Navaminda Kasatriyadhiraj Royal Thai Air Force Academy

(PEGASUS01)

1.Object Detection, Classification, and Localization:

1.1 System design:

- The team designed this part with the following steps in order.

1.1.1 Object Detection:

- **Color Separation**: Images are resized to reduce processing time. K-Means clustering is used to separate colors into 3 clusters.
- **Background Removing**: The largest color cluster, defined as the background, is filtered out.
- **Target Detection**: After background removal, a contour is drawn within a specific area range, and unwanted noise is filtered out.

1.1.2 Object Classification:

- **Shape Identification**: The contour area of interest is cropped and fed into a pre-trained deep learning algorithm for shape identification.
- **Alphabet Analyzation**: The K-Means algorithm is used to extract alphanumeric characters from the identified shapes. These characters are rotated by 15 degrees and fed into Tesseract OCR (Tesseract is particularly powerful because it can recognize text in various languages and is highly accurate) iteratively until the highest confidence result is obtained, determining the answer alphabet and angle.

1.1.3 Localization:

- **Synchronizing Image Location:** This step is about making sure that the location of the image you have captured matches the real-world location where the image was taken. It is like tagging a photo on a map.

  The image location is synchronized using the camera's hot shoe feedback.

- **Ground Sample Distance (GSD):** GSD is like a ruler for your image. It tells you how big each pixel in the image is in real life. If you know the GSD, you can figure out how far away objects in the image are from the camera. The equation is as follows:

$$GSD = \frac{SW \cdot H}{F \cdot I} \quad (1)$$

,where H = Altitude (inch),
SW = Sensor Width (inch),

I = Image Width (Pixel),
F = Focal Length (inch)

- **Object Bearing Angle:** This is the angle that tells you in which direction an object is located in relation to the camera. The equation used is:

$$AngleofBearing(\theta B) = \psi + \tan^{-1}\left(\frac{Xlength}{Ylength}\right) \quad (2)$$

, where ψ: vehicle's heading

- **Object's Coordinates:** This is like finding the GPS coordinates (latitude and longitude) of the object in the image. It is a way to pinpoint exactly where the object is on Earth's surface.

$$lat2 = \sin^{-1}\left(\sin(lat1)\cdot\cos\left(\frac{L}{R}\right) + \cos(lat1)\cdot\sin\left(\frac{L}{R}\right)\cos(\theta B)\right) \quad (3)$$

$$lon2 = lon1 + \tan^{-1}\left(\frac{\sin(\theta B)\cdot\sin\left(\frac{L}{R}\right)\cdot\cos(lat1)}{\cos\left(\frac{L}{R}\right)-\sin(lat1)\cdot\sin(lat2)}\right) \quad (4)$$

, where L = Distance Between object and image center
R = Earth Earth's Radius
Lat 1, Lon 1 = Latitude and Longitude of an image center
Lat 2, Lon 2 = Latitude and Longitude of an object

1.2 Testing:

The team tested each subsystem through 5 phases.

- Phase 1: Included testing the background and object detection.
- Phase 2: Test shape and color identification.
- Phase 3: Test alphanumeric identification.
- Phase 4: Test object geolocation.
- Phase 5: Full mission test.

2.Mapping:

2.1 System design:

- **Map Template Creation**: A blank map template is created with a 16:9 aspect ratio, and its dimensions are calculated based on specific mission parameters. These parameters include the map's center coordinate and its height.

- **Image Pre-processing:** To start, we gather all the photos and their accompanying information, organizing them neatly within a dedicated folder. Then, we perform a transformation, converting the geographic locations of these photos (specified by latitude and longitude) into pixel positions on our map using specialized mathematical formulas. Additionally, we ensure that the direction of each photo aligns correctly, adjusting it based on the UAV's heading, which indicates the angle relative to North. To maintain uniformity, we fine-tune the size and enhance the sharpness quality of the photos, conforming them to predefined standards.
- **Image Stitching**: The pre-processed images, each having its own coordinates, direction, and height information, are then stitched onto the created blank map. This step involves aligning and overlaying the images onto the map in a way that accurately represents their respective positions.

2.2 Testing:

The 5 phases considered in this part are:

- Phase 1: Develop mapping algorithm.
- Phase 2: Test algorithm with the dataset from Google map.
- Phase 3: Collect the aerial images by UAV and process with the algorithm.
- Phase 4: Improve algorithm efficiency.
- Phase 5: Full mission test.

3.Autopilot:

3.1 System design:

- **Choice of Autopilot**: The team opted for the Pixhawk 2 (CUBE) as their autopilot for autonomous flight missions.
- **Compact Design**: The Pixhawk 2 (CUBE) houses both the Flight Management Unit (FMU) and Inertial Management Unit (IMU) within a relatively small form factor, which is advantageous for the setup.
- **GPS Modules:** Instead of using a Real-Time Kinematic (RTK) positioning system, the team decided to use multiple GPS modules. This choice likely simplifies setup and provides adequate positioning accuracy.
- **Firmware:** The autopilot runs on ArduCopter v4.0.6 hexa firmware.
- **Communication Protocol**: MAVlink serves as the communication protocol that enables data exchange between the autopilot, an onboard processing unit, and the ground control station.

3.2 Testing:

The team considered the following 5 phases of testing in autopilot part.

- Phase 1: Test positioning system precision.
- Phase 2: Integration test with avoidance algorithm.
- Phase 3: Improve waypoint navigation performance.
- Phase 4: Test autonomous flight with uncertainties.
- Phase 5: Full mission test.

4.Obstacle Avoidance:

4.1 System design:

The algorithm used is divided into two parts. The first part uses a Potential Field algorithm inspired by how water flows downhill. It considers the starting point and obstacles as high places and the target waypoint as a low place, ensuring the UAV avoids colliding with obstacles. To enhance this, a custom "Ometum" algorithm was developed by the team to prioritize finding the shortest path to avoid the nearest obstacle. The Potential Field algorithm plans a safe flight trajectory and includes four additional routes before Ometum takes over during the mission. Ometum actively avoids obstacles within about 656.2 feet using specific processes, as shown in the figure.

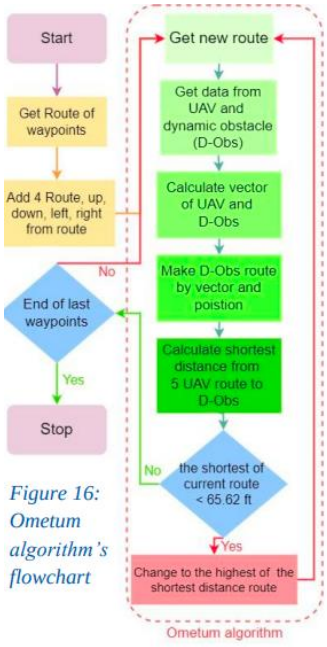

Figure 16: Ometum algorithm's flowchart

4.2 Testing:

The 5 phases of testing here are:

- Phase 1: Simulate static avoidance algorithm.
- Phase 2: Simulate dynamic avoidance algorithm.
- Phase 3: Integrate and simulate both algorithms.
- Phase 4: Test obstacle avoidance on the prototype UAV.
- Phase 5: Full mission test.