SAFMC 2024

Final Presentation | CAT E | Team DOTA7, NYP

01

Introduction

The Team

Team DOTA7, NYP

Sanjeev

General

Armando

Commander

Lynuz

Camera Specialist

Ashween

Design & Media Specialist

Rishi

Hardware Specialist

Aldrin

Logistics Specialist

Pierce

HR Specialist

02

Hardware

- The Sensor Suite
- Design Rationale

The Sensor Suite

Each drone is equipped with a:

1. Camera
2. Infrared Sensor
3. Inertial Measurement Unit(IMU)
4. ESP32-CAM

1. Drone to **hover in place**
2. Fly indoors more **precisely**
3. Aids with **obstacle avoidance**, **localisation** & **victim detection**.

Our Drone

Design Rationale

Factors we considered as we iterated on our design:

1. Weight
2. Temperature
3. Battery life

Final mount!

Evolution of our ESP32-CAM mount

03 Software

- Localisation Process
- Obstacle Avoidance
- The A* Algorithm

The Localization Process

3-Step Localization Process

01

Inertial Measurement Units (IMU)

02

Radio Signal Strength Indicators (RSSI)

03

Visual Localization

3-Step Localization Process

Inertial Measurement Unit

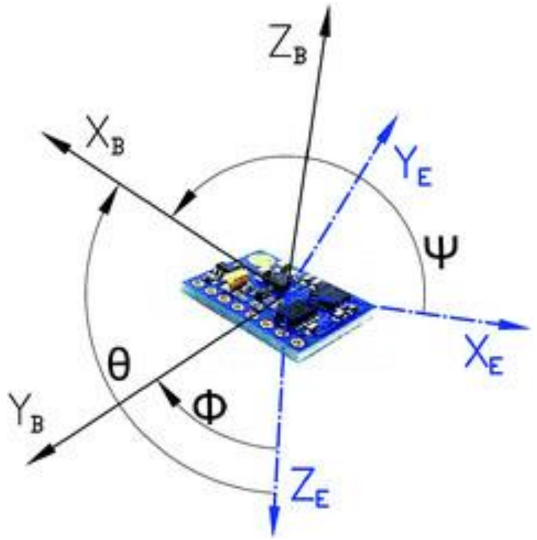
01

02

03

Using the IMU:

1. **Sensor fusion** to calculate the yaw, pitch and roll angle
2. **Double integration** to calculate distance
3. **Measure the distance moved** in the x-axis and y-axis



3-Step Localization Process

RSSI Localization

Why use **BLE (Bluetooth Low Energy)** for RSSI localization?

- **Wi-Fi RSSI** is very noisy and easily affected by environmental factors
- **GPS** does not work indoors and has too much error (about 1-5 km)

Bluetooth Low Energy

01

02

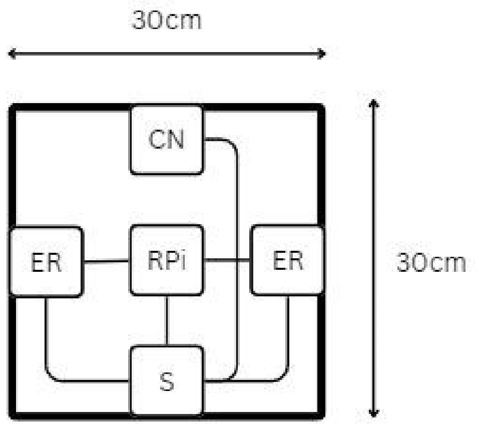
03

3-Step Localization Process

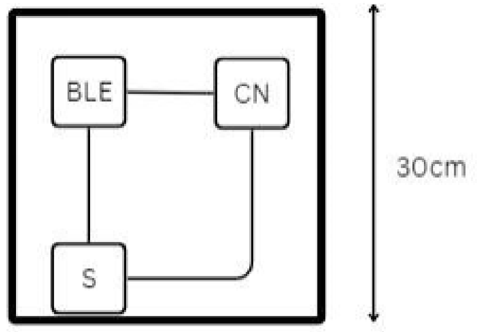
RSSI Localization

To assist our localisation:

1. Created a few **nodes & bases**
2. Base station consists of an ESP32-WROOM **emitting BLE signals**
3. Calibration node to calculate **environmental constants and other components**
4. Consists of **3 ESP32-CAMs receiving RSSI signals** from the drone



Localization Server



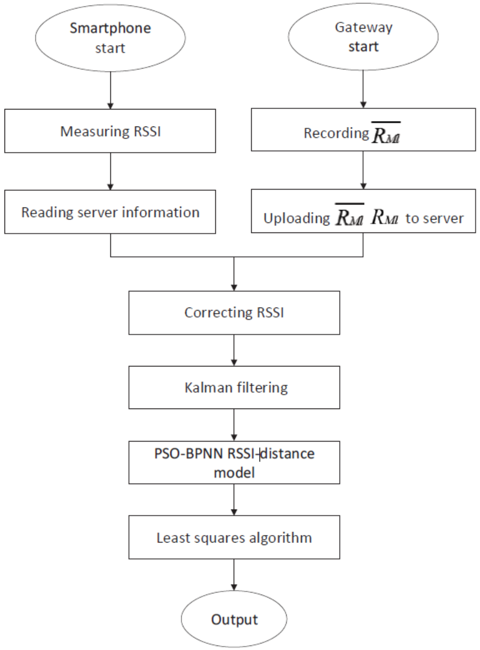
Base Stations

3-Step Localization Process

RSSI Localization

Finally, the **localization process**:

1. Signals are connected to a **Raspberry Pi 5**, which runs an **RSSI correcting algorithm**
2. Next, through a **Kalman filter** and is fed into our artificial neural network
3. **Backpropagation** optimized with a **particle swarm algorithm** to give us the final distance



Localization Process

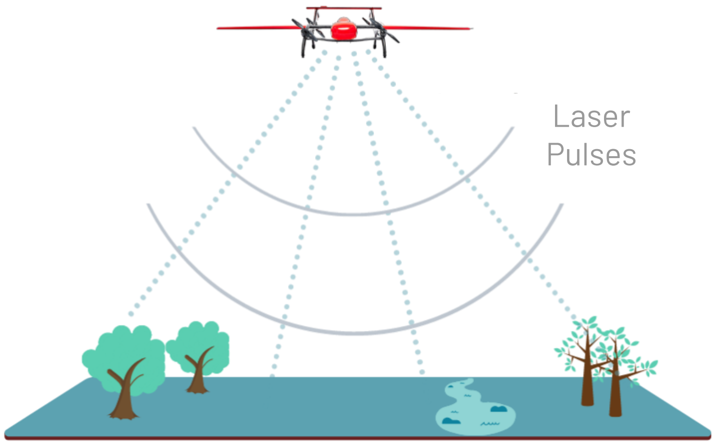
3-Step Localization Process

Visual Localization

Before the drone begins taking pictures:

1. **3D mesh render of the arena** will be taken with **LIDAR**
2. Using **Open3D**, a virtual camera creates a **8 images of each coordinate point** in the arena

Light Detection and Ranging



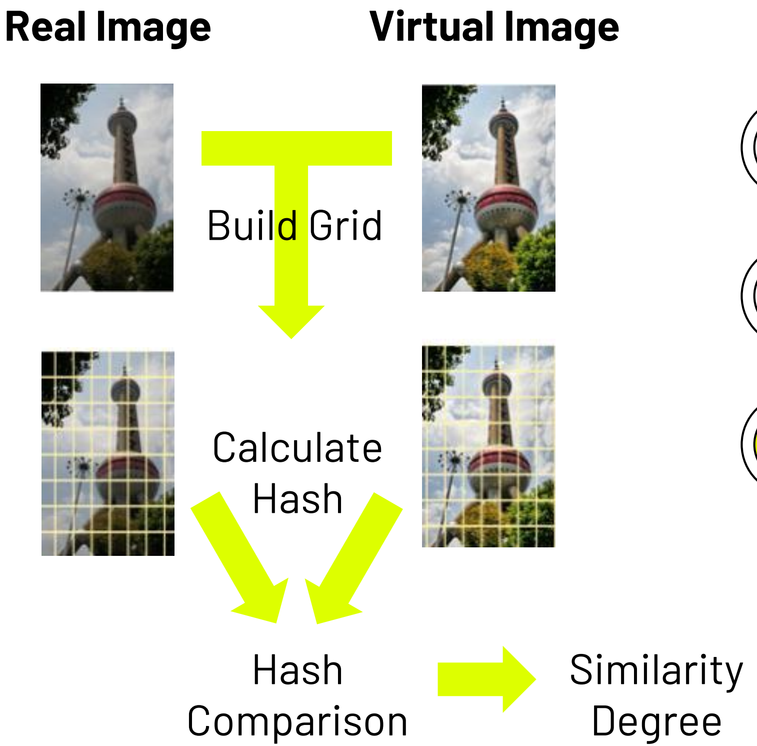
OPEN3D

01
02
03

3-Step Localization Process

Visual Localization

Perceptual Hashing



01
02
03

Before the drone begins taking pictures:

3. Using **perceptual hashing**, both images' coordinate points are compared to **determine the accurate location of the drone**
4. Once determined, the real-life image replaces of the coordinate point of the virtual images

3-Step Localization Process

Visual Localization

01
02
03

3 different localization modes
in the Visual Localization:

1. Left-Right
2. Front-Back
3. 360 Degree Localization

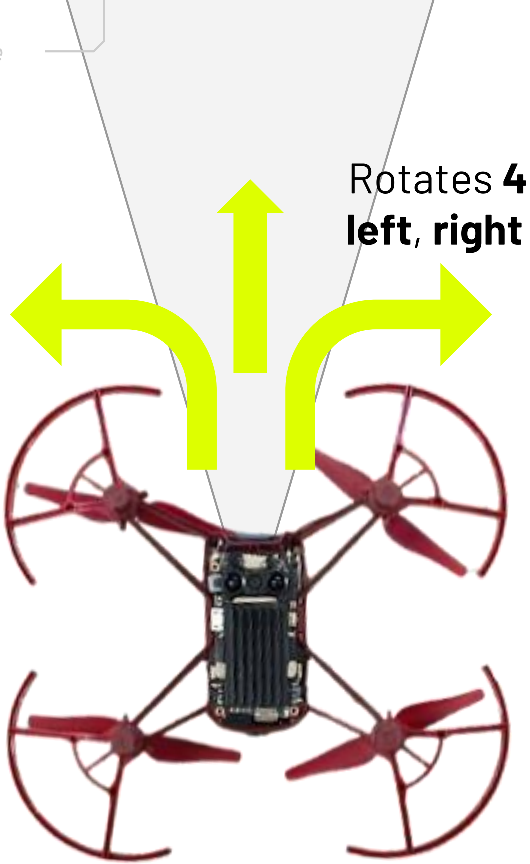
3-Step Localization Process

Visual Localization

3 different localization modes in the Visual Localization:

1. Left-Right
2. Front-Back
3. 360° Localization

Rotates **45°**, facing **left**, **right** and **front**



Takes **3 photos**, 1 at each orientation

01
02
03

3-Step Localization Process

Visual Localization

3 different localization modes in the Visual Localization:

1. Left-Right
2. Front-Back
3. 360° Localization

Rotates **180°**, facing **back** and **front**

Takes **2 photos**, 1 at each orientation

01

02

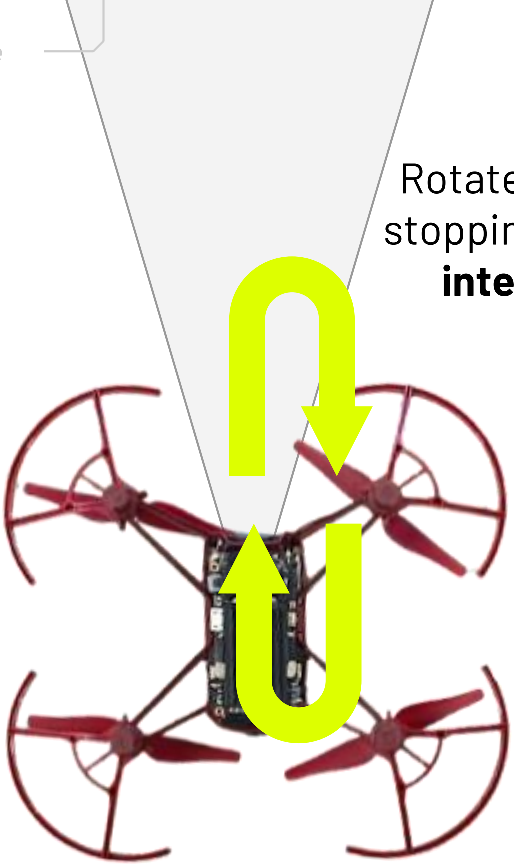
03

3-Step Localization Process

Visual Localization

3 different localization modes
in the Visual Localization:

1. Left-Right
2. Front-Back
3. 360° Localization



Rotates **360°**, stopping at **45° intervals**

Takes **8 photos**, 1 at each orientation

01

02

03

Obstacle Avoidance

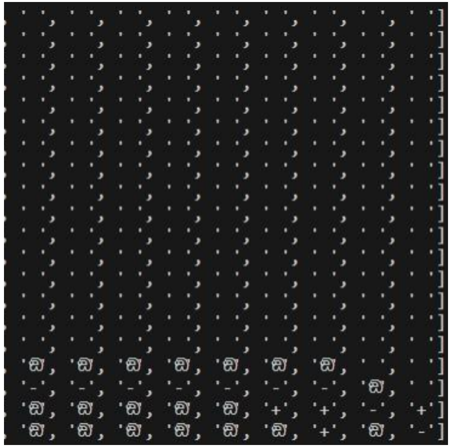
1. The drone takes an **image of its surroundings**
2. Calculates the estimate depth of the obstacles around it using **Zoe depth monocular depth estimation**, an Artificial Intelligence model
3. The AI takes the image, divides it into 3x3 sections and checks the **average depth and proximity** of each section
4. **1 central server laptop** is used to control 4 drones that **divides the computational load** allowing them to be **run simultaneously**

Depth Map

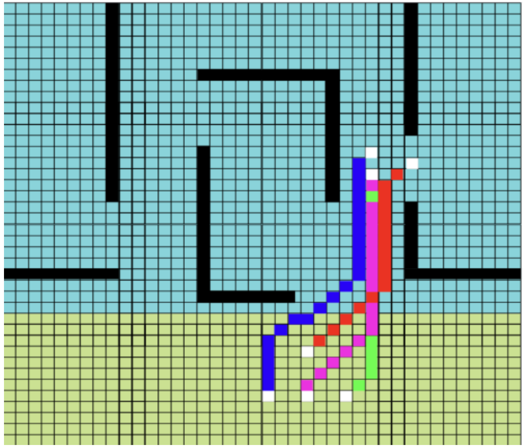
The A* Algorithm

A **pathfinding** algorithm
- Find the **shortest or most efficient route**

A star algorithm uses open set and close set
- Open set to explore
- Close set not explore



Prototype Code Output

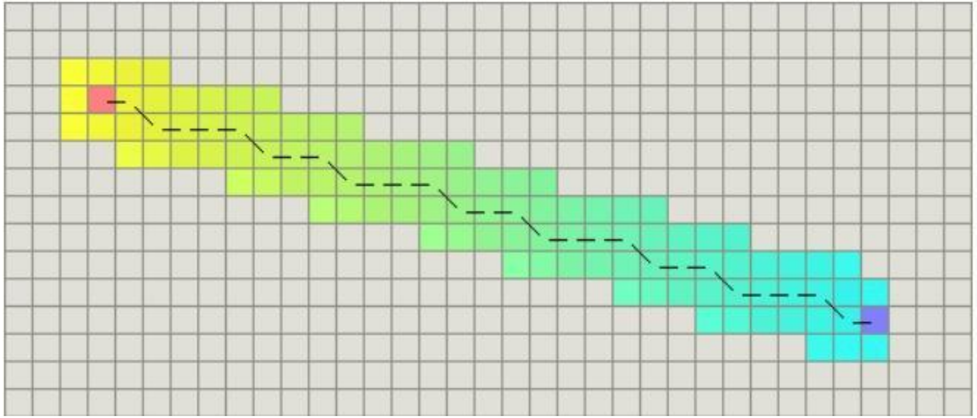


Final Code Output

The A* Algorithm

1. A **pathfinding** algorithm
2. Find the **shortest or most efficient route**

Combines a cost function that measures the actual path cost (g(n)) with a heuristic function (h(n)) that estimates the remaining cost to the goal. This combination guides the algorithm efficiently.



Heuristics

04

Mission Plan

- Search Strategy
- Robust Intelligent Swarm Control
- Other Strategies
- Challenges & Lessons

Search Strategy

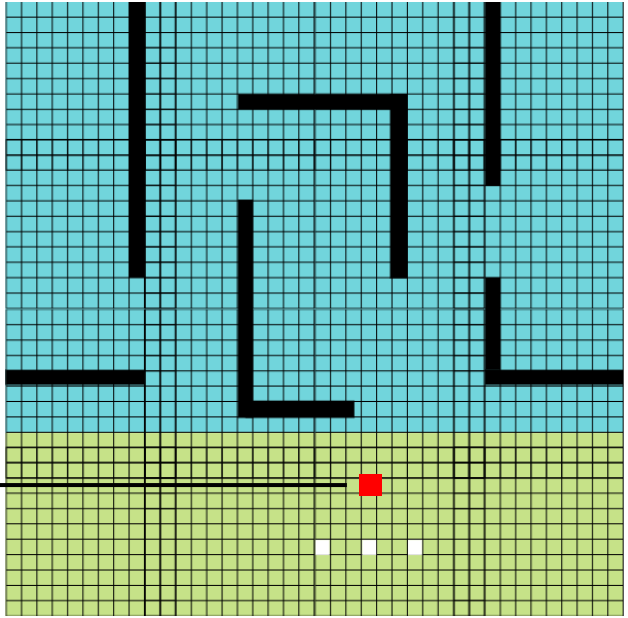
Color Coordinated Victims in the Server

Double Rescue

Single Rescue

01 Divided the map into **4 sectors**

02 **1 scout drone** from each group of drones

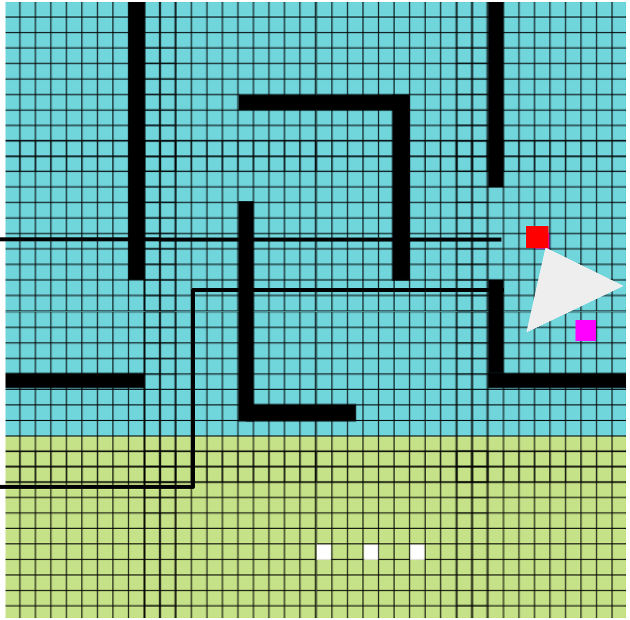


The Mission Map

Search Strategy

03 **Localising** and confirming that the drone is in the room, it **descends** and **looks for the victims**

04 It calculates the **distance to the victim** and the **drone's orientation** using the **magnetometer**. With Trigonometry, the victim coordinates are calculated & relayed to the server, to ascertain its status



The Mission Map

Search Strategy

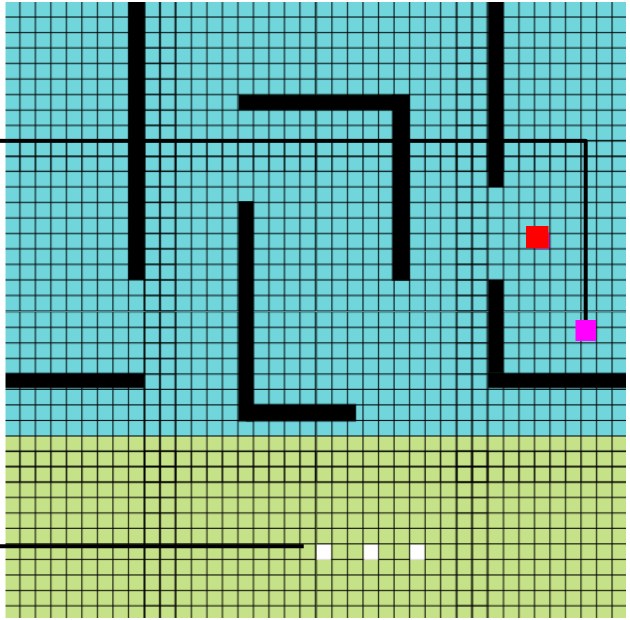
05

If victim status is **"unknown"**, the status is updated to **"found - drones required"**

06

A rescue drone informs the server that it is on the way

*If it is a **double rescue victim**, the server ensures another drone is routed to the victim*

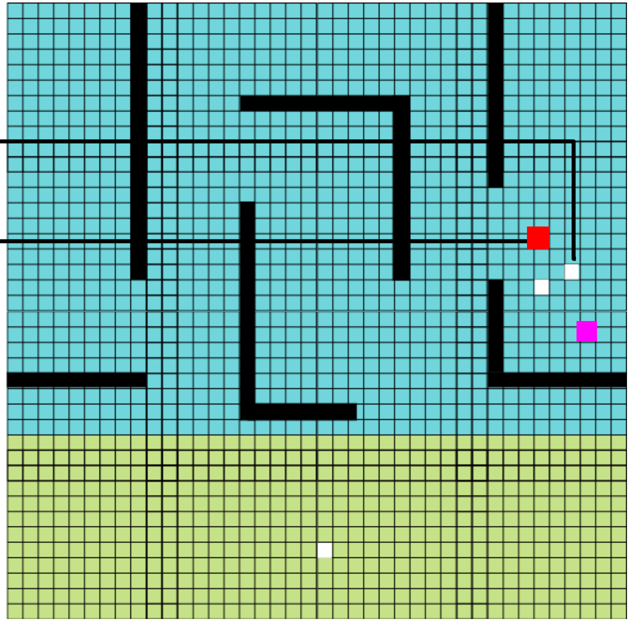


The Mission Map

Search Strategy

07 At the victim, the rescue drone localises before landing and **updating the server on the relevant status** of the victim

08 Until the victim's status is **NOT "found – drones required"**, the server checks for assigned drones to the victim. Otherwise, the search continues!



The Mission Map

Robust Intelligent Swarm Control

1. **Four laptops** to run 16 drones
2. Another laptop as a **centralized server**
3. Specifically chosen for its **beam forming technology** and it **AI traffic optimiser**

Routers

1. Greatly **reduced computational power**
2. To communicate with all the drones **simultaneously**
3. **Improve the latency** and **reduces data loss**

Reducing Mission Time

1. Leveraging **multiprocessing** and **multithreading** to run **localization**
2. IMU data processing and obstacle avoidance **concurrently**
3. **Optimized pathfinding** using A* algorithm uses more time-intensive localizations less frequently

Parallel processing

Efficient data handling

Faster decision making

Balance precision with speed

Other Strategies

Using a **LiDAR camera**:

1. **Scan rooms in advance** to acquire environmental data
2. SLAM system to **construct a detailed virtual model** for A* pathfinding
3. **Chilling the mission pads** used at takeoff points

Chilling the mission pads

Challenges & Lessons

Challenges Faced

1. Technical hurdles
2. Design optimizations

Lessons Learnt

1. Pre-scanning environments with LIDAR
2. Chilled mission pads for drone cooling

Thank you!

Final Presentation | CAT E | Team DOTA7, NYP