Nicolas Grobelny

Optimal Camera Placement

CO3201 Interim Report

Contents

[Aims and Objectives 2](#_Toc87971049)

[Aims 2](#_Toc87971050)

# Aims and Objectives

## Aims

This project aims to determine the best solution/solutions to the Optimal Camera Placement Problem by conducting a technical review of the field, attempting to determine whether present algorithms are valid and can produce reproducible result.

Additionally, a web application with an accessible interface would be developed allowing users to find solutions using results of the technical review. The program should allow them to digitalise their surroundings, determine potential camera placements and get an optimal solution based on maximisation parameters, and constraints.

In this way, the research not only improves the field by comparing existing results but also allows users, such as security experts, to place their cameras with greater accuracy whilst using less time. So far, the most common method to place cameras involves previous knowledge and the “rule of thumb”, making it inefficient.

## Objectives

The main objectives required to achieve the project are:

1. Stage One – Problem Research:
   * Conduct research into the Optimal Camera Placement Problem, maximising number of potentially valid algorithms that could be used to solve that problem.
2. Stage Two – Implementation Research:
   * Determine the best Frontend technology for this application, focusing on accessibility and simplicity (HTML5/CSS4/JavaScript).
   * Determine the best Backend technology, focusing on speed and single page applications. (Python-Flask)
   * Determine the best connecting technology, focusing on simplicity and performance. (Web Sockets, more specifically Sockets.io)
3. Stage Three – Environment Implementation:
   * Implement a drawing environment on the client site using JavaScript. The environment should resemble a simple drawing application such as Paint.
   * Implement a single page Backend allowing user to fetch web page and conduct updates to it using Sockets.
   * Implement the socket connection so that user can exchange information with the server in an efficient and secure way.
4. Stage Four – Algorithm Implementation:
   1. Pick a random algorithm from the algorithms researched during stage one.
   2. Determine its validity based on claimed time complexity and difficulty to implement. If the algorithm does not meet criteria, disregard further steps.
   3. Implement and test the algorithm.
   4. Go to step “a” and continue till all algorithms have been considered.
5. Stage Five – Algorithm Comparison:
   1. Get sample data for the Optimal Camera Placement using sources such as GECCO OCP.
   2. Run each algorithm using sample data and collect results.
   3. Determine the best algorithm if possible. Otherwise identify the situation in which the algorithm has the best performance.

Each stage must be completed in the given order. Inside each stage, unordered lists can be finished in any order whereas subtasks from ordered sections need to be executed one after the other. This gives a clear structure and prevents from implementing features in a way that later forces refactoring.

# Planning and Timescales

## Methodology

As the tasks for this project are easily divisible into deliverable chunks, Scrum methodology allows to effectively present them to the client (Project Supervisor) without impacting quality.

## Scrum

Scrum is a project methodology based on philosophy of Agile. Scrum consists of short periods of time (Two weeks in this case) during which several user stories (features) is delivered. User stories are only delivered once they are considered done, maximising quality at the cost of time.

At the start of each sprint the client and I would decide the best feature/features that should be implemented. After the sprint finishes, the supervisor would be able to see results of that implementation.

In this case however, I would be using a variation of Scrum where each user story belongs to an overarching epic that corresponds to each project stage.