[Document title]

[Company name] | [Company address]

[Document subtitle]

Mikołaj Grobelny

[Year]

Contents

# Abstract

# Introduction

## What is an Optimal Camera Placement?

FIGURE showing an example of camera placement problem.

Optimal Camera Placement is a type of NP-Complete problem where the objective is to place cameras in a way that maximises the coverage of cameras in a system whilst minimising cost. An example of such system can be seen in figure 1.1-1.

The need to solve this issue comes from the fact that security companies are struggling to create optimal camera systems. Their current approach is based on trial-and-error and experience of their employees **Invalid source specified.**. This results in a number of cameras that is not ideal, leading to either blind spots or a high maintenance cost.

## Aims and Objectives

The **main aim** of this project is to develop a web application allowing for security employees to receive a viable camera placement based on provided data. Viable solution means achieving performance same as or higher than a human camera placer whilst taking less than a minute to execute (Given that area is not large).

The **secondary aim** is to conduct a survey of algorithms that can be used to solve the camera placement problem and compare them against each other. This allows other researchers to continue the work done in this paper, leading to better algorithms that will be improve user’s experience.

Following objectives are required to achieve these aims:

1. Background
   1. Research currently available Optimal Camera Placement algorithms.
   2. Filter out algorithms that were not designed to operate in real-life situations.
   3. Determine difficulty of implementation for algorithms and rank them from simplest to hardest.
2. Requirements
   1. Research how security companies place cameras currently.
   2. Write user stories based on the research
   3. Translate user stories into Functional Requirements
3. Design and Specification
   1. Research technologies best suited to meet requirements
   2. Design an Application Architecture Diagram
   3. Make a Low-Fidelity Prototype of Application’s interface
4. Backend Implementation
   1. Implement the home page
   2. Implement connection between Backend and Frontend
   3. Implement backend environment and helper functions (Such as visibility checker)
5. Frontend Implementation
   1. Implement the User Interface designed in part 3c.
   2. Implement the Functionality required by part 2c.
   3. Allow Frontend to send information to the Backend
6. Algorithm Implementation
   1. Implement as many algorithms from list 1c as possible in the time frame
   2. Conduct an analysis of implementations, comparing vs number of cameras vs performance
   3. Based on previous results, determine which algorithm has the highest average performance.

# Background

## Related Work

The first step on creating an Optimal Camera Placement application was evaluation of systems with similar functionalities.

Whilst there are application allowing to user to manually place cameras and examine coverage, such as one present in <https://ipvm.com/reports/camera-design-tool-with-google-maps-integration>, there is currently no publicly available programs that allow user to automatically place cameras based on user requirements.

As such, this project be inspired by features manual placement applications and where necessary use suggestions from research papers focused on real-life scenarios.