Team Apache Stealth: Flying through the Unknown!

Ankit Mittal
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: amittal@wpi.edu

Rutwik Kulkarni
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: rkulkarni1@wpi.edu

Abstract*—*(Utilizing 2 late days) **This project addresses the challenge of enabling a small drone to autonomously navigate through an irregularly shaped window in a diverse environment. Given the computational and weight limitations inherent to small drone platforms, the project focuses on developing innovative, resource-efficient algorithms suitable for the NVIDIA Jetson Orin Nano. The key components include a versatile perception stack for detecting and tracking window gaps, and a dynamic planning and control stack capable of generating safe navigation paths. The project emphasizes the use of simulated environments for preliminary testing, employing a variety of window shapes and textures to ensure robustness. The final evaluation involves a live demonstration where the drones navigate through unpredictably placed windows, with an emphasis on accuracy and speed of navigation. The project's success will be measured by the drone's ability to safely and swiftly identify and traverse the largest available gap without reliance on cloud computing resources, showcasing the potential of compact, intelligent autonomous systems in complex real-world scenarios.**

I. Introduction

A. Background and Context

The previous course module focused on navigating through environments with known structures, like square windows, using deep learning techniques (e.g., U-Net) and Perspective-n-Point (PnP) algorithms. This approach, while effective for predefined shapes, faces challenges in more unpredictable and unstructured environments.

B. Problem Statement

Real-world scenarios often require drones to navigate through irregular and unknown-shaped windows, crucial for applications in search and rescue, exploration, or reconnaissance. The complexity of these environments demands a more adaptable and versatile approach than what conventional methods offer.

C. Objective

This paper builds upon the coursework and research in implementing visual perception methods using our drone's monocular camera. The primary objective is to extend the capabilities of our drones to navigate through irregularly shaped gaps using only visual cues. This involves developing and evaluating methodologies that enable drones to accurately perceive and interpret complex environments with a single camera.

D. Contribution

The key contribution of this report lies in its application to a specific use case of navigating through irregular gaps. We introduce an approach that combines inferring optical flow using a deep learning-based method with post-processing techniques to segment out the largest gap. The rest of the cavities and open spaces are effectively ignored, focusing solely on the viable navigation path. This is complemented by visual servoing techniques to guide the drone through the identified window, showcasing an integrated solution for complex navigation tasks.

E. Structure of the Report

The report is organized as follows:

- **Hardware Specifications**: Outlines the drone's hardware, focusing on its onboard computational unit, the NVIDIA Jetson Orin Nano, sensors, and camera systems.
- **Environment Setup**: Describes the testing environment, detailing the irregular windows' shapes, sizes, and textures.
- **Perception Stack**: Discusses the visual perception methods for gap detection and segmentation, emphasizing their integration with the drone's monocular camera system for effective navigation.
- **Planning and Control**: Covers the strategies for path planning and control, focusing on algorithms for precise maneuvering and obstacle avoidance.
- **Results**: Presents experimental findings, including the drone's gap detection accuracy and navigation efficiency, with a comparative analysis of existing methods.
- **Conclusion and Future Work**: Summarizes key findings, contributions, and limitations, while suggesting avenues for future research in drone navigation and perception enhancements.

Link To Videos: **Click Here**

II. Hardware Specifications

The DJI Tello Edu drone, equipped with a 5-megapixel camera (2592x1936 photo resolution, 960x720 video streaming), weighs 82.6 grams and has a 13-minute flight capability with a 1.1 Ah/3.8 V battery. It operates within a 100-meter range and up to 10-meter altitude, using the DJITelloPy Edu Python library for programming. Complementing it is the NVIDIA Jetson Orin Nano, a compact AI and edge computing module, essential for processing the drone's real-time image processing and autonomous navigation tasks, ensuring efficient and precise visual data handling in complex environments.

III. Environment Setup

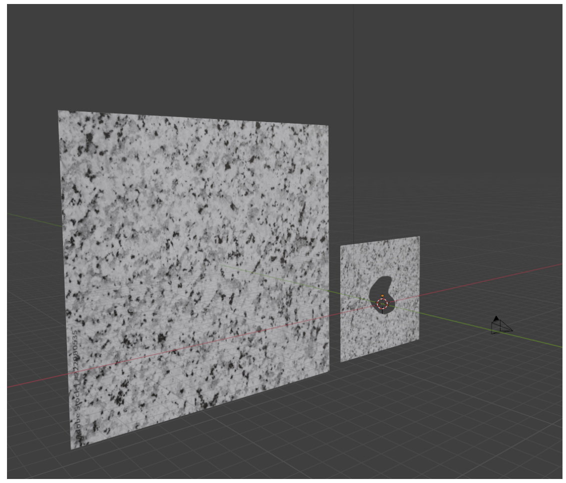

Fig. 1: Environment Image in Blender

Fig. 2: Environment Image in Real Environment

IV. Perception Stack

A. *Sensors and Data Acquisition*

The data acquisition for this project relies on the monocular camera equipped on the DJI Tello Edu drone, as detailed in the Hardware Specifications section. The primary challenge in data acquisition was positioning the drone so that the camera's field of view encompasses the potential gaps in the wall. This positioning is critical for ensuring that the subsequent image processing and gap detection algorithms have the necessary visual data to identify viable paths for navigation.

Fig. 3: Positioning the Drone such that the Gap is in FOV of Drone's Camera

B. *Image Processing*

The cornerstone of our image processing approach is the inference of optical flow using the SPyNet network. Optical flow, a concept in computer vision and image processing, describes the pattern of apparent motion of objects, surfaces, and edges in a visual scene, caused by the relative movement between an observer and the scene. SPyNet's processing time ranges from 0.2 to 0.5 seconds per inference, striking a balance between speed and accuracy. In the realm of computational considerations, SPyNet was selected over other candidates like FlowNet 2.0 and PWC-Net for its optimal balance between rapid inference time and adequate accuracy. Classical methods of optical flow determination, though expedient, lacked the robustness and precision provided by contemporary deep learning-based approaches. We also investigated other deep-learning models for optical flow, such as FlowNet2.0 and PWC-Net. However, these models are computationally demanding, with over 100 million parameters, making them resource-intensive. Hence, We chose SpyNet for our needs, and to boost its precision, we conducted several optical flow evaluations. Through this process, we pinpointed and chose the region most likely to contain a gap by analyzing the gap probability map associated with each optical flow map. This compensates for the potential shakiness or distortion in the drone's captured images. This methodology ensures that the final optical flow output is both reliable and representative of the actual scene dynamics, facilitating accurate gap identification. The conceptual foundation for this approach is derived from Dr. Nitin Sanket's GapFlyt paper, which addresses a similar challenge in drone navigation.

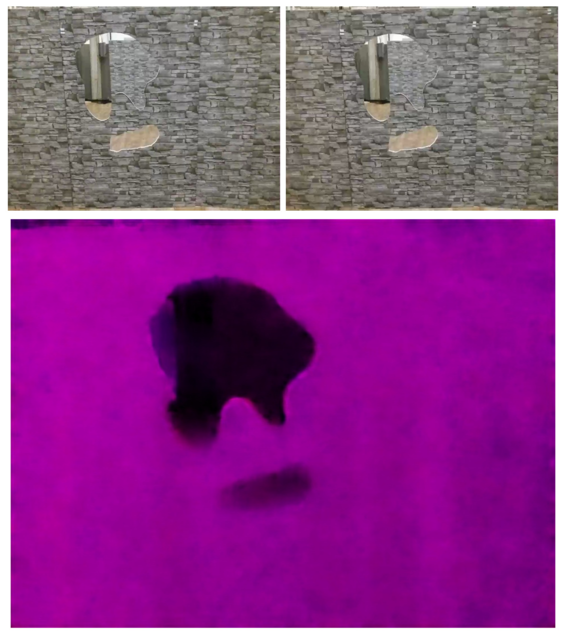

Fig. 4: [Top left] - **Image1**, [Top Right] - **Image2**, [Bottom] - **Optical Flow of Image1 and Image2**

To generate the probability map for each optical flow, we applied the Otsu thresholding method. This technique effectively highlights the regions most likely to represent gaps. Following this, we overlaid the images, employing binary operations on each pixel to accurately identify the gaps. This approach proved to be highly effective, consistently yielding reliable results even when dealing with suboptimal/bad optical flow maps.

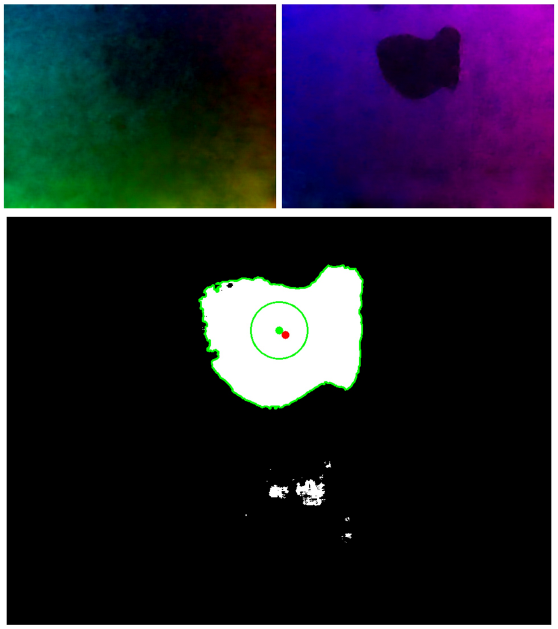

Fig. 5: [Top left] - **Optical Flow 1**, [Top Right] - **Optical Flow 2**, [Bottom] - **Binary Mask Image obtained after Thresholding with Position Markers**

Figure 5 illustrates that the left optical flow map lacked accuracy, primarily due to image shakiness and distortion from the drone's camera. Similarly, the right side image also hinted at a potential gap near the bottom. However, when these maps were superimposed, the resulting composite enabled us to obtain a precise binary mask. This mask effectively highlighted the areas with the highest probability of being gaps, as demonstrated in the figure. In the binary image, the green dot represents the center of the image, which has been adjusted upwards by 150 pixels for 960 X 720 images (determined based on an experimental basis) to account for the camera's tilt. Meanwhile, the red dot indicates the center of the contour.

C. *Softwares and Frameworks*

The project employed Python and PyTorch for programming and deep learning tasks, respectively, with OpenCV for image processing. DJITelloPy was used for drone control, ensuring a cohesive and efficient system for real-time autonomous drone navigation.

V. Path Planning and Control

A. *Initial Positioning and Movement*

Considering the wall's specified distance of 1.8 to 3 meters from the origin, with a possible tilt angle of -20 to +20 degrees, the drone's initial strategy involves a predetermined forward movement. This initial maneuver is designed to compensate for potential takeoff errors or hardware inconsistencies, ensuring that the drone starts from a consistent and reliable position relative to the wall.

B. *Gap Detection and Alignment*

Utilizing the Perception Stack, the drone computes the optical flow from its camera feed and post-processes this data to identify the center of the largest gap in the wall. As detailed in the Perception Stack section, the processed output is a binary image with a red dot marking the estimated gap center and a green dot at the image center, surrounded by a threshold-indicating green circle. This setup is integral to the subsequent visual servoing algorithm.

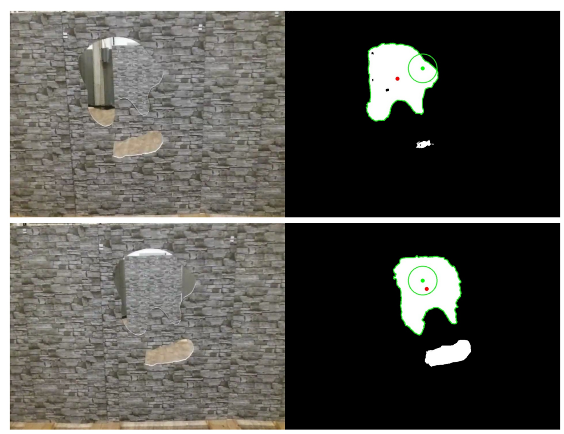

Fig. 6: Visual Servoing in Action. [Top] - **Drone not aligned with Gap Center.** [Bottom] - **Drone Aligned with Gap Center**

C. *Visual Servoing Implementation*

Visual servoing is a critical component of the drone's navigation strategy. It operates by ensuring

that the red dot (gap center in the image) is aligned within the green circle (the threshold around the image center). When the alignment is achieved, indicating that the drone is positioned correctly relative to the gap, a forward command is issued to the drone.
The visual servoing mechanism is executed using the position control feature in the DJITelloPy library. The process involves calculating the distance between the red and green dots on the x and y axes of the image plane. As the perception stack does not provide depth information, an experimental approach was taken to estimate a workable depth between the drone and the wall, considering the known distance range. This estimation was used to compute a scaling factor, transforming the pixel distances into metric units. The drone then moves according to these calculated distances, repeating the process until the desired alignment is achieved.

D. Depth Estimation and Scaling Factor

The depth estimation process is crucial for the success of the visual servoing strategy. Without direct depth information from the perception stack, the team conducted multiple trials to identify a practical distance between the drone and the wall. This experimental distance was then used to derive a scaling factor, crucial for converting the distances on the image plane to real-world metrics. This scaling factor plays a pivotal role in ensuring that the drone's movements are precise and aligned with the real-world dimensions of the environment.

In summary, the planning and control strategy of the drone involves an initial forward movement for position normalization, followed by gap detection and alignment using a sophisticated visual servoing approach. This approach leverages experimental depth estimation and scaling to translate image plane measurements into real-world navigational commands, ensuring precise and effective movement toward the target gap.

VI. Results

Below is the link to a live demonstration showcasing the capabilities of our perception, planning, and control stack in action. **LIVE DEMO!!**

A. *Optical Flow Results for Blender Test Set*

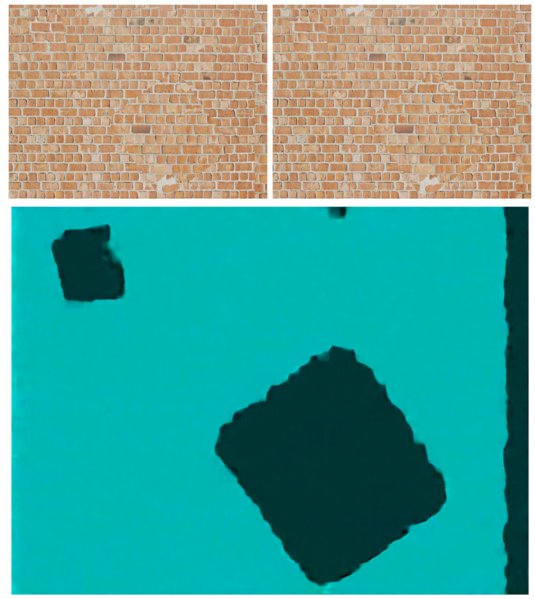

Fig. 7: Blender Environment 1: Brick Texture

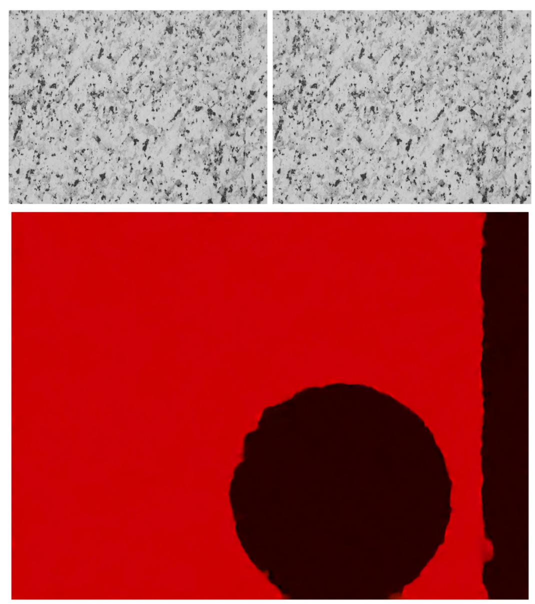

Fig. 8: Blender Environment 2: Granite Texture

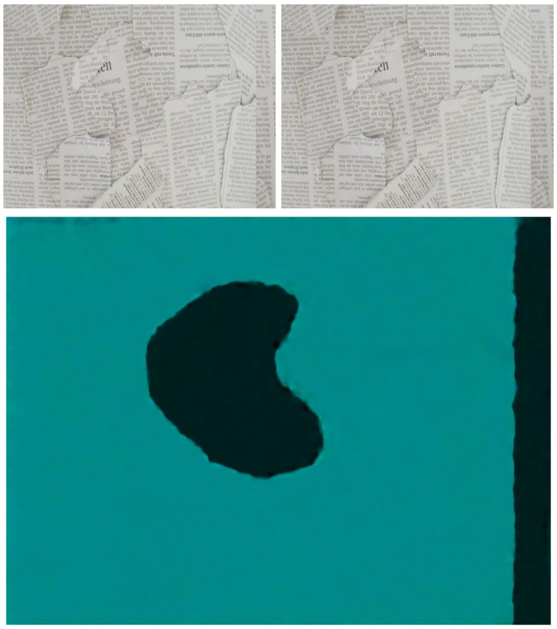

Fig. 9: Blender Environment 3: Paper Texture

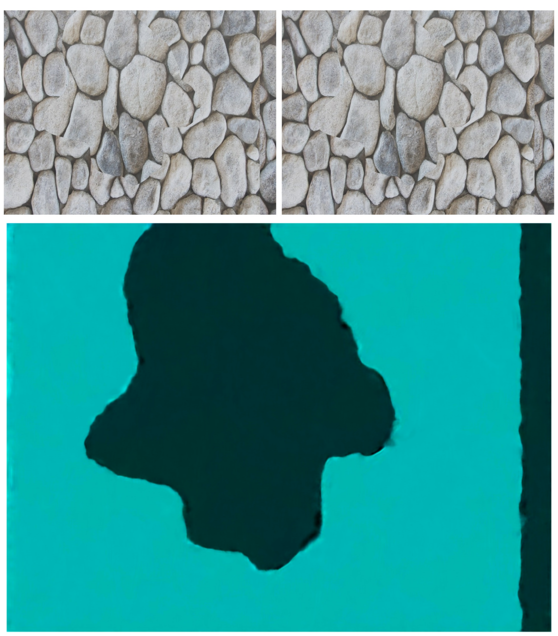

Fig. 10: Blender Environment 4: Stone Texture

B. *Important Frames from the Live Feed of Demo*

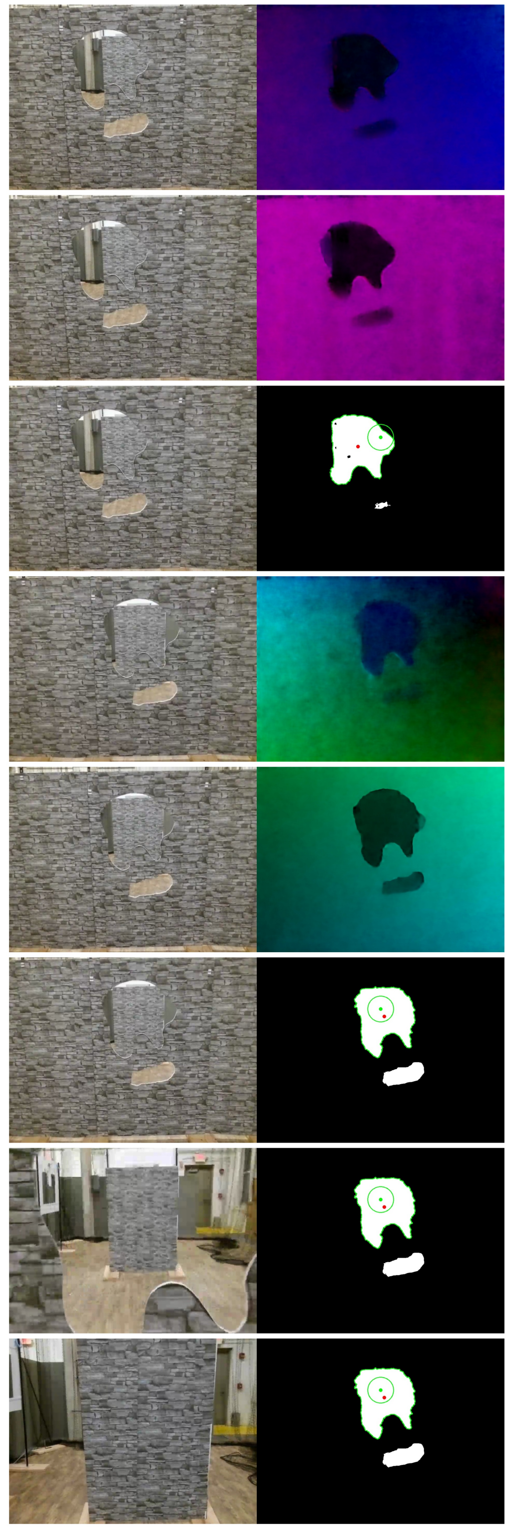

Fig. 11: Snippets from the Test Run

VII. Conclusions and Future Work

A. Conclusions

This project successfully demonstrated the capability of a small drone, specifically the DJI Tello Edu, to autonomously navigate through irregularly shaped windows in a controlled environment. Employing the NVIDIA Jetson Orin Nano, we developed a robust perception stack that utilizes optical flow for gap detection and a visual servoing approach for precise navigation. Our approach effectively overcomes the challenges posed by limited computational resources and the constraints of using a monocular camera system. The live demonstrations and tests confirmed the viability of our system in accurately identifying and traversing through the largest gaps in various window shapes, highlighting the potential of compact drones in complex navigation tasks.

B. Future Work

Looking forward, several enhancements and expansions are envisioned for this project:

- **Depth and Pose Estimation**: One of the primary areas for future development is the integration of methods for depth and pose estimation alongside optical flow. Current reliance on visual cues from a monocular camera limits the robustness of the navigation system. Implementing depth estimation algorithms would enable more accurate and safer navigation through complex environments. Techniques such as stereo vision or structured light approaches could be explored to provide the necessary depth information.

- **Sensor Fusion with Onboard Odometry**: Another critical enhancement is the fusion of camera data with onboard inertial measurement unit (IMU) sensors for improved localization. This sensor fusion would provide a more comprehensive understanding of the drone's position and orientation in space, facilitating navigation in truly unknown environments. By leveraging the IMU data, the system can be more resilient to visual ambiguities and environmental variations, paving the way for real-world applications beyond controlled settings.

VIII. Acknowledgment

The author would like to thank Prof. Nitin Sanket and the TA of this course RBE595.

References

[1] RBE595-Hands-On Autonomous Robotics Course Website**Link**

[2] DJITelloPy **Link**

[3] Optical Flow Estimation using a Spatial Pyramid Network **Link**

[4] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermüller and Y. Aloimonos, "GapFlyt: Active Vision Based Minimalist Structure-Less Gap Detection For Quadrotor Flight," in IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 2799-2806, Oct. 2018, doi: 10.1109/LRA.2018.2843445. **Link**