A Low-Cost Approach to Autonomous Litter Collection

By James Heselden

Completed as part of a third year BSc in Computer Science

School of Computer Science

University of Lincoln

2019

# Introduction and Background:

## Introduction:

The aim of the project is to design and develop an autonomous system to deploy a robot to pick up litter. The project is focused around the requirements of the system being low-cost, low-maintenance, and high-efficacy. To ensure these requirements are met, special emphasis is placed on them during the decision-making processes throughout the development.

With the application of autonomy invading all areas, I say why not anti-littering

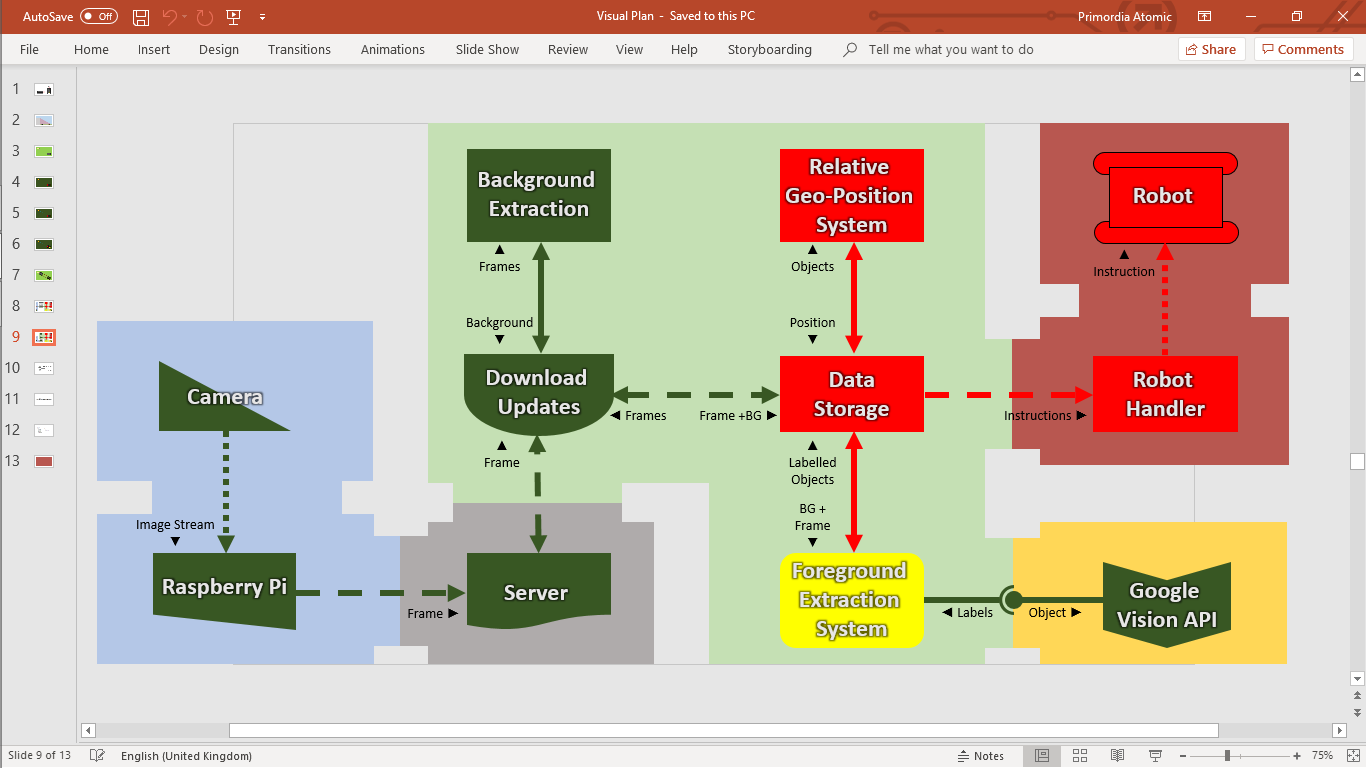
What are the biggest hurdles preventing autonomy in different areas?

* Cost? Effectiveness? Maintenance/Complexity? Social/Legal/Ethical?

This project plans to investigate, design and develop an autonomous system to deploy a robot to identify litter within an independent live video feed.

## Background:

The project is broken down into different stages, in this section, each stage will be analysed, and the development will continue from these decisions.



FOR EACH SECTION, LIST THE DECISIONS MADE FOR NOW

### Section1: Wireless Camera System

Was going to use Wired USB Camera but Changed to use (wireless)/Wifi Camera as wired didn’t allow for having multiple cameras over a large area. This was simpler and easier to implement for cost reduction.

Changed to use Wired USB Camera connected to RPI as it was easier then connecting cam to internet or setting up own campus-wide subnet or adding own routers for each wifi cam.

### Section2: Communications with Main Processing System

Was going to use FTP to connect directly with desktop… didn’t have admin perms to do so, so for the sake of the report a server is used as a middle ground where the cam publishes to, and the processing unit subscribes to.

### Section3: Background Extraction

Significant research concluded for this…

Various image stacking algorithms have been tested beginning with mean stacking, it was removed as a day-night cycle and weather shifts would drastically change the overall appearance of the images.

Entropy blurring was attempted but the implementation for such a system meant the quality of images were reduced making small objects harder to identify.

Stacking with edge detection was tested, however for it to work the quality of the camera needed to be improved, and the material patterns on the floor, greatly impacted the quality of the output.

Mode stacking on a relatively small timed cycle was found to be the most effective and allow the smallest impact of changes. The only issue currently remaining is random parts have high values when they should not… so more development needs to be put into making this work smarter. The maths behind why this is so effective at detecting change stems from the lack of maths. Mean is impacted by random sparks, and when there is a lot of change, it cannot identify the background. While mode is not impacted by random changes. It takes what is most static in the frame-set, which should always represent the background.

### Section4: Foreground Extraction

Simply, it is the new frame minus the background, where the value is 0 or near-0 then there is no change and should be disregarded.

### Section5: Object Identification

Object identification was originally planned to extract many features form the objects and apply SVM on them to classify against a dataset or labelled data. This was changed to use the Google Vision API, as the dataset of labelled data and its accuracy is much higher then what could be achieved in the time frame.

I may come back to it later on in the project and develop a custom model from the Google ML API.

### Section6: Litter Filter

Essentially, I need to list out all valid matches…. How do I choose what is defined as trash though?

### Section7: SLAM System

I kindda have an idea. We use the footage from the camera to find the relative position of the trash returned from the object identification and calculate how much the robot must rotate until its head and tail line up pointing at the trash. Then we move it forward until it hits the trash.

[Look up list of slam methods on Wikipedia]

### Section8: Robot Development

### Section9: Robot Movement Systems

### Section10: Robot Wireless Communications