Machine Vision System for Optical Tracking of a Laser Power Beaming System

Sponsored by: Power Light Technologies
Industrial Mentor: Tom Nugent
Faculty Mentor: Professor Howard Chizeck

Acknowledgements: Professor Jengq-Neng Hwang and Professor Maolong Tang

Team: Qingrou Deng, Chenglong Li, Issac Huang

Introduction

Power Light Technologies develops technology for transmitting power via laser; commonly termed “laser power beaming.” One of the main commercial application is beaming power to unmanned aerial vehicles (UAVs).

Figure 1: UAV with mounted power receiver
Photo credit: Power Light Technologies

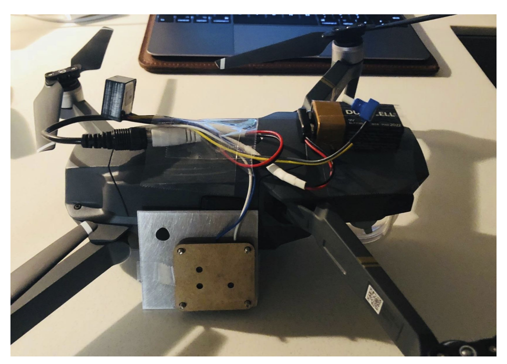


Figure 2: Project prototype, power receiver mounted on drone

Our objective is to develop a tracking system that aims the laser at the power receiver. The tracking system should have high accuracy and high update rate.

From the project specifications we have developed the following system requirements:

- ➔ Define effective object to track
- ➔ Detect and track target object
- ➔ Ensure high accuracy
- ➔ Obtain high frame rate (minimum 60 FPS)
- ➔ Ensure system is robust to a diverse set of background noises

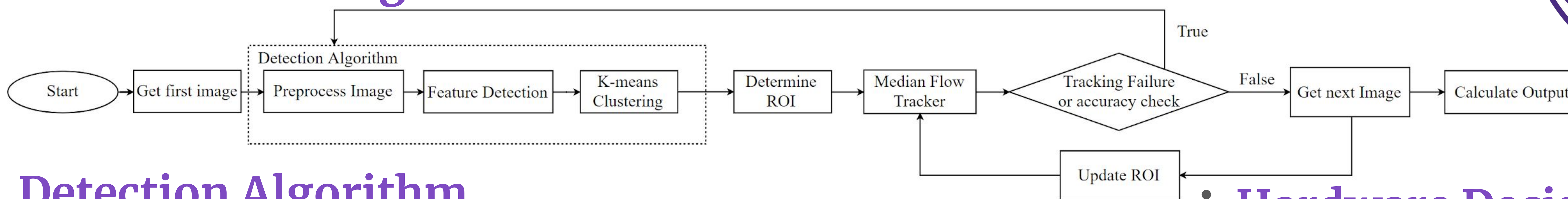
Technical Approach

Based on the project requirements the project is divided into four major portions, with each section further broken down into the following:

1. Hardware and software interface
   1.1. Obtain camera that meets specifications
   1.2. Control camera with Python
2. Detection algorithm
   2.1. Pre-processing image to remove noise
   2.2. Detecting target object
   2.3. Post-processing with K-means clustering
3. Tracking algorithm
   3.1. Find most robust algorithm
   3.2. Failure recovery
   3.3. High frame rate and accuracy evaluation
4. System Integration
   4.1. User interface
   4.2. Calculate required outputs

Please refer to the control flow diagram for more details.

Control Flow Diagram



Detection Algorithm

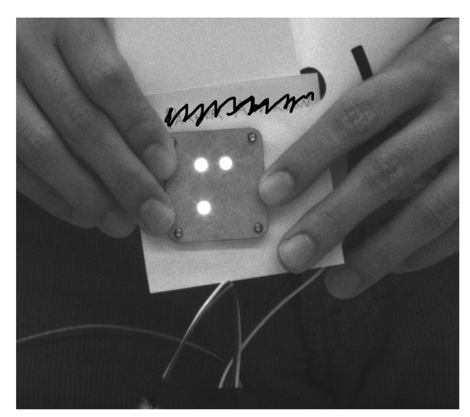


Figure 2.1: Original image

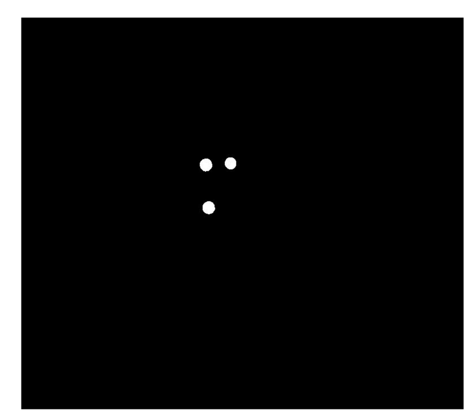


Figure 2.2: Result of Pre-processing; applied Median filter and Thresholding

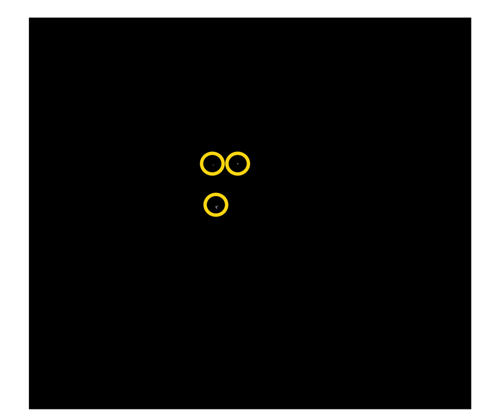


Figure 2.3: Result after applying erosion; beacons are reduced to few pixels in size

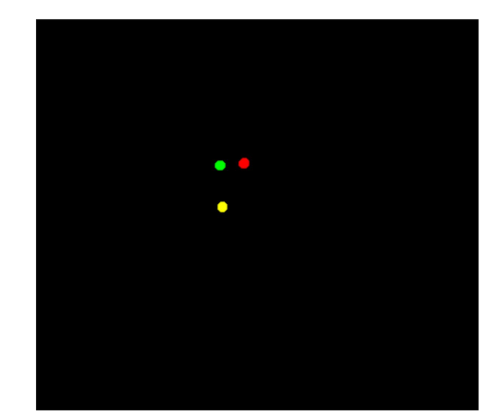


Figure 2.4: Result of applying K-means clustering on ORB key points; clusters are plotted in different colors

Pre-Processing Image

The original image is processed using the median filter, thresholding, and erosion.

- ➔ Thresholding the image to binary values removes all background noise with intensity values less than the beacons’ intensity.
- ➔ Median filter remove isolated pixel noises, similar to salt and pepper noise.
- ➔ Erosion reduces the size of the beacons and allows for accurate calculation of the beacons’ centers.

Feature Detection

ORB (Orientated FAST and Rotated BRIEF) method, developed by OpenCV, is used to detect feature points of the image after pre-processing. ORB fuses the keypoint detector of the FAST method and the descriptor of the BRIEF method. In the case of this project, the ORB method gave the best results because the ORB method was able to detect beacon sizes of a few pixels big while other feature detectors could not.

K-means Clustering

K-means Clustering is a basic machine learning algorithm. Here the algorithm is applied to analyze the key points returned by the feature detection method. The algorithm classifies each key point, based on the point’s x and y-values, to determine which beacon a key point is associated with. The results are used to calculate the center and radius of each beacon, which are sent to the tracking algorithm.

Tracking: Median Flow Tracker

Theory Overview

The goal of video tracking is to locate the target object over consecutive frames. The tracking algorithm used in this project is the Median flow tracker from the OpenCV library. The median flow tracker checks the target object’s motion both forward and backwards in time; based on these measurements the tracker minimize the forward and backwards errors to choose the most likely path of motion.

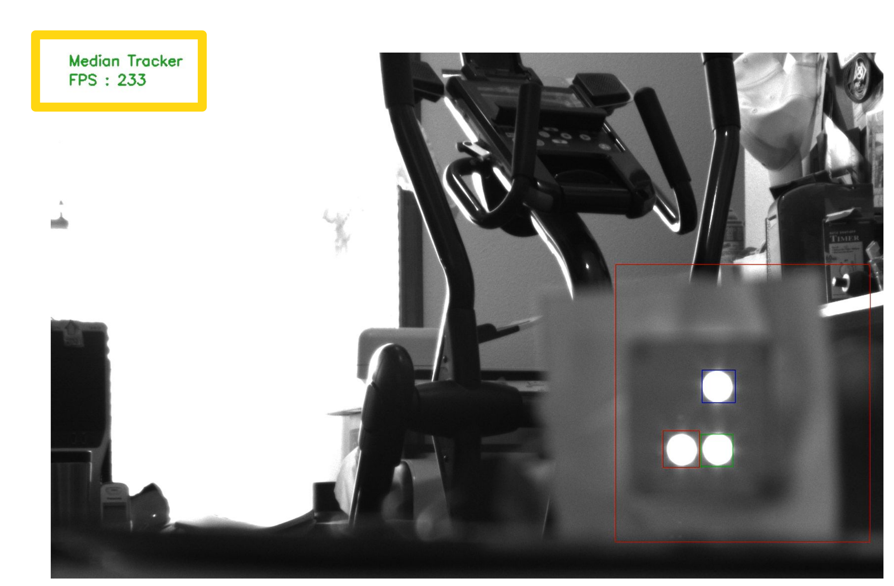


Figure 3: Median Tracker with ROI

Achieving High Update Rate

To reduce the complexity of the post analysis algorithm, used to calculate the center of mass, we decided to use three trackers instead of one; one tracker for each beacon. However, the processing FPS does not meet our requirements regardless of using one or three trackers. Therefore, to achieve the required processing FPS, we input a region of interest (ROI) around the beacons rather than the entire image. The processing rate increases significantly using this method; in the specific case of Figure 3 shown on the right, using the ROI method resulted in an average processing FPS of 214.4 while not using the method resulted in an average processing FPS of 12.4. The increase in processing rate depends on the parameters of the ROI and the size of the beacons; generally, processing frame rate decreases with increasing ROI size.

Hardware Decisions

Camera

The specifications required a camera with high resolution and high FPS. The camera chosen was the Grasshopper3 USB3 monochrome camera; the camera has the following relevant specifications:

Resolution	2048 x 1536
Megapixels	3.2 MP
Frame Rate	121 FPS
Sensor Type	CMOS
Readout Method	Global Shutter

Figure 4: Camera, lens, lens filter and camera mount set up

Other Hardware

- ➔ Variable zoom lens
- ➔ Lens filter
- ➔ Camera related accessories
- ➔ Mounting apparatus
- ➔ Beacon prototype

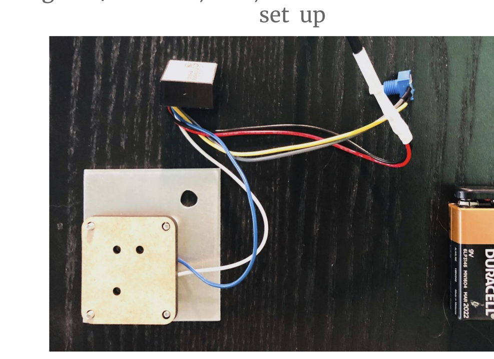


Figure 5: Beacon prototype, LEDs lights up red when turned on

Conclusion

What We Have Accomplished:

- ➔ Output beacon information from a high frame rate, high resolution input
- ➔ Integration of camera, detecting and tracking programs

What We Have Learned:

- ➔ Tracking theory
- ➔ Image processing theory
- ➔ Machine learning concepts
- ➔ Hardware–software integration
- ➔ Communication
- ➔ Project management
- ➔ Effective work division
- ➔ Time Management

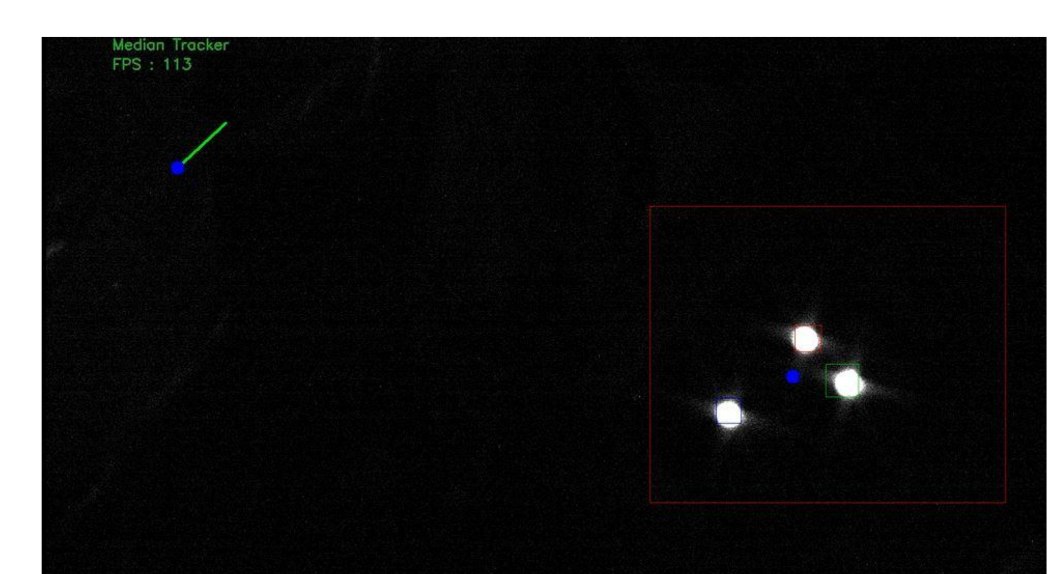


Figure 6: The video result shows the beacon location, beacon center and beacon orientation.