

Static Illumination in AirSim using MinDist and Quota Balanced Illumination

Henil Shelat | Collaborator: Baladitya Swaika

Objective:

Perform the simulation of creating a static illumination using Flying Light Specks, implementing the MinDist and Quota balanced Algorithm. Further, to simulate the architecture of the FLS Display using AirSim.

Unmanned Aerial Vehicles are used for diverse applications. Flying Light Specks or FLSs is a miniature sized drone which has light sources to generate different colors and textures. Cooperating them in a swarm can lead to a rendering which would create a static illumination, However, creating a virtual 3D object in a physical space using FLS drones is a challenging task, because of the cost and a lot of open research in the field. Hence, we simulate the construction of a static illumination using AirSim, a cross-platform drone and ground vehicle simulator which is built on top of Epic Games' Unreal engine.

Project Description:

Using FLS drones, we simulate the methods offered for displaying 3D illuminations. Here, coordinated swarms of cooperative FLSs create a display by illuminating virtual objects in a prespecified 3D space[1].

This project is the implementation of the paper 'Display of 3D illuminations using Flying Light Specks' by Prof. Shahram Ghandehaziradeh, here I am providing the static illumination of a rose using FLS drones. Here, we use the research mentioned as the foundation.

Despite it being an implementation of the paper, it is challenging to perform it, since it also requires us to generate the architecture of an FLS Display which requires a conveyor belt to simulate the collection process.

Milestone1: Environment Setup and Feasibility check - END 10/12/2022

During the first week of the project duration, we will be focusing on assessing the feasibility of an FLS rendering a light source in the AirSim. We will be setting up the unreal engine with AirSim to generate the simulations. We will divide the research amongst ourselves to check the feasibility. This task would require attaching lights to the drone's surfaces and performing dry runs to see if the simulation is feasible for the project. We will explore other options like Gazebo, hector, etc. if AirSim does not meet the feasibility check. Further, parsing through the ROS Bags provided by the professor for rendering the point clouds. Here we divide the work amongst partners and perform independent research and share the results to check the feasibility of the system.

Milestone 2: Simulating the flight paths using MinDist Algorithm - END 10/23/2022

An algorithm called MinDist[1] cycles through each point, calculates its distance from a dispatcher and then assigns it to the dispatcher with the smallest euclidean distance.

Based on the algorithm mentioned in the paper, the task would be to implement that and provide a simulation of the algorithm. However, there is a huge disadvantage to the MinDist algorithm, it does not take into account the distance traveled by the FLSs.

Milestone 3: Simulating the flight paths through the dispatch stations using the QuotaBalanced Algorithm and use of the simulator to generate different point clouds - END 11/04/2022

QuotaBalanced Algorithm reduces each dispatch station to a function that considers the travel time of the FLSs[1]. Here, the goal is to implement the QuotaBalanced Algorithm based on the paper. It considers several variables like the distance, speed of the FLSs, and the rate of the FLS flying out of the dispatcher.

We render the simulations to render 2 different point clouds of the Princeton Benchmark Car and the first frame of the Rose Clip point cloud sequence which implement the aforementioned algorithm to perform the tasks.

Milestone 4: Design the FLS Display Architecture and implement the conveyor belt for garbage collection - END 10/28/2022

Designing the FLS Display Architecture with the essential components to represent the concept indicated in the study and incorporating that into the simulation to mimic the concept mentioned in the research paper. Here, there are several components to implement. First, the inclusion of the components such as Hangar, Dispatcher, and Terminus. Further, this would be a challenging task since we would have to demonstrate the change in the number of drones, the replacement mechanism, and the implementation of the garbage collector using a conveyor belt.

Milestone 5: Charging the FLSs battery and implementation of STAG - END 11/14/2022

Actual drones do consume power and are needed to be recharged. In the design mentioned in the paper, we have a dedicated charging station associated with the drones where the drones are charged and after which they move to the hangar to be available for operation again. In the simulation using AirSim drones don't discharge, so we create a simulation where the drones fly to the charging station and we offer a time delay after which the drones are available again for the process.

STAG is a novel algorithm that takes into account the charging of the drone and creates an objective function to reduce the number of charging stations and the number of FLSs to be used

for the rendering. As mentioned before, we simulate this process to showcase a real-world scenario since drones in AirSim do not get discharged.

Milestone 6: Collision Detection - END 11/23/2022

Collision detection is extremely crucial in a real-world scenario. Not only can it affect the rendering, but it is also dangerous causing unexpected damage. The cost of the collision scales up when the swarms have tens of thousands of drones. Hence, we use the garbage collection architecture mentioned in the paper, to handle collision and failures of drones, which moves the drones to the terminus present on either side of the display architecture.

A QuotaBalanced algorithm may result in collisions. Airsim houses a variety of sensors, one of which is a collision sensor, which can report when the object collides with another object, as Quotabanced can cause collisions. And we will try to analyze data collected by AirSim to see how collisions play a role in our architecture.

Milestone 7: Failure Handling and Analysis - END 11/27/2022

Mechanical objects tend to fail. The implementation of the handling procedures is to make sure this task is handled carefully, not to create further damage or hinder the path of other drones. 1:1 backup scheme and garbage collection with the use of a conveyor belt that keeps removing drones. It keeps sending those failed drones to a Terminus in the station which houses all the failed drones in that compartment. Since this is a simulation, we will make drones fail as they will actually not fail and thus we will not be implementing detecting failure.

Analysis before the final submission includes making use of the data generated from the failures and collisions to understand the algorithm in a better way to simulate the trade-offs between various schemes. The analysis based on failures helps us understand the algorithms in a better way. Also, use this time to create an application demo that would be required.

Conclusion

We want to be able to use and simulate the professor's paper's foundations by the project's conclusion. This is a fascinating paper, and putting it into practice would show us what the future has in store for us. This will present us with intriguing opportunities.

GitHub Link: https://github.com/henilshelat7/Static_illumination_using_FLS

References:

[1] Ghandeharizadeh, Shahram. (2022). Display of 3D Illuminations using Flying Light Specks. 10.1145/3503161.3548250.

<https://doi.org/10.1145/3503161.3548250>

[2] Shital Shah, Debadeepta Dey, Chris Lovett, Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles.

<https://arxiv.org/abs/1705.05065>

[3] AirSim: <https://microsoft.github.io/AirSim/>