

# Distributed Robotics



Summer 2017

Anissa Alexander, Stirling Carter, Tim Liang,  
Christina Wang, Austin Wilms, Mark Yang



# Today's Schedule

1. Introduction to what we've been working on
2. Video
3. Demos and discussion
4. Short break
5. Programming robots with Python
6. Q&A time

# 2017 Summer Research

- Institute for Software Integrated Systems
  - VeriVITAL - safety with cyber-physical systems
- Summer research program
  - StarL - distributed robotics platform



Image: BBC

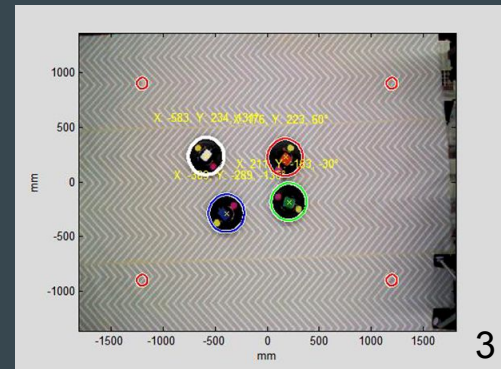
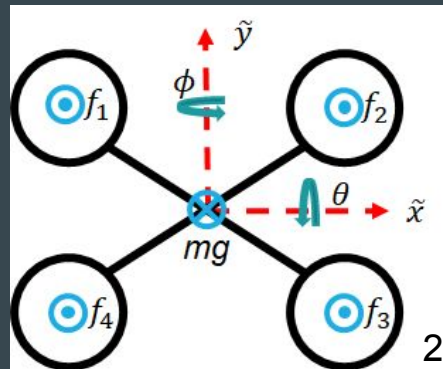
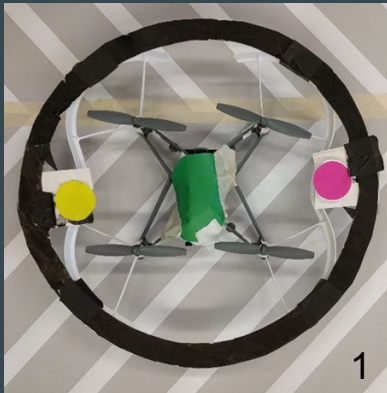
# Showcasing drone capabilities



# Distributed Robotics

How do we get a group of robots to accomplish a task together in a meaningful way?

1. Localization
2. Models and controllers
3. Coordination



# Stabilizing Robotics Programming Language (StarL)

- Platform to coordinate robots
  - Path finding
  - Collision avoidance
  - Waypoint searching
- Tracking system - Kinect
- Monitor - Android device
- Robot - Drones and Roombas



# FollowApp Demo

- Robots go to a set waypoint
- Coordinate and move to another
- Can set number and location of waypoints

```
public class FollowApp extends LogicThread {
    private static final String TAG = "Follow App";
    public static final int ARRIVED_MSG = 22;
    private int destIndex;
    private int messageCount = 0;
    private int numBots;
    private int numWaypoints;
    private boolean arrived = false;
    private boolean goForever = true;
    private int msgNum = 0;
    private HashSet<RobotMessage> receivedMsgs = new HashSet<>();

    final Map<String, ItemPosition> destinations = new HashMap<>();
    ItemPosition currentDestination;

    private enum Stage {
        INIT, PICK, GO, DONE, WAIT
    };

    private Stage stage = Stage.INIT;

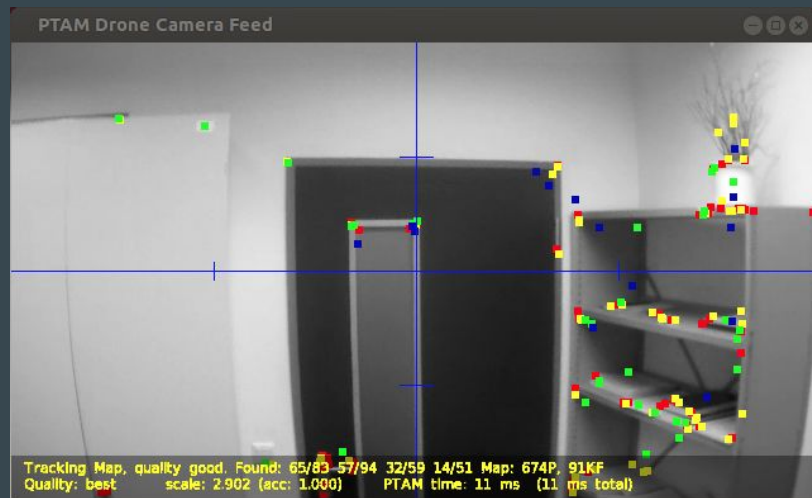
    public FollowApp(GlobalVarHolder gvh) {
        super(gvh);
        MotionParameters.Builder settings = new MotionParameters.Builder();
        settings.ROBOT_RADIUS(400);
        settings.COLAVOID_MODE(COLAVOID_MODE_TYPE.USE_COLAVOID);
        MotionParameters param = settings.build();
        gvh.plat.moat.setParameters(param);
        for(ItemPosition i : gvh.gps.getWaypointPositions())
            destinations.put(i.getName(), i);
        gvh.comms.addMsgListener(this, ARRIVED_MSG);
        // bot names must be bot0, bot1, ... botn for this to work
        String intValue = name.replaceAll("[^0-9]", "");
        destIndex = Integer.parseInt(intValue);
        numBots = gvh.id.getParticipants().size();
    }
}
```

# SLAM Demo (simultaneous localization and mapping)

- Positions itself based on its monocular camera
- Attempts to maintain the same position



Image: Parrot





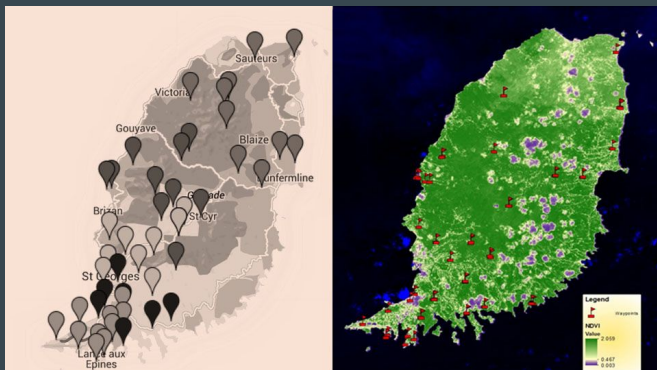
# Discussion



Distributed Robot Applications

# Microsoft Project Premonition

- Working to detect outbreaks before they happen
  - Capture mosquitoes to identify and analyze infectious diseases
- 3 step process:
  - Finding hotspots
  - Collecting mosquitoes
  - Detecting diseases



Images: Microsoft

# Amazon Kiva

- Warehouses can have thousands of these robots that pick up shelves and bring them to employees
  - From 1.5 hours to 15 minutes per order



# Earthquake Detection

- After the 2008 Sichuan earthquake, used to locate and assess the worst damages
  - Locate choke points
  - Help with surveying and reconstruction plans



Image: Chien-Min Chung



# Distributed Roombas

- House cleaning robot
- Can be programmed using Python with a Raspberry Pi

