Ground Plane Based Autonomous Navigation of Micro Air Vehicle

Group 7: Deepak Paramkusam, Xinyu Gao, Gokul Nayar, Ziqi Wang, Mingzhou Yin, Yanggu Zheng
Technische Universiteit Delft, 2600 AA, Delft, The Netherlands

ABSTRACT

The camera is one of the most used types of sensors for obstacle detection and robot navigation. Due to extremely limited resources, it is specially important for micro air vehicles (MAV). This article illustrates a strategy for autonomously navigating Parrot Bebop[1] (can be considered as a robot / agent) using vision algorithms with single front camera based on Paparazzi[2] and OptiTrack[3]. Different from traditional obstacle detection strategies (e.g. optical flow), performed in this specific project was an approach inspired by road obstacle detection using ground plane construction to separate objects. Although only a single camera is available, the theory of 3D stereo construction was adopted to perform the object identification. It was proved that with this algorithm, a MAV can identify the ground plane and obstacles placed directly on it. Thus, an autonomous flight can be achieved.

1 INTRODUCTION

Computer vision based algorithms are developed to an extent that they have already been widely applied to drone navigation to avoid obstacles. However, most of the applications are constricted to specific scenarios. Most commonly used techniques of obstacle classification are briefly explained in Section 2.

In this article, a new obstacle avoidance technique is presented in which the ground plane is used as a decision boundary to check whether and where there are obstacles in front of the MAV. Furthermore, the algorithm has to decide whether the distance between them is safe and change its heading when necessary.

The rest of the article organizes as follows. Next section of this article provides general introduction to related existing techniques and their trial implementation in this project. Section 3 and Section 4 discusses the use of ground based vision algorithm in this experiment setup. The final result is shown in Section 5 followed by discussion and conclusion.

2 EXPLORED TECHNIQUES

2.1 Optical Flow

Optical flow is a term to describe how surrounding moves with respect to the agent itself. In terms of obstacles avoidance, the quantity of divergence or time-to-contact is of utmost interest. The divergence for a linear optical flow field is defined as Equation 1, where (u, v) is the velocity at (x, y), (x_0, y_0) is the focus of expansion. The time-to-contact is defined as the reciprocal of divergence. In general, a too large divergence for a point indicates a close obstacle at that point.

$$\text{Divergence} = \frac{u}{x - x_0} = \frac{v}{y - y_0} \tag{1}$$

The optical flow module has been implemented in Paparazzi. First, FAST9 corner detection algorithm is applied on the image to extract corners. Then, Lucas-Kanade optical flow algorithm is applied for the corners to obtain their flow vectors. The following algorithm is used to calculate the divergence. First, the point with minimum magnitude of flow vectors is specified and considered as the focus of expansion. Then, the overall divergence value is calculated as the average value of divergence at each corner defined as Equation 2, where v_i is the flow vector of the i^{th} corner, and r_i is the position of the i^{th} corner with respect to the focus of expansion.

$$\text{Divergence} = \frac{\|v_i\|}{\|v_i + r_i\| - \|r_i\|} \tag{2}$$

However, target obstacles for this project typically do not contain much good corners for optical flow calculation as shown in Figure 1. Furthermore, the venue of the project is surrounded by protection nets which provides a lot of false corners. As a result, the divergence value is too noisy for practical use.

2.2 Contour Detection

Contour detection is a simple object detection method based on color. It involves drawing a curve around all pixels having similar color or intensity. Detected pixels close to each other are assumed to belong to the same object. Different types of curves can be approximated around these pixels. The curve that fits the best is inferred to be the shape of the object as shown in Figure 2. *findContours* function of OpenCV was used for this method. An implementation of

[1] Parrot Bebop is a lightweight commercial MAV equipped with a full HD front camera.

[2] Paparazzi is an open-source drone development platform, which will be explained in Section 4.

[3] OptiTrack is a 3D tracking system in lieu of GPS in an indoor setup, which will be explained in Section 4.

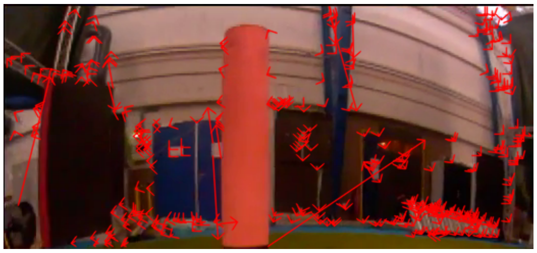

Figure 1: Illustration of detected flows for the optical flow method.

Douglas-Peucker algorithm is used for the contour approximation. Once the contour is found, its area can be used to estimate distance to the object.

The disadvantage of this method is that objects of only a single color can be detected at a time. Also, it has many false positives if the required color is common in the environment. In cyberzoo for example, black objects were difficult to detect.

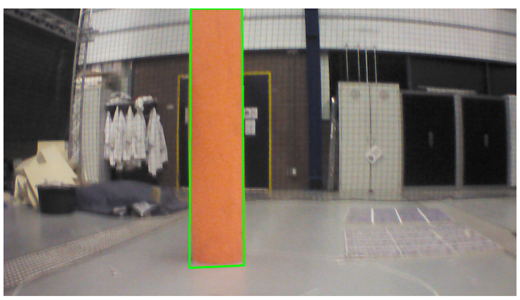

Figure 2: Object detection using contours.

2.3 *Scale-Invariant Feature Transform (SIFT)*

SIFT is a standard local edge detection technique developed by David Lowe at UBC. The algorithm involves five steps - scale-space extrema detection, keypoint localisation, orientation assignment, keypoint descriptor, and keypoint matching. First keypoints from the training image are generated that are invariant to scaling, orientation, distortion and illumination. Then the keypoints are matched in the query image.

The algorithm is already implemented in OpenCV. *SIFT_create* and *detectAndCompute* functions of OpenCV generate the keypoints. Matching is done using *drawMatchesKnn*, which uses k-nearest neighbor search.

When tested for the project, it was noticed that the main obstacles (the orange pillars) in the cyberzoo, did not have enough keypoints for accurate detection. This is hypothesized to be due to the smooth texture of the obstacles. This made obstacle detection difficult. Also, multiple irrelevant keypoints were generated due to the net surrounding the cyberzoo. It was also noticed that even though SIFT accurately detects the obstacle, it gives no information about how far the object is from the drone.

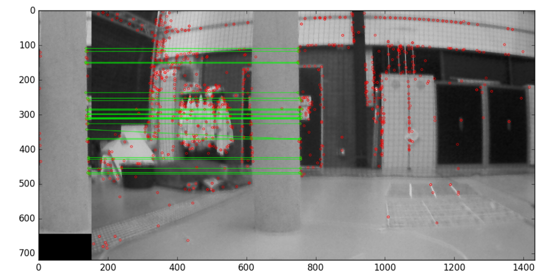

Figure 3: SIFT matching with training image on the left and query image on the right.

2.4 *Support Vector Machines (SVM)*

SVM is a traditional machine learning theory performed basically in binary-class classification problems. One of its typical applications is pedestrian recognition where pedestrians can be classified as obstacles while others are assumed to be non-obstacles. SVM can be used as either supervised learning approach or semi-supervised (Transductive SVM) classifier or even unsupervised clustering.

The main problem of implementing SVM on MAVs is that it requires significant pre-training before real-time operation which can be considered as testing procedure from the view of machine learning. However, sufficient training datasets are not available for this project as well as most real-world scenarios.

2.5 *Simultaneously Localization and Mapping (SLAM)*

SLAM aims at constructing or updating maps around the agent and keeping track of the agent at the same time. Kalman filtering and particle filtering are two main statistical approximation approaches involved in SLAM. SLAM is useful especially under the circumstances where GPS cannot be used to locate the agent.

The primary difficulty of performing SLAM on MAVs resides in the fact that SLAM demands a large storage and computation capability which is not available on MAVs.

3 Ground Plane Based Vision Algorithm

3.1 *Theory*

The ground plane is where the objects are located and the unoccupied region on the ground plane is called free space. When an object is placed on the ground plane, a boundary lower than the horizon emerges on the ground plane. The depth of the boundary from horizon indicates its distance from the front camera. The farther the object is located, the closer the boundary is to the horizon for the same image taken by the camera. As an illustration, objects are marked by stixels on Figure 4, whereas the ground plane is clear without marking.

The aforementioned theory can be applied to detect objects in front of MAV. In this project, a color-based approach was used to identify the ground plane and boundaries of obstacles. Images taken by the front camera were split into YUV422 true color space in this project. It is assumed that the color of the ground plane is coherent, which is reasonable for most of practical cases like the asphalt road, dirt road or even sky. In the venue of this project, the ground plane is green. Coloring filtering was then applied to separate the ground plane from the obstacles.

After separating the objects from the ground plane, the height of the lowest boundary of objects on the image can be decided by scanning each line from the bottom. When the amount of ground plane pixels is lower than a threshold value, a boundary resulted from an obstacle that is too close to the MAV is detected. Then, the horizontal position of the obstacle is obtained by calculating the center-of-mass of the non-ground-plane pixels. In this way, a tri-value output can be generated from this algorithm, namely -1 for close obstacles on the left, 0 for no close obstacles, and 1 for close obstacles on the right.

Figure 4: Illustration of the ground plane theory.

3.2 *Algorithm*

The algorithm applied to process image is described in Figure 5. The code runs at a certain frequency to update the command for the MAV.

4 EXPERIMENT SETUP

4.1 *Operation Platform*

This project was implemented with Paparazzi platform and OptiTrack system based on an existing module `orangeAvoider`, which detects orange by a naive color filter. The Paparazzi platform includes the following layers of codes. 1) Airframe `XML` file: control basic settings of the MAV. In this project, front camera settings were altered to display more pixels on the ground for better detection of the ground plane. 2) Flight plan `XML` file: control basic flight routine of the MAV. Since a similar control strategy as `orangeAvoider` was applied based on the ground plane identification result, no change is made to the flight plan file. 3) Module `XML` file: compile and call related C-functions. The refresh rate of the algorithm is determined by the running frequency of the main control program specified in module files. 4) Main control program: Calculated the desired moving heading and distance. This part is similar to

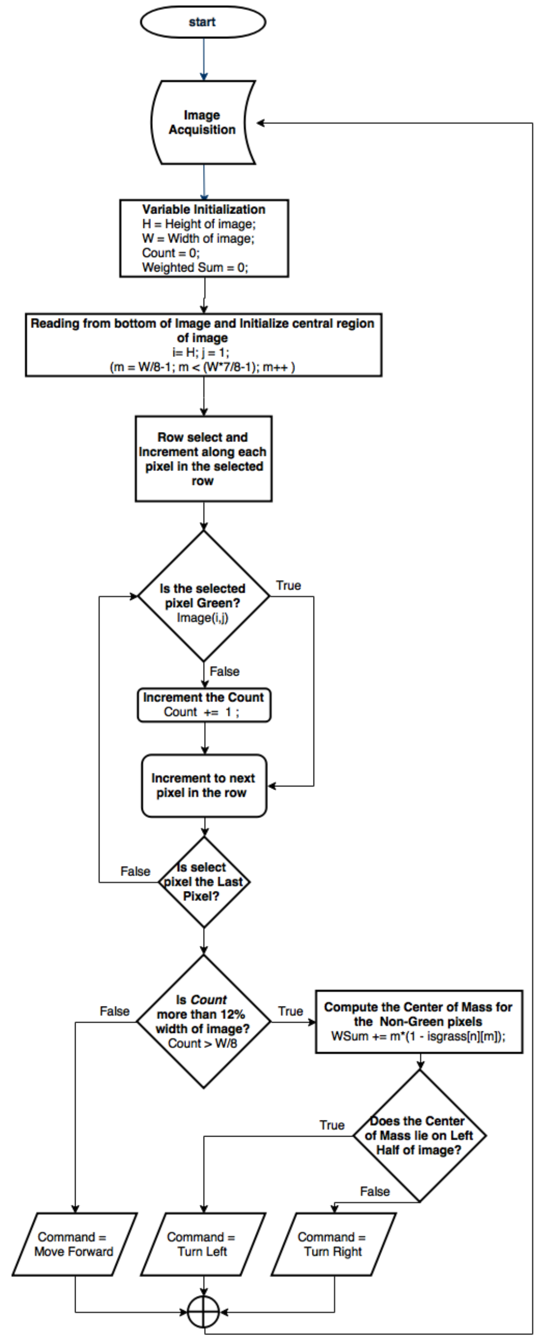

Figure 5: Flow chart of the ground plane algorithm.

the original `orangeAvoider` except that the heading is not randomly chosen. 5) Ground plane detection program: gives information about the position and distance of the obstacle. Details of the algorithm are introduced in Section 3.2. The overall hierarchy of the platform is illustrated in Figure 6.

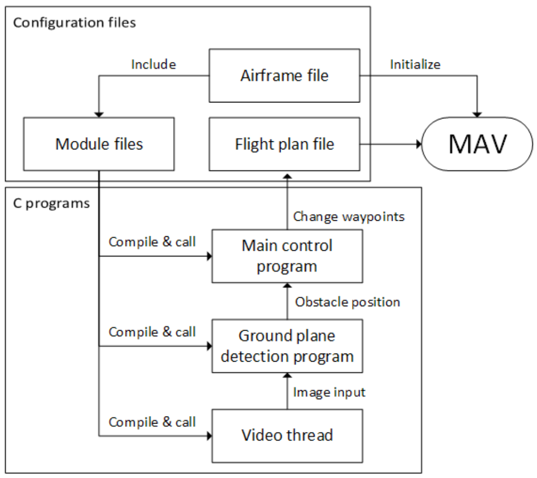

Figure 6: Hierarchy of the Paparazzi platform.

The OptiTrack system provides indoor 3D tracking of MAV's position. Therefore a waypoint-based control strategy can be applied, i.e. the movement of the MAV is controlled by moving targeting waypoints with desired distances and headings. Inner loops of the MAV were not directly controlled.

4.2 Obstacles

Obstacles used in this experiment are variant. The main obstacles adopted were orange pillars. Red frame and blackboard were also used during the competition. In particular, There were also textured cloth covers of different colors on the top of orange pillars and the blackboard. Besides all of these obstacles, a flag with the name of the course was set in the zone as well.

It can be seen that the obstacles are of different colors and shapes. Therefore, algorithms based on characteristics of obstracles are not appropriate like contour detection based on color. Alternatively, the ground color in cyberzoo is uniform which is useful for ground plane detection.

5 Result

In the beginning, the optical flow method was used to detect the obstacle. However, it did not work well as expected. The protection net of the cyberzoo has plenty of corners while the obstacles has very few corners, which makes the MAV cannot distinguish the obstacle from the environment. As shown in Figure 1, there are too much noise that makes the result unreliable.

The ground plane based vision algorithm gave a better performance in the cyberzoo which can be seen in Figure 7. The line clearly shows the contacting boundary between the obstacle and the ground plane. The green circle is the decision part which guides the MAV to go forward. The arrows pointing to the left or right will gave the decision that the MAV should turn left or right.

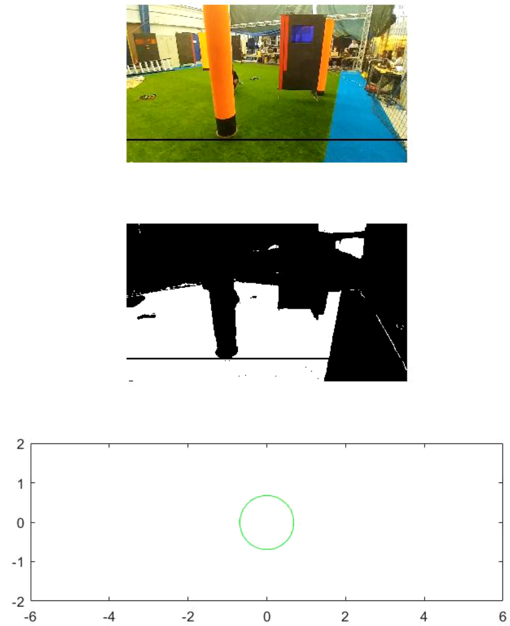

Figure 7: Detected boundary of obstacles with the ground plane.

On top of the detection algorithm, the performance is also affected by parameter tuning. The values of the various parameters of the control strategy were tuned according to test flights. Table 1 shows the values of the parameters applied in the final competition.

Parameter	**Value**
Flight height	1.5m
Maximum velocity	8m/s
Position resolution	0.05m
Yaw rate	40°/s
Yaw resolution	10°
Refresh rate	4Hz

Table 1: Applied parameters in the competition.

6 Discussion & Conclusion

The ground plane based vision algorithm works better than optical flow and other algorithms explored so far in the cyberzoo. However, it is realized that the disadvantages of this algorithm lies in the fact that the color of the ground can be variant instead of a single color outside the cyberzoo.

In the future, improvements can be made to enhance performance of the algorithm. First, the bottom camera can be used to detect the color of the ground plane before the boundary of obstacles is detected, which makes the algorithm capable to handle more scenarios. Another way to detect the obstacle is to use stereo cameras to rebuild the obstacle which can be more reliable.