasibility. Such an approach offers a systematic way to balance expressive freedom with operational rigor in creative robotics.

Several avenues for future development follow naturally from this work. The pipeline could be extended to full-color image processing, 3D mesh sampling, or basic temporal linking to support multi-frame choreography. A unified interface, possibly as a Blender or Skybrush extension, could integrate the stages into a coherent design environment. More broadly, richer forms of generative input could be incorporated as long as the downstream analytical constraints remain explicit and enforceable.

In sum, the contribution of this project lies not only in the implemented prototype but in the methodological structure it exemplifies. It provides a foundation for future systems that couple generative creativity with analytic validation and outlines a path toward more interpretable, modular, and operationally grounded design workflows for drone swarm performances and related creative domains.

Additional Data Analysis

[Additional analysis content]

Code Documentation

[Code documentation]

LLM Prompt Examples and Outputs

[LLM prompts and outputs]