Nicolas Grobelny

Optimal Camera Placement

CO3201 Interim Report

Contents

[Aims and Objectives 2](#_Toc88056696)

[Aims 2](#_Toc88056697)

[Objectives 3](#_Toc88056698)

[Planning and Timescales 4](#_Toc88056699)

[Methodology 4](#_Toc88056700)

[Scrum 4](#_Toc88056701)

[Why Scrum 4](#_Toc88056702)

[Timescales 5](#_Toc88056703)

[Semester One 5](#_Toc88056704)

[Semester Two 6](#_Toc88056705)

# 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

### Scrum

Scrum is a project management framework based around Agile philosophy. It is based around a sprint – A short, time-boxed period during which a Scrum team attempts to complete a set amount of work. Sprints offer greater flexibility as that project requirements can be changed at the start of every sprint without much of an impact on project deadline.

Another advantage of Scrum is its deliverability – At the end of each sprint, the stakeholder would be able to see finished tasks and determine whether they are of acceptable quality.

### Why Scrum

The Optimal Camera Placement Application appears to be the first project of its type, it is near impossible to visualise all features that should be part of it. Currently, a list of required features tends to evolve naturally, be it from client’s feedback or from inaccessibility of the application itself. For example, an ability to display coordinates of the grid space using both grid space and real-world dimensions was not a planned feature but testing of the application revealed that it is near impossible to draw accurate buildings without it.

As such, I required a framework that can quickly react to changes and can be easily prototyped for client to see. Scrum performs well in this situation as its short sprint window (Two Weeks in this case) allows me to prototype requested features, show them to user and change them if necessary.

## Timescales

### Semester One

#### Done

Stage One – Problem Research **(11th October – 20th October):**

* Conduct research into the Optimal Camera Placement Problem, maximising number of potentially valid algorithms that could be used to solve that problem.

Stage Two – Implementation Research **(11th October – 20th October):**

* 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)

#### Started

Stage Three – Environment Implementation **(20th October – 1st December):**

* 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. **(Done)**
* Implement the socket connection so that user can exchange information with the server in an efficient and secure way. **(Done)**

Stage Four – Algorithm Implementation **(3rd November – 21st February)**:

Each individual algorithm should take approximately a week to implement, with enough time left for adjustments.

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. **(Done 1 out of 10)**

### Semester Two

#### Not Started

Stage Five – Algorithm Comparison (21st February – 7th April):

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.

Stage Six – Testing, Improvements and User Suggestions (7th April – 5th May)

* Ensure that test coverage is at least at 75 %, excluding network functions.
* Implement additional stakeholder suggestions (Such as optional feature or quality of life improvements).
* Add accessibility features and ensure compliance with web standard.

### Justification

Whilst adding exact dates might allow to better track whether the project is progressing at a right rate, the nature of Scrum sprint prevents me from specifying delivery dates within each epic.

Doing so would compromise quality as features will be rushed to meet the expected date. Another option is to ignore exact dates until the task is done, removing the reason for those times to be there in the first place.

Each epic has been chosen in a way to overestimate amount of time needed to implement it. At the current moment the project is progressing faster than expected, allowing to assign a significantly larger duration to Algorithm Implementation and Comparison. In case that time proves to be insufficient to deliver the project, the Stage Six is optional and can be cut out permanently to ensure quality of the deliverable.

## Description of the Prototype

The Optimal Camera Placement Application (OCPA) is implemented as a web application.

### Technologies Used

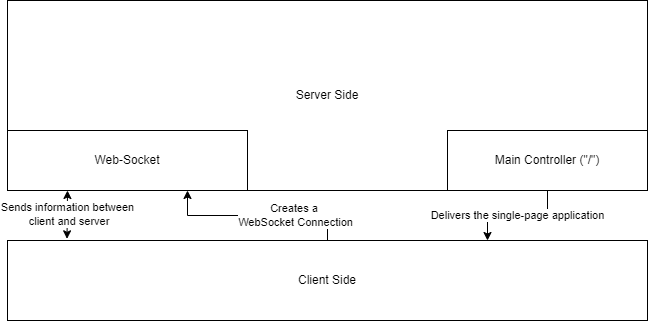
#### Backend

The backend of the application (Otherwise known as server side) is based around Python web framework known as Flask.

Flask applications have an advantage of being fast to write and even easier to deploy. This is helpful when considering rapid prototyping required by Scrum methodology.

Another advantage of Flask is its optimisation (Faster than other Python frameworks such as Django), especially for single page applications.

Together with Web Sockets, this allows server to communicate with client whilst using minimal number of resources and internet bandwidth. At the current point, the application sends non-empty board information to the server and receives optimal camera placements.

Optimisation needs to be conducted at every step as Camera Placement problem cannot be accurately solved in a Polynomial time. The time complexity of considered algorithms can vary from n\*s\*c to c\*(n^c) (Where n is a number of possible camera placements, c is a number of cameras and s is the number of sample points). This ensures that the server has enough resources to conduct these resource intensive operations without impacting the user.

As can be seen in the image above, the actual application has been simplified as much as possible. The application’s connection between client and the server consists of two main controllers. Firstly, there is a Main Controller that is responsible with delivering all of the static data and HTML once user requests the website.

The other controller is based around Python implementation of the Web Socket library known as Socket.io.

Other possibilities, such as MVC or AJAX have been considered and ultimately rejected for reasons below:

MVC – Model-View-Controller requires a large number of conversions and data transfer, especially when transferring model data to and from client. Whilst implementing this framework type would drastically simplify client-side code, the performance drawback is too great for this project.

AJAX – Asynchronous JavaScript and XML works similarly to the Web Socket technology in a way that a user can send and receive data after the web page has loaded. The main advantage of AJAX is security that can be provided at both client side and server side. Unfortunately, the security is not a concern in this application as no private data is transferred between client and the server. As such, Web Socket is a better alternative to AJAX.