
Tartu University

Faculty of Science and Technology

Institute of Technology

Naghma Afreen

# Optimal agent positioning for dynamic event monitoring and analysis.

Master's thesis (30 EAP)
Robotics and Computer Engineering

Supervisor:

Aditya Savio Paul,

Tartu Observatory

University of Tartu

Estonia

Tartu 2024

# Resümee/Abstract

**Optimaalne agendi positsioneerimine sündmuste dünaamilise jälgimise ja analüüsi jaoks.**

Paljud looduses ja kosmoses esinevad nähtused on oma olemuselt dünaamilised, ettearvamatud ja võivad muutuda hetkega, mistõttu on neid väga keeruline jälgida ja iseloomustada. Näiteks kui selliste sündmuste jälgimiseks kasutada tavapärast fotograafiat, mis on reeglina mõeldud praktiliselt staatilise olukorra jälgimiseks, on tulemuseks madalam täpsus ja andmekadu. Dünaamiliselt kiiresti muutuvate nähtuste või sündmuste vaatlemine vajab seega kogu vaatlussüsteemi optimeerimist. Selles töös pakume välja meetodi kaamerasüsteemide optimeerimiseks dünaamiliste sündmuste jälgimise jaoks. See sisaldab vaatlusaluses ruumis kaamerate erineval viisil ümberpaigutamist ja nende abil sündmuse jäädvustamist, mille põhjal leitakse vaatlusaluse olukorra lahend, millele saab seejärel anda numbrilise hinnangu hindamisfunktsiooni abil. See võimaldab erinevaid konfiguratsioone läbi proovida, et leida optimaalne konfiguratsioon selle sündmuse jälgimiseks.

**CERCS:** T120 Süsteemitehnika, arvutitehnoloogia; T125 Automatiseerimine, robootika, juhtimistehnika, T320 Kosmosetehnoloogia [10])

**Märksõnad:** kaamerasüsteemid, simulatsioon, sündmuste tuvastamine, monitooring, andurite optimeerimine, arvutinägemine, arvutigraafika, kosmosetehnoloogia, arvutitehnika.

**Optimal agent positioning for dynamic event monitoring and analysis.**

Events like natural hazards and celestial occurrences are dynamic in nature, they are unpredictable and change instantaneously. In order to study events better, it is essential to develop methods for better understanding. However, observing such events is challenging due to their changing nature compared to observation of events that do not change with time. Moreover, traditional camera systems developed for observing static events tend to generate results with less accuracy and information loss. Dynamic event observation requires dynamic setup of camera systems to improve the study of such phenomena. We propose a method for optimizing camera configurations for observing dynamic events. The observation space is sampled using a volumetric sampler. The cameras are iterated over these samples and the event is captured; an optimal solution then evaluated using a scoring criteria. The scoring function assists in analyzing the optimal configuration of camera setup for observing the desired event. The research aims to develop and improve observational techniques for dynamic events.

**CERCS:** T120 Systems engineering, computer technology; T125 Automation, robotics, control engineering, T320 Space technology [10])

**Keywords:** camera systems, simulation, event detection, monitoring, sensor optimization, computer vision, computer graphic, space technology, computer engineering.

# Contents

# List of Figures

# Abbreviations. Constants. Generic Terms

**RRT** - Rapidly Exploring Random Tree

**UAV** - Unmanned Aerial Vehicle

**CPP** - Coverage Path Planning

**TSP** - Travelling Salesman Problem

**DRL** - Deep Reinforcement Learning

**LPC** - Long Period Comet

**KBO** - Kuiper Belt Object

**DART** - Double Asteroid Redirection Test

**HST** - Hubble Space Telescope

**OCP** - Optimal Camera Placement Problem

**AGP** - Art Gallery Problem

**SCP** - Set Cover Problem

**FoV** - Field-of-View

**KDE** - Kernel Density Estimation

## nomenclature

$\epsilon$ - event definition

$C$ - Coverage Function

$\delta$ - Score function

$d$ - Euclidean distance

$\rho$ - Valid illuminated pixels

$V$ - Volumetric extent

$df$ - Dataframe

$W$ - Width of the target object

$H$ - Height of the target object

$\omega$ - Camera angle

$fov$ - Field-of-View

# 1 Introduction

From outer space to our confined homes there are numerous dynamic events that scientists and engineers are interested in observing and analyzing.While studying or observing these events are highly significant, they could be equally challenging. These setbacks are prominent in domains like security, surveillance, scientific research and disaster management. For instance, observing celestial phenomena like comet flybys and asteroid collisions equip us with important knowledge about their trajectories and composition. Monitoring geological events helps in managing risks posed by natural hazards and surveillance systems with good coverage possess the ability to regulate the surrounding. Insufficient coverage leads to limited information, which could be hazardous in numerous settings. Using standard techniques which mostly rely on static camera configurations is inefficient while working with dynamic events. Such events require us to be cautious of rapid changes and uncertainty. Thus, there is a need for exploring approaches to optimize camera configurations for dynamic event observation. The camera system which is capable of adapting to the changes would be instrumental in assisting us in investigating rapidly evolving scenarios or phenomena.

Developments in the fields of computer vision and simulation modeling have generated new possibilities to advance and refine dynamic event observation technology. These technologies facilitate strategies to establish and implement camera systems capable of optimizing coverage while monitoring crucial information and adapt their configuration dynamically with changing conditions.

## 1.1 Motivation

The objective of the research is to develop an approach for configuring camera systems [1] which aids in finding optimal solutions for coverage of potential events and targets. The motivation behind the project originates from the importance of observing dynamic events occurring in the real world. These events, defined by uncertainty, pose a challenge for observers and the research provides an approach to deal with such complexities using volumetric sampling of chaotic systems, computer vision and simulation modeling.

[1] A camera system in this thesis refers to set of cameras that are used for simultaneously observing an event or target.

## 1.2 Objectives and Roadmap

The optimal positioning of the agents, which in our research are cameras, for dynamic observation is realized by exploring sampling techniques and experimentation in simulation environments and validation through analogue environments. The work is directed by following milestones:

- Generating samples of cameras configuration by volumetric sampling technique
- Environment modeling and simulation framework development
- Defining scoring function based on quality of observation from different camera systems.
- Validating the optimal configurations through experimentation in an analogue environment.

# 2 Background and Related Study

## 2.1 Sampling

The concept of exploring samples in a state-space has been prevalent in the field of robotics for quite a long time. The method for assessing volumetric information gain has been explored by Delmerico et al [14], where the volumetric information gain metrics aided in optimizing the reconstruction of 3D models by finding the optimal positions for sensor placement. The study quantified the volumetric information contained in voxels of a probabilistic volumetric map and then compared them through simulated experiments. Newell [16] introduce a method in digital image synthesis using procedural models, where algorithmic methods have been used to create complex images. The framework generates highly detailed objects and environment digitally. Curless and Levoy [11], develop a volumetric method for integrating range images, that posses properties like incremental updating, representation of directional uncertainty, ability to fill gaps in the reconstruction, and robustness in the presence of outliers. The volumetric representation in the proposed method consists of collective weighted signed distance function, where every range image is worked with at a time, by converting each to a distance function, then combining with the data acquired through additive scheme. The final manifold is generated by extracting an isosurface from the volumetric grid. The technique aids in development of comprehensive 3D models through various range images.

Bircher et al [4], present a path planning algorithm for exploration and surface inspection, using receding horizon. They update the planned path continuously using the sensor data which optimizes the trajectory in real-time. Naderi et al [36], introduce real-time path planning algorithm (RT-RTT*) which enhances the Rapidly Exploring Random Tree (RRT*) algorithm by including real-time constraints. They introduce a strategy for rewiring the tree online, which made it possible for the agent to move with the tree root without changing the previously sampled paths. Otte and Frazzoli [40], propose a sampling-based replanning algorithm that is asymptotically optimal and single-query. They enhanced the RRT for efficient path adjustments in response to changes in environment. The algorithm adjust and updates the search-graph throughout the navigation and whenever there is change in the environment, the existing search-graph is remodeled and the shortest-path-to-goal sub-tree is updated. Paul and Otte [42], introduce a receding horizon approach for orbital maneuvering for missions near small solar system bodies using gravity model refinement. The method Combines the adaptive navigation with continuous gravitational model updates, which helps in mission planning for space exploration by enhancing the accuracy of navigation.

The samples are generally explored in 2 dimensions to provide a cartesian waypoint

for ground robots. However, with the inception of drones, the exploration has also extended to 3 dimensions. Within volumetric sampling, continuous samples within the 3D space of defined boundaries are explored, to produce discrete solutions. Wardhana et al [50] explore the possibility of volumetric path planning in virtual world where they subdivide the virtual world and voxelize the region. The waypoints are then sampled at the corner of those regions. Almadhoun et al [2] introduce a coverage path planning algorithm to construct accurate 3D models of complex large structures, using adaptive viewpoint sampling. They sample the waypoint using the coverage path planners for the UAV to navigate through the observation space.

This work leverages the available computational capabilities and samples the coordinates within the entire bounded region. This is to respect the event sequence and plan for all possible unbiased locations, given the requirement that the probability of event occurrence can be high at any given instance within the bounded regions.

## 2.2 Scene Coverage

Scene coverage within robotic systems has been widely explored for the requirements of agent perception and cognition of the environment, in which they should act. This is accomplished over a variety of sensor systems that provide an environmental understanding of the space within which the agent action takes place. The information regarding scene coverage is accumulated from sensors that are either modeled within simulations or real-time perception from physical sensors. Jiang [27] uses a distributed optimization of visual sensors to cover a large 3D scene using visible triangle-based optimization. They strategize on coverage performance by compensating for the lost in rough space partition, resulting in an optimal deployment configuration of the camera network. Matignon[34] provides a comparison of different strategies for coverage and mapping of complex scenes, assuming that the robot's motion in the scene is perfect. For occlusions, Border at al.[7] present solutions for handling occlusions within unstructured environments using ray-casting to evaluate the amount of information available.

Dang et al [13], propose a strategy to autonomous exploration using aerial robots. The aerial robot explore environment, detect objects belonging to classes of interest and relating these detection to the location on the map. An enhanced exploration path-planning algorithm is designed to define a collision free path by maximizing a gain related to exploring the environment. Kyaw et al [32], present an approach for solving the coverage path planning (CPP) problem using Travelling Salesman Problem (TSP) and Deep Reinforcement Learning (DRL) by using the grid-based maps. The algorithm uses cellular decomposition to decompose the environment to generate the coverage path by solving each cell as TSP. The proposed method is demonstrated through simulation experiments showing complete coverage while efficiently generating coverage paths. Ramachandran et al [44], introduce a distributed method for generating an occupancy grid map for an unknown environment using a swarm of robots, using global localization capabilities. They used Levy walks by maximizing mutual information between robot's occupancy grid map and distance measurement. Each robot generates its occupancy grid map by continuously combining the distance measurements between itself and the neighboring robots. Jonschkowski and Brock [29], integrates the

robotic algorithms with machine learning techniques to learn task-specific information. The framework combines these techniques to generate end-to-end learnable histogram filter, which helps in developing flexible and adaptive system for robots which are capable of learning through experience.

## 2.3 Target bodies in Space

### 2.3.1 Comets

Comets, are mainly composed of ice, rocks and dust, which heat up while approaching the sun and release gas and dust forming a glowing coma and a tail. Researches have shown that the comet structure consists of a soft interiors and a hard crusts.NASA's Deep Impact [37] and the European Space Agency's Rosetta [21] mission have provided insights into the structure of the comets, which supports the concept[1]. There are several dynamic activities that occur around comets. As evident from the studies these activities, are usually driven by the interaction between their components and the solar wind. The nucleus of the comet consists of frozen molecules, which includes $H_2O$, CO, $CO_2$ , $CH_3OH$, $CH_4$ , $H_2S$ and $NH_3$ [6].
When comets approach the sun, due to sublimation of the ice molecules there is an increase in gas pressure. This pressure increase leads to ejection of gas and dust from the comet's surface, which can be observed as comet tails or jets [23]. The dust particles from the surface are believed to be the remnants of the early solar system, hence can provide crucial information regarding it's evolution and formation.

Comet Interceptor [19], an ESA upcoming mission which aims to explore long period comets (LPCs) (comets with orbital period greater than 200 years), will assist in furthering our knowledge of cometary science. The main goal of the mission involves to observing and analyzing characteristics of the target object through close approach (see Figure 2.1). The close observation of the comet would assist in investigating its surface composition, shape, morphology and structure. Characterizing the nucleus of the comet provides crucial information regarding the evolution of the surface with time. The mission will provide insightful characteristics of LPCs which can be compared with other celestial bodies [28].

There are many comets which are explored by various space missions, in Figure 2.2, the image demonstrates the significance of studying the comets upclose to compare their features. The image of Arrokoth, a Kuiper Belt Object (KBO) displays the smooth surface with uniform texture which aligns with the idea that KBOs are similar to the primordial bodies that have experienced least amount of surface evolution since their formation[28].

Missions with in situ observation capabilities,support comprehensive study of the surface of the celestial bodies. Figure 2.3. Illustrates observed features on comet 67P/ Churyumov-Gerasimenko. The pits on the surface of the comet show regions affected by erosion or sublimation; they act as evidence for active surface processes due to sublimation of unstable materials in the vicinity of the sun. Images like these offer insight into surface features on a comet.[28].

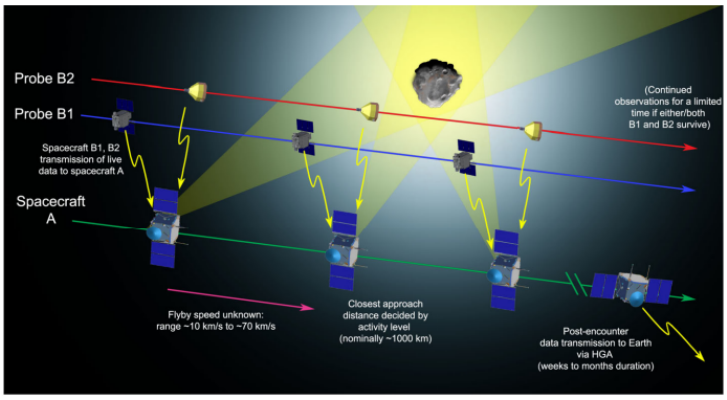



Figure 2.1: Sketch of the Comet Interceptor flyby, not to scale.[28]

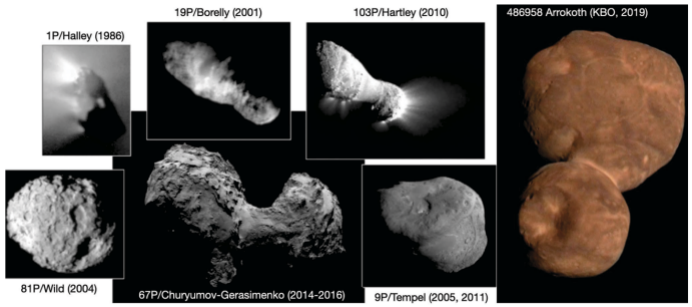



Figure 2.2: A subset of the cometary nuclei that have been visited by spacecraft and on the right an image of Arrokoth, a Kuiper Belt Object[28]. Not shown to scale

### 2.3.2 Asteroids

Asteroids [18] are rocky, dry celestial objects orbiting the sun. Study of their surface is crucial for understanding the early stages of planet formation and evolution. Asteroids have a dynamic dust environment around their surface as well as on their surfaces. These dynamic activities which leads to erosion, are usually the result of external influence like micro meteoroid impacts or solar wind and radiation[52].

NASA's Double Asteroid Redirection Test (DART) mission [45] launched in 2021, aimed at studying how kinetic impacts changes the orbit of Dimorphos around its binary component, Didymos [45]. The intentional impact of the spacecraft with the asteroid, resulted in plume formation. This impact debris or ejecta evolved into a tail. The observation of the ejecta provides information regarding the surface components of the Dimorphos. The impact of the DART spacecraft was observed by the LICIACube CUbeSat, Hubble Space Telescope (HST) and ground-based telescopes.

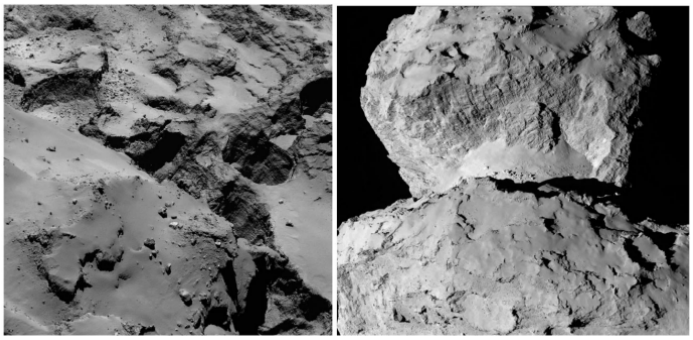


Figure 2.3: Pits on the surface of comet 67P/Churyumov-Gerasimenko (left). Hathor cliff above the 'neck' region between the two lobes of 67P (right)[20], [28].

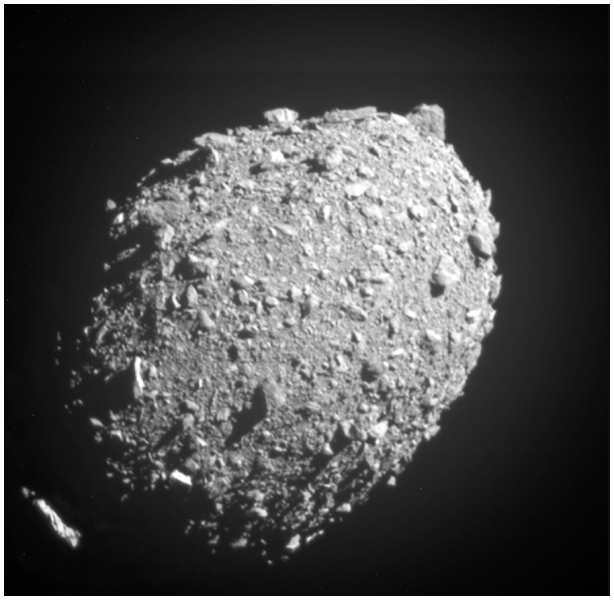


Figure 2.4: Asteroid moonlet Dimorphos as seen by the DART spacecraft 11 seconds before impact [45]

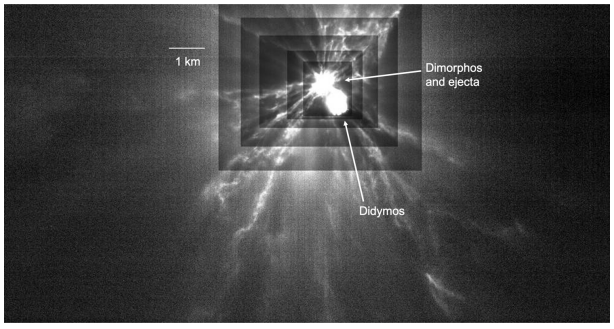



Figure 2.5: Ejecta observed by LICIACube after the impact on Dimorphos. There is a differential stretch in the contrast to make ejecta to be at same brightness, it is increased with each larger reference box.[12]

## 2.4 Camera system Optimization

The optimal camera placement problem (OCP) originates from computational geometry, distinctively in the Art Gallery Problem (AGP) introduced by O'Rourke [38]. The Art Gallery Problem main goal is to determine the least amount of observing equipment needed to observe every point on a polygon's boundary. The experiments around the optimal camera placement utilizes computational geometry, computer vision and image processing. By using these tools scientists aim to formulate approaches to configure camera systems that optimize the result bound under given constraints.

The area being surveyed by OCP can be modeled by two approaches: discrete or continuous [31]. The discrete method works by sampling the area while the continuous method examines the area as plane or volume. Various sampling methods have been explored in discrete approaches, like early works of Erdem and Sclaroff [17] and Horster and Lienhart[24] formulating the OCP using binary linear programs, enforcing the discretization of every parameter. The frequency of the sampling is set and implemented on spatial dimensions, this generates points the technique can cover. Another approach introduced by Murray et al.[35], for sampling using square rectangular grids where each grid cell shows the level of requirement for coverage from at least one camera.

**Set Cover problem (SCP)**

The structure of OCP is similar to the Set Cover problem (SCP), one of Karp's 21 NP-complete problem [30]. The OCP, where the focus is on analyzing visibility and optimizing camera configurations can be related to the SCP where the main goal is to evaluate the smallest subset of sets that covers a given set of elements. By allocating each configuration of camera, a cost, the OCP can also be formulated as a minimization problem similar to SCP [31].

**SCP formulation :**

Selected sets cost minimization,

$$Min \sum_{j=1}^{|J|} c_j x_j \tag{2.1}$$

Ensuring that each element is covered,

$$\sum_{j=1}^{|J|} a_{ij} x_j \geq 1 \forall i \in [1, ..., .|I|] \tag{2.2}$$

Where,
$I$: Set of all coverable points in the surveillance area.
$J$: Set of cameras, expressed as subsets of $I$.
$c_j$: Cost associated with camera configuration $j$.
$x_j$:Binary decision variable indicating if camera $j$ is selected (1) or not (0).
$a_{ij}$: Binary element indicating whether camera $j$ covers point $i$.

The SCP formulation [31] defines a target function and constraints to ensure optimization of solution to the problem. The target function aims to minimize the cost of camera configuration selection. To ensure that each point is covered by at least one selected camera configuration, constraints are implemented. This formulation illustrates the mathematical definition of a problem whose optimal solution can be found using optimization techniques.

## 2.5 Computer Vision techniques

**Image segmentation**

Image segmentation [47] is a technique which assists in extracting attributes or partitioning an image into various parts or regions. Segmentation algorithms mostly depend on two essential intensity value properties: similarity and discontinuity [43]. Discontinuity based approach uses sudden intensity changes for segmenting images, like edges. The similarity based method separates images according to the areas with similar attributes. Some of the examples of the similarity based techniques includethresholding, region splitting and merging. The combination of these two methods or techniques improves the segmentation accuracy.

$R$ represents the whole spatial region utilized by an image. The image segmentation is a method [43] to divide $R$ into $n$ segments or subregions, $R_1, R_2, R_3, \ldots, R_n$, such that

***A.*** $\cup_{i=1}^{n} R_i$ = R.
***B.*** $R_i$ is a connected set, $i$ = *1,2,..,n*.
***C.*** $R_i \bigcap$R$_j$= $\varnothing$ for all $i$ and j,i $\neq$ j.
***D.*** $Q(R_i)$= *TRUE* for $i$ = 1,2,..,n.
***E.*** $Q(R_i \cup R_j)$= FALSE for any adjacent regions $R_i$ and $R_j$.

Where, $Q(R_k)$ is a logical predicate which is defined over the points within the set $R_k$, and $\varnothing$ is a null set. Two regions $R_i$ and $R_j$ are adjacent if their union is a connected set. The predicate is a condition which determines the difference in adjacent regions. When implemented on regions $R_i$ and $R_j$ , the logical predicate analyzes to either TRUE or FALSE. If TRUE, it indicates that the regions $R_i$ and $R_j$ are different , which means there is variation between them. If FALSE, it implies that these regions are similar to each other [43].

***Condition A.*** suggests that every pixel in the image must be allocated to a region and confirms that no pixel is uncategorized, which leads to thorough segmentation of the complete image.

***Condition B.*** Instructs that points in a region must be connected in some predetermined sense.

***Condition C.*** Demands the segmented regions to disjoint, which means that they do not overlap or share common points.

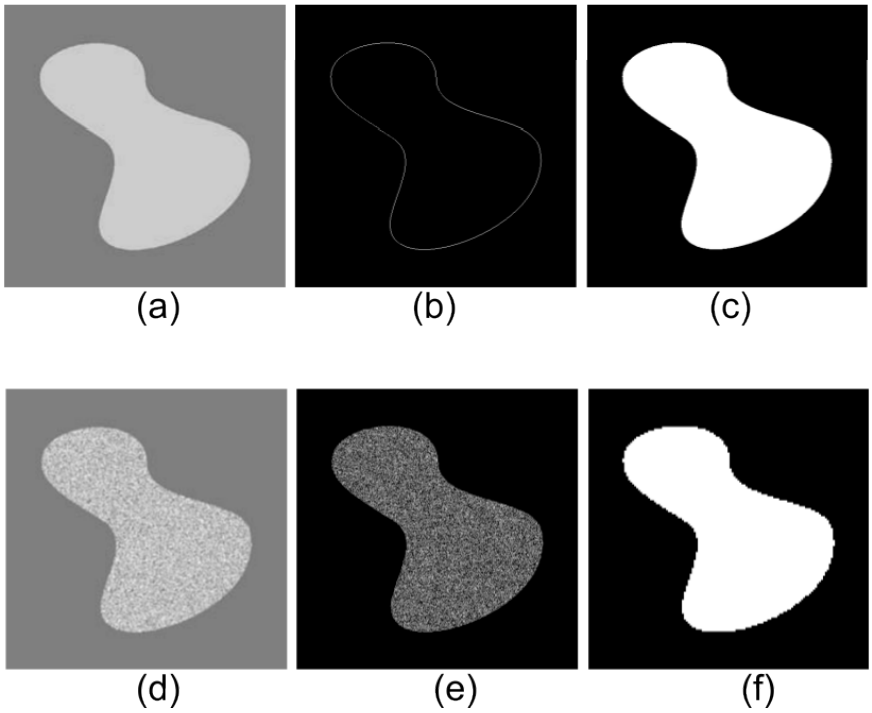



Figure 2.6: Image Segmentation [43]

***Condition D.*** deals with the properties that must be fulfilled by pixels in each segmented region, that is ensuring that every pixel belonging to a region has similar characteristics.

***Condition E.*** indicates that the adjacent regions must be different according to the predicate logic.

In Figure 2.6 (a), Image with a region of constant intensity is shown, which is overlapped on a dark background. (b) inner region boundary is calculated using the intensity discontinuities, the points inside and outside remains same as the intensity inside and outside is same. (c) to segment the image the pixels outside and inside of the boundary are allocated different levels, black and white respectively. (d) Inner region of the image is textured, which makes segmentation by edge inefficient. As the outer region remains constant, segmentation based on region should be a productive method for segmentation of such images. (e) Segmentation is attained by splitting the given image into sections or subregions and labeling each of these subregions according to the pixel values standard deviation. (f) The subregions which have positive standard deviation are labeled white while the ones with zero are black labeled [43].

**Thresholding**

Splitting of an image into subregions according to the values of the pixel, assists in segmenting and extracting the desired object or region from the whole image. One of the basic methods of thresholding is the intensity thresholding, where the separation is done based on the intensity values of the pixels [43]. The accuracy of such a method

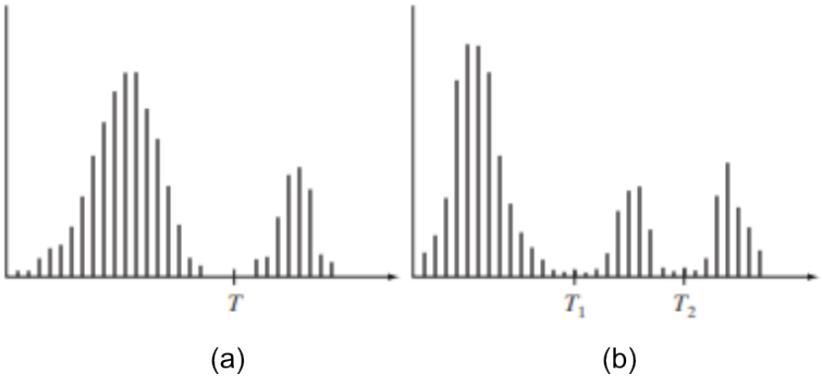



Figure 2.7: Intensity histograms that can be partitioned (a) by a single threshold, $T_1$, and (b) by dual thresholds,$T_1$and$T_1$.[43]

relies on various parameters such as the peaks separation in the intensity histogram, noise in the image and the size of the object and the background, and the illumination properties [26].

For extracting or separating the object from the background the basic method [43] is to define a threshold $T$ that can efficiently differentiate between the pixel values related to both. Which implies that any point *(x,y)* in the image which has intensity greater than the threshold $T$ is the object, otherwise it is labeled as background.

So, a segmented image *g(x,y)* is defined as:

$$g(x, y) = 1 : f(x, y) > T, 0 : f(x, y) \leq T \tag{2.3}$$

Where , $f(x, y)$ is an image with light objects on a dark background.

# 3 Problem Statement(s)

The problem addressed in this thesis is to assimilate an optimal set of agent configurations to observe an event within a reasonable spatial resolution. The spatial resolution relates to the event being observed as a function that attains a higher score based on the amount of information captured by the agents. To achieve this, subset of the problem statement are defined, which are formally defined below. For a set of observing agents $A = [a_1, a_2, a_3. . . .a_n]$, find the camera positions that can observe the event with a spatial resolution higher than $S$.

**Problem statement 01 (PS01) – Observational Criteria**
For a sequence of event occurrences $\epsilon_s$ within event $\epsilon$, observe an instance of the event $\epsilon_i$ , where ( $[\epsilon_i \subset \epsilon_s] \subset \epsilon$) for which the likelihood of event observations $P$ is maximum. This is modelled as a function where the spatial parameters provide the least challenging conditions for spatial observations.

**Problem statement 02 (PS02) – Spatial agent distribution**
Given the event instance, spatially distribute the observing agents, within the volumetric space at a distance that captures the event within the observational extent.

**Problem statement 03 (PS03) – Optimal agent positioning**
Iterate the agent distribution with the observational extent to capture the event with a relatively higher score function. For the scope of the thesis, the agents are modelled as optical sensors [] that are inherently capable to observe the event, given the conditions for the identified set of functions are met. The observational extent is a combination of the distance and the field of view of the sensors. The likelihood of event observations (P) is highest at $\epsilon_i$. A set of sensor configurations comprises of 3 sensors simultaneously observing the event. The spatial parameter is the amount of ambient luminous flux for which the event is conveniently visible. One lumen (lm) is equal to the luminous flux emitted within unit solid angle (one steradian,sr) for a luminous intensity of one candela (cd). [15]

$$1lm = 1cd \times 1sr \tag{3.1}$$

The various functions considered to solve the problem statement are defined below.

**Coverage Function**
The coverage function calculates the various coverage of the event instance and maximizes to utilize the entire volumetric space. [33] [reference the camera paper here]

$$C_n = \sum_{n=1}^{N} Max \iiint C_{n[xyz]} \otimes C_{n-1} \tag{3.2}$$

**Score Function**
The score function $\delta$ is maximized as the density of the valid illuminated (with masking and without noise) pixels on the sensor observing the event with a given FoV ($fov$) and Distance ($d$) from the event.

$$fov = 2 * \arctan(sensorsize/2 * focallength) \tag{3.3}$$

$$d = EucledianDistance(EventLocation, AgentLocation) \tag{3.4}$$

$$\rho = No.of illuminated pixels/Areaofsensor \tag{3.5}$$

$$\delta = \sum_{V_{obs}} Max(fov, \rho, d) \tag{3.6}$$

**Limit Function**
Volumetric Boundary values define the cartesian boundary extents within which the observational location lie.

$$V_{xyz} < V_{obs} < V_{XYZ} \tag{3.7}$$

The main goal of this project is Developing an approach and modeling framework for optimal camera configuration for dynamic event monitoring. The following objectives are specifically targeted by the research:

1. **Developing sampling technique**
   The project involved working with different sampling techniques and finding the most suitable technique for our goal.

2. **Generating camera configuration**
   Using sampling strategies to generate a variety of camera setups. Through methodical sampling from many characteristics, including camera placements, orientations, and Field-of-View (FoV) angles, we generated an extensive array of camera configurations for assessment.

3. **Developing simulation environment**
   In order to model dynamic events and camera setups in a virtual environment, developing a flexible simulation framework using Blender software. For testing and assessing various camera setups under various event situations, this framework offered a practical and adaptable platform.

4. **Data gathering**
   The data for our project includes rendered images from the simulation environment and the captured images from the physical environment. Multiple campaigns or experiments were conducted for gathering sufficient quantity of data for validation .

5. **Target detection**
   For extracting target object from the gathered image data, various computer vision and image processing techniques were performed on both physical and simulation environment images. Analysis of intensity of the target object with respect to its environment was also performed, along with computing coverage area of the target object from FoV of each camera in the setup.


Figure 3.1: Binary Image


Figure 3.2: Binary Masking

6. **Score calculation**
   On the basis of the distance between the cameras and the target, the FoV of cameras, and the overlapping area between three camera groups the Scoring function was defined. The function provided analysis of how different camera configurations observe the target. During evaluation the function used cameras in a group of three, and each of these combination scores was calculated based on coverage and the FoV. The scores were calculated to find the camera combinations that produced the best observation results, which helps in selecting the most effective camera setup for dynamic event monitoring.

7. **Evaluation and validation**
   Calculating the error range and comparing between the coverage and intensity of the target within the images from the simulation and physical environments,

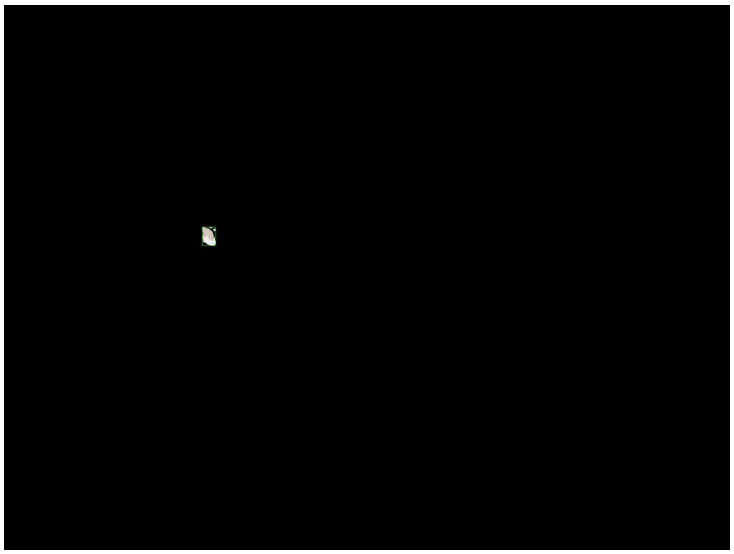


Figure 3.3: Masked out target

Using the result from the physical environment images to evaluate the results from the simulation environment and validate our approach.

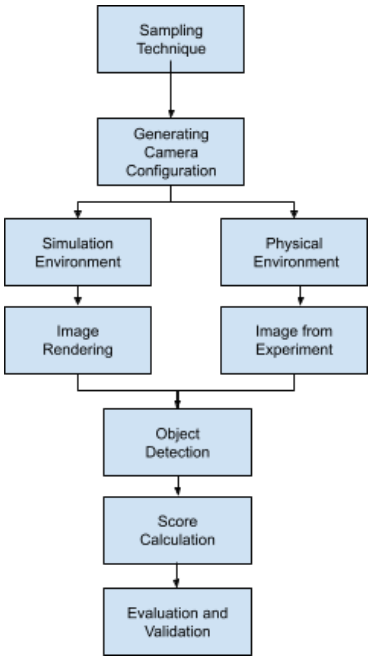



Figure 3.4: Work flow

# 4 State of the Art

Strategies for optimal placement of agents for event observation has taken leaps with advancement in computer vision and image processing. Agent or camera optimal positioning techniques for dynamic event monitoring are developed on methods like Set Cover Problem and Greedy algorithms [24]. By leveraging computational geometry and optical sciences, an approach for generating optimal camera configuration on the basis of quality of the captured images can be established . The quality of the resulting images also depends on the agent/camera optical properties like the focal length, camera angles and the sensor size and type. These optical properties assist in analyzing the FoV of the cameras, which provides the information about the coverage area the cameras can possibly acquire during observations.

## 4.1 Software Resources

Resources for 3D modeling, like Blender [5], facilitate development of simulation framework and environment modeling, which contributes towards event setup. Blender has been an exceptional 3D Modeling software, it offers various tools for modeling and animation to refine the development of realistic simulations. The ability of Blender to be integrated with python, automates various tasks for generating controlled simulations for the experiment while customizing the workflow for specific needs and goals. Through various rendering engines provided in Blender, delivery of high-quality images is possible. These are particularly advantageous while analyzing the resulting images and interpreting the data. With the help of real-time rendering capabilities, Blender speeds up the development and testing, with minimum information loss. As an Open-Source software, Blender provides various support and resources for the community. Programming languages like Python assist in testing the approach while also developing algorithms. Libraries like OpenCV [25] help in implementing computer vision and image processing algorithms, effortlessly.

## 4.2 Experimental Approaches

Optimal positioning of camera systems, a fundamental element for dynamic event observation, has been studied thoroughly where the inspiration has been drawn from computational geometry [3] and Art Gallery problem [38]. Optimal placements of agents for capturing events have been used in various fields, from security, urban planning to space mission planning. Utilizing optimal placement of agents/cameras for observing desired events, by implementing Set Cover Problem[31] has been a motivation for developing the approach of the research.

# 5 Methodology

## 5.1 Simulation Framework Development and Environment Modelling

Development of an efficient simulation framework is essential for effectively modeling dynamic events and evaluating camera configuration performance. For building a virtual environment for simulation, we used Blender software, an open-source 3D creation suite. Because of Blender's many features and adaptability, we were able to precisely create and modify virtual environments, making sure that our simulations closely resembled actual events.

During our project, we developed our experiments leveraging the facilities of the Tartu Observatory Space Mission Simulation Center [48] , mentioned as 'spacebunker' in the thesis. The development of simulation framework, involved careful layout planning of the simulation environment to make it as close to the physical environment as achievable. The section of the spacebunker that we used for our experiment, was reconstructed digitally using the pipeline [49]. Implementing this setup we were able to simulate, observe and analyze various dynamic events. The simulation environment was a vital part in our experiments for gathering and analyzing the data, it provided the means to observe dynamic events under controlled conditions.

The camera position, orientation and FoV, within the simulation environment was configured to be similar to the physical environment, the spacebunker. The parameters were carefully calibrated to optimize the efficacy of observation, reduce blind spots, and guarantee comprehensive coverage of the target.

Throughout the simulation phase, rendering techniques were fundamental since they enabled us to generate visuals from various camera positions inside the simulation environment. The process of rendering involved producing clear images that showed the scene from multiple angles. We were able to evaluate the effectiveness of camera configurations and combinations in various settings and portray the dynamic nature of events by simulating many perspectives of the simulation environment.

## 5.2 Camera Configuration Generation (Sampling Techniques)

The utilization of sampling techniques was instrumental in producing a variety of camera configurations in the virtual environment. In order to guarantee thorough coverage of the scene, these techniques comprised methodical approaches to choosing camera positions and orientations. The simulation could mimic a variety of observation scenarios by sampling from a range of predefined parameters, such as spatial distributions and FoV. The idea was to design camera setups that provided the best possible coverage while capturing important details of dynamic events from various angles.

Volumetric sampling of chaotic system, was used for camera configuration generation. This sampling technique provides camera coordinates inside the environment that gives a thorough coverage of the scene. This technique allows us to create camera setups that can completely capture the volume of the scene.

The chaotic sampler, proposed by Paul[41] is used to sample the agent locations within the volumetric space (see 3.7) in the given boundary conditions. Sampled points, returned as discrete solutions are guaranteed, however, the sample exploration in the volumetric space is always an asymptotic solution.

## 5.3 Event Simulation and Analysis

In order to accurately imitate the real-world observation settings, images from each camera were captured. By using Computer Vision methods, with the help of OpenCV library, we were able to locate and examine the target in relation to the FoV of view of each camera. The first step is to set up the virtual environment with the predefined camera configurations. After the cameras with the given positions and orientations are set, images are captured from each camera to replicate the observation procedure. These images provide the foundation for analyzing the coverage and visibility of the target object from various angles.

Finding and analyzing the target object in the rendered images were essential for our analysis, we detected and located [9], [51], [39], [8] the target object in each image and extracted relevant information about its position. After the target object had been located, its position inside each camera's FoV of view was analyzed. This involved determining the intersection areas between the boundaries of the target object and the camera's FoV of view. These intersection areas provided important information about the object's coverage and visibility from the viewpoints of each camera. By analyzing the intersection areas, we can evaluate the effectiveness of each camera, hence each set of cameras. This analysis helped us in locating blind spots in the scene. It also enabled us to assess the target object's overall visibility and the quality of the captured images. The approach helped in defining the scores of the camera setups.

## 5.4 Algorithms

### 5.4.1 Environment Coordinate transformation

To enable adaptation of the simulation environment to the physical environment, the constraints of the environment were defined and the scale factor of blender was decided. The translation transformation [46], [3] and scaling was applied on the simulation environment coordinates to obtain corresponding physical world coordinates. A reference point was selected which served as a standard point for relating both environments; this point provided correspondence between the two coordinate systems.

The algorithm calculates the scaling factor based on the ratio of constraints corresponding to the simulation system to the physical system along with the scale used in simulation modeling. The scaling factor concludes how simulation coordinates would be modified to be viable in the physical world. After calculating the scale factor for transformation the algorithm loops through the dataset and transforms the blender coordinates into the physical environment coordinates which were then used to carry out experiments in the spacebunker.

**Algorithm 1** Coordinate transformation

Define:
Physical environment constraints $V_{xyz}$ ;
Simulation environment constraints $V_{XYZ}$ ;
Simulation scale $s_{XYZ}$ ;
Reference coordinates $x_r, y_r$ ;
Read simulation environment coordinates $X,Y$ from $df$ ;
Compute scaling factor:
$S_x = (V_X/V_x) \times s_X$ ;
$S_y = (V_Y/V_y) \times s_Y$ ;
**while** $df \neq 0$ **do**
    Extract $X$, $Y$ ;
    Calculate physical environment coordinates :
    $x = x_r + X \times S_x$ ;
    $y = y_r + Y \times S_y$ ;
    Append df ;
**end while**

### 5.4.2 Camera Score Calculation

A camera score calculation function was defined which evaluates how well a system of cameras can monitor the given event. The scoring function depends on the parameters like, the distance between the cameras and the target, the FoV of cameras and the coverage area. The objective of the algorithm was to present quantitative analysis of camera systems, investigating the ability of each configuration for optimal observation of the given event. For calculating scores of the camera system the target coordinates, target dimensions, focal length of cameras, camera angle and sensor size was defined. These factors influence the FoV of a camera.

For an effective observation with a minimum number of observing sensors, three camera systems were employed. The algorithm iterated through each combination or system of cameras calculating the Euclidean distance between the camera and the target, the FoV (based on focal length, camera angle and the sensor size). The maximum and minimum coverage were computed which defined the boundary of the FoV of a camera, these coverage were used to evaluate the intersection between the camera FoV and the target object. The formula for calculating the maximum and the minimum coverage were derived by assuming the FoV area to be rectangular around the camera. Hence, for evaluating the boundaries of the coverage, half of FoV was added for maximum coverage and half of FoV was subtracted for minimum coverage.

The intersection area of camera FoV and the target object exhibited an important role during the evaluation of the scores. To measure the intersection area, the overlap between the camera FoV and the target object were calculated where the results were compared to zero to prevent negative overlaps in case of no intersection along a particular axis. Using the overlap along both x and y direction, the intersection area was measured. The intersection area depicts the area where the camera FoV and the target object overlap in both x and y axes. For calculating the score the distance and the intersection area were combined for each camera. The analysis was made with respect to the maximum possible score of the cameras which provided a reference point for the score calculation.

**Algorithm 2** Score Calculation

Define:
Target coordinates $x_t, y_t$ ;
Target dimensions $W,H$ ;
Camera angle $\omega$;
*ReadCameracoordinates*$x_c, y_c$ *from df ;*
***while*** $df \neq 0$ ***do***
*Group cameras in sets of 3,* $A_c$ ;
***while*** $i \leq len(A_c)$ ***do***
*Calculate distance :*
$d = \sqrt{(x_t - x_c)^2 + (y_t - y_c)^2}$ ;
*Calculate FoV :*
$fov = 2 * \arctan(sensorsize/2 * focallength) + \omega$ ;
*Maximum coverage within fov of a camera:*
$Max(fov(x)) = x_c + fov \times 2$
$Max(fov(y)) = y_c + fov \times 2$
*Minimum coverage within fov of a camera:*
$Min(fov(x)) = x_c - fov \times 2$
$Min(fov(y)) = y_c - fov \times 2$
*Calculate overlap :*
$x_{overlap} = Max(0, Min(Max(fov(x), (x_t + W)) - Max(Min(fov(x), x_t))))$
$y_{overlap} = Max(0, Min(Max(fov(y), (y_t + H)) - Max(Min(fov(y), y_t))))$
*Calculate Intersection area :*
$a = x_{overlap} \times y_{overlap}$
*Calculate Score :*
$\delta = d + a$
***end while***
*return* $\delta$;
***end while***

### 5.4.3 Object Detection and Analysis

After data gathering the target information was extracted from both physical and simulation environment images. This process involved computer vision techniques to detect and analyze the target object within each image captured and rendered during experiments and simulations. Algorithm 3 defines the steps involved for detecting the target object.

The results from this procedure were further utilized for analyzing the intensity of the target with respect to its environment along with computing coverage area of the target within the camera's FoV. A reference image capturing the target object within the environment was used with an image of the environment without the target object to extract or subtract out the desired target. After subtraction, the resulting image was converted to grayscale along with mapping the pixel values to assist with establishing a threshold for the binary mask. The mask helped in emphasizing the area associated with the target in the image. The process was iterated through every image and a bounding box was generated around the detected target object in each of these images.

**Algorithm 3** Object Masking and Analysis

Reference Image of target object with background/environment $I_f$ ;
Image of background/environment without target object $I_b$ ;
Images from experiment with target/object to analyze $I_t$ ;
**for** each image **do**
Subtraction :
$diff = I_b - I_f$ ;
Convert to grayscale :
$Gray_{diff} = ConvertToGrayscale(diff)$ ;
Map pixel values :
$Map_{diff} = NormalizedTo255(gray_diff)$ ;
Binary Mask :
$mask = Threshold(map_diff)$ ;
Reduce Noise :
$mask = MorphologicalOperation(mask)$ ;
Extract object :
$target = ApplyMask(I_t, mask)$ ;
**while** target is True **do**
*Draw bounding box* ;
*Calculate Coverage area* ;
**end while**
return *target + bounding box* ;

## 5.5 Spacebunker Experiment Setup and Data Collection

The spacebunker setup and data collection aimed to validate the performance of the cameras and verify the accuracy of simulation results. By capturing images of the target object from various camera angles, we evaluated the visibility and coverage of the object. This technique of gathering empirical data was essential to assessing the simulation framework's dependability and suitability for practical use. The spacebunker provided a controlled environment with minimal disturbance from the outside world, which allowed us to precisely control variables and conditions during the experiments. It was possible to make insightful comparisons between the outcomes of the experiments and the simulations because of the controlled setting that guaranteed consistency in data collection.

Different camera setups were tested within the spacebunker to take images of target that were positioned at predefined coordinates. These camera configurations included various angles and distances from the target object, simulating observational scenarios. We were able to compare the efficiency of various camera configurations in capturing meaningful data by varying these parameters.

During data collection we recorded images captured by each camera. A comprehensive analysis was performed on the collected images in order to quantify different variables pertaining to object visibility and camera performance. We employed image processing methods and computer vision algorithms to derive relevant information from the collected images. The analysis provided evidence of camera performance by assessing important criteria like object detection accuracy, image quality and coverage area. Comparing the results obtained from simulation with the spacebunker experiments helped us in validating the accuracy and reliability of our approach. We could also evaluate the quality of the simulation for the real-world scenarios by comparing the experimental observations with simulated predictions. In order to find possible areas where the simulation framework could be improved upon, differences or deviations between the results of the experiment and the simulation were closely examined.

Moreover, the spacebunker experiments yielded significant insights into the real-world challenges and constraints that arise in observation. We were able to deal directly with elements that are frequently disregarded in simulation models, such as lighting conditions, environmental obstacles, and camera calibration problems. The simulation framework was improved with this practical experience in mind, ensuring its resilience and efficacy in diverse real-world scenarios.

Figure 5.1: Layout of Simulation Environment (reconstructed spacebunker [49]) in blender with three-camera setup.

# 6 Data Gathering

## 6.1 Simulation Environment

During the research, we developed simulation environments for dynamic events like volcanic eruption, two body collision, comet flybys. Using blender we were able to generate extensive models for these events and the targets. We also used a reconstructed model of the analogue environment, the Tartu Observatory Space Simulation Center [48] or spacebunker, in the blender environment to help us validate the approach and experiments. During most part of the optimal camera placement experiments we have used the 3D model of comet 67P [22] as the observational target object. For Data gathering in a simulation environment we started with setting up our environment and using the generated coordinates to make camera systems for our experiments. The target object positional values and the constraints were defined and each system with three cameras setup were iterated to capture images from each camera of every system. For clarity and organizing purposes we developed a naming convention for storing the data gathered. The naming convention used for the cameras was set to:

e_x_camera_y (where, e is the experiment number, x is the number of System and y is the number of camera in that system).
For experiment-1:
system-1 had images captured and rendered by :
1_1_camera_1, 1_1_camera_2 and 1_1_camera_3
system-2 had images captured and rendered by :
1_2_camera_1, 1_2_camera_2 and 1_2_camera_3,
and so on, for every system in every experiment. This helped in ensuring consistency and better identification of images from each camera within different systems during the experiments.

## 6.2 Analogue / Physical Environment

The Analogue experiments were performed in the Tartu Observatory Space Simulation Center, or spacebunker. The center offers facilities for testing space instruments and to replicate conditions close to real space missions. The Center has a long corridor or tunnel which provides testing grounds for instruments and experiments that require greater freedom of movement. Our experiments have been conducted in this corridor area to attain better placement configurations. The analogue experiment setup included,

- A 3D model of the comet 67P [22] similar to the one used in simulation experiments, with fixed position

- Illumination source for illuminating the target object, with fixed position
- Three camera setup within each system

The transformation was performed on the simulation coordinates to replicate the experiment setup within the spacebunker. The coordinate transformation ensured consistency between the simulation and the physical environment. After setting up the environment with the target object and the light, the cameras were placed according to transformed coordinates; the images were captured iterating through each camera within every system. For data keeping the naming convention for space bunker experiments followed a similar structure, as of simulation experiments.

# 7 Experimentation and Analysis

To assess the effectiveness of the created simulation framework and the optimal camera systems, experiments were carried out. Through these experiments, numerous dynamic events were simulated, and the efficacy of various camera setups in capturing important event details were examined. Through methodical manipulation of camera locations, we evaluated the resilience and flexibility of the modeling structure. The suggested approach's advantages and disadvantages were identified through examination of the experimental data, which contributed to determine future developments and adjustments. The initial experiments, Experiment-01 and Experiment-02 tested the methodology by conducting simulation of dynamic events.

## 7.1 Experiment 01: Comet Flyby

**Experimental setup and Outcomes**

The Experiment 01 consisted of modeling a simulation environment to imitate a comet flyby event. A 3D model for the comet-like environment was generated and animated to move on a trajectory. The model comet was simulated to display various properties of an actual comet, including released gas and dust forming a glowing coma and a tail.

The simulation environment was set up with three camera systems, which were randomly placed within a defined boundary to make observations of the comet flyby. The animation of comet flyby was iterated numerous times with cameras placed at different positions every iteration.(Figure 7.1) The observation was made from each camera within a system, and the captured images were rendered for further understanding. The rendered images from the experiment were analyzed to examine pixel intensity changes throughout the frames of the images captured from every camera. Scores were then calculate for each system. The image analysis facilitated extracting useful information regarding the trajectory and the behavior of the observed comet.

**Result Analysis**

Analyzing the resulting images from the experiment presented crucial information for finding the optimal positions for camera placement. By comparing images captured from different cameras, based on varied positions within a fixed boundary, we were able to determine which configurations observed the event with the most clarity, providing valuable insights of the event. This helped in finding optimal positions to ensure extensive coverage of desired targets. The analysis of the pixel intensity in each of the rendered images enabled tracking or trajectory mapping by defining position of the

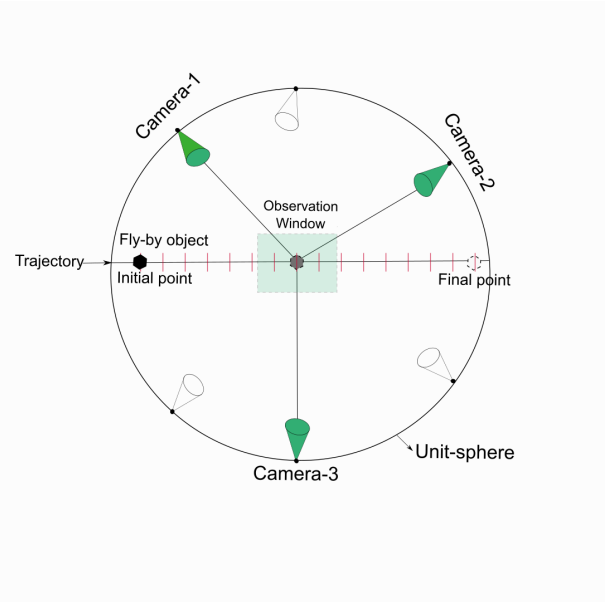



Figure 7.1: Schematic illustrating Event Setup for Comet flyby experiment

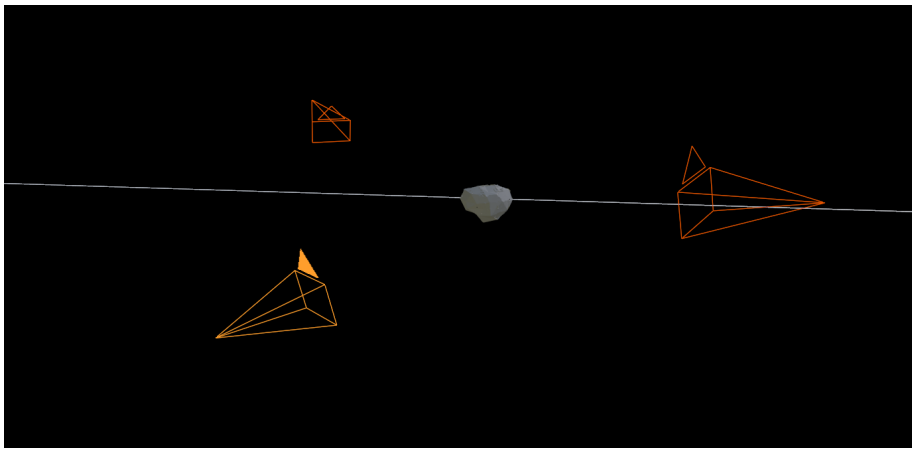


Figure 7.2: Layout of Simulation Environment for Comet flyby experiment in blender with three-camera setup

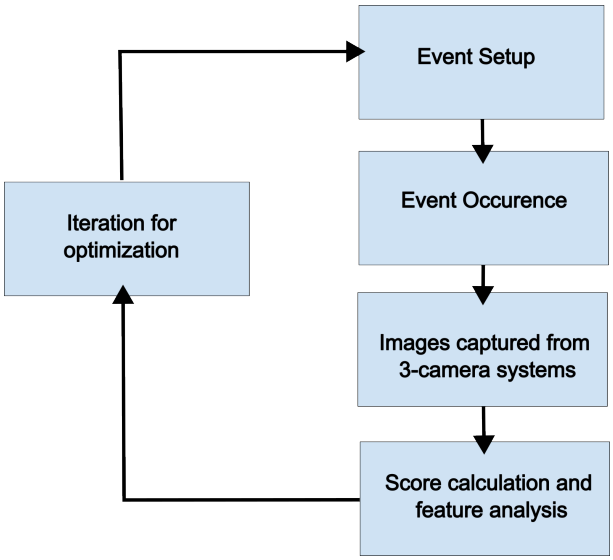
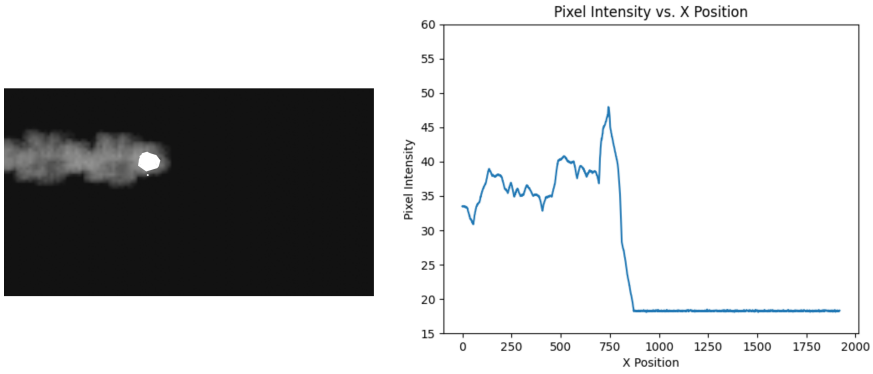



Figure 7.3: Steps involved in comet flyby experiment

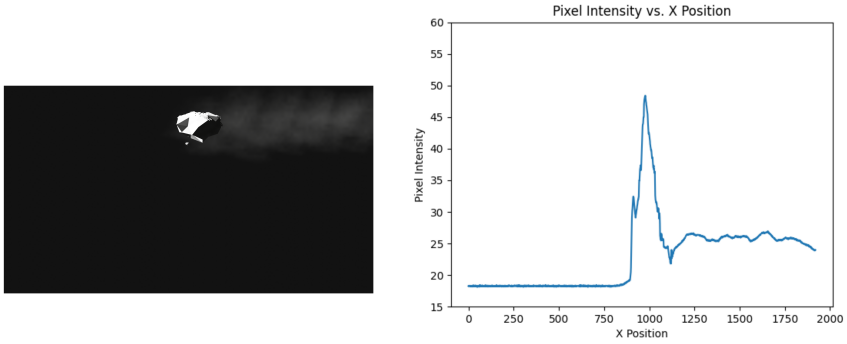
comet in successive frames.

The intensity of the pixels related to the comet model is much higher than its environment, this is similar to real world comet flyby images captured through space missions, where usually, the focus or the most illuminated object in the FoV is the target. By combining the observation from images captured by different cameras through successive frames, we can define the motion of the target comet.

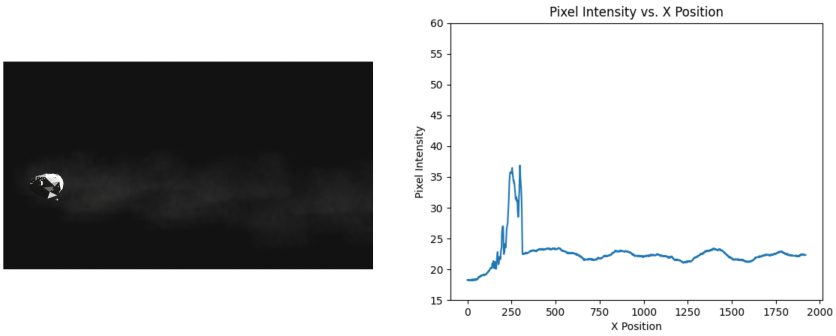
By the Figure 7.4, it is prominent that different cameras provide observation from different angles which can be useful for studying its structure and compositions. The figure 7.4(b) shows that camera-2 in the setup captured the comet structure with more precision. From figure 7.4(a) we can observe that camera-1 provides comparatively better understanding of the tail around the comet. By following the intensity graphs throughout the frames, we can understand and identify the instant at which the comet appeared in the FoV of the camera.




(a) Camera-1




(b) Camera-2




(c) Camera-3

Figure 7.4: Comet flyby images from three different cameras of a system, along with corresponding intensity graphs.

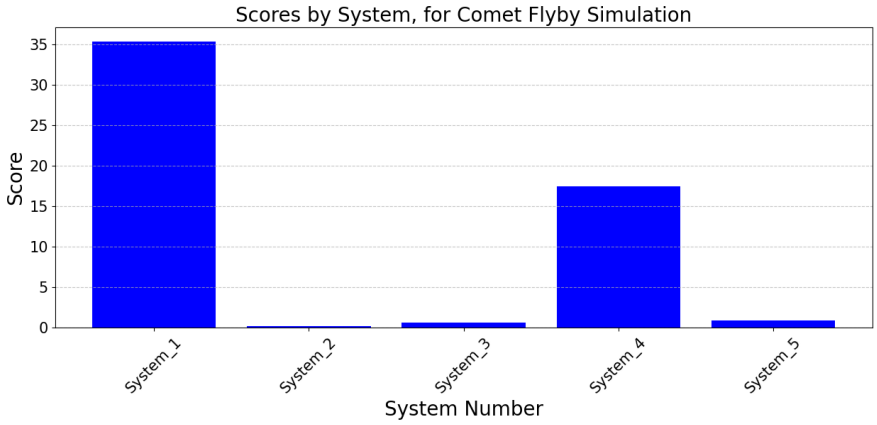



Figure 7.5: Scores of different systems for the comet flyby experiment.

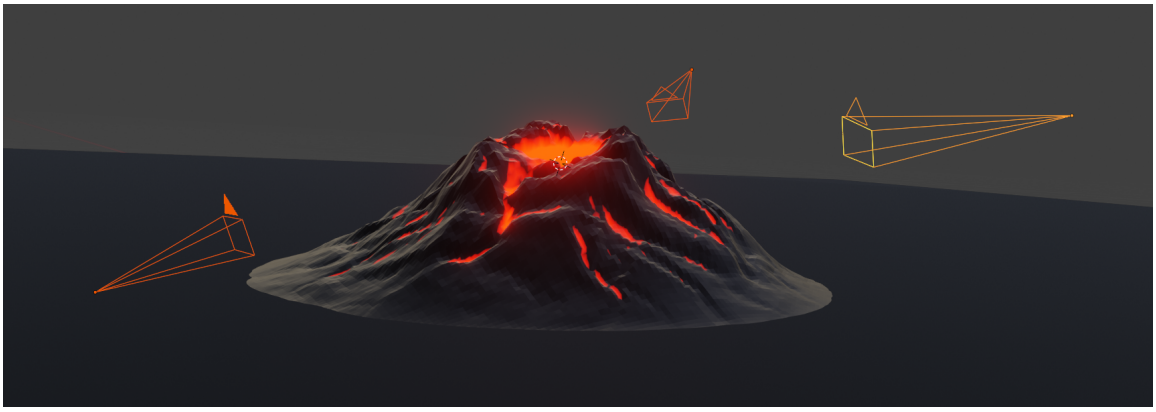


Figure 7.6: Layout of simulation environment for volcanic eruption experiment in blender with three-camera setup

## 7.2 Experiment 02: Volcanic Eruption

**Experimental setup and Outcomes**

The experiment aimed to simulate and examine the behavior of volcanic eruptions. It involved recreating the eruption scenario in a blender environment and observing the dynamics of volcanic plumes and evaluation of camera configuration for optimal observation.

1. **Event Setup**: The Model for volcano was set up and the parameters for volcanic plume was defined. The number of iterations and time steps for the observation was provided.

2. **Dynamic Eruption Loop:** The evolution of volcanic plume was simulated over time with the changes in position and intensity. The eruption intensity was captured for each iteration at each time step.

3. **Peak Intensity Event**:At a random step there was a peak intensity event which shows the highest intensity of plume or eruption. For every iteration this peak event was recorded.

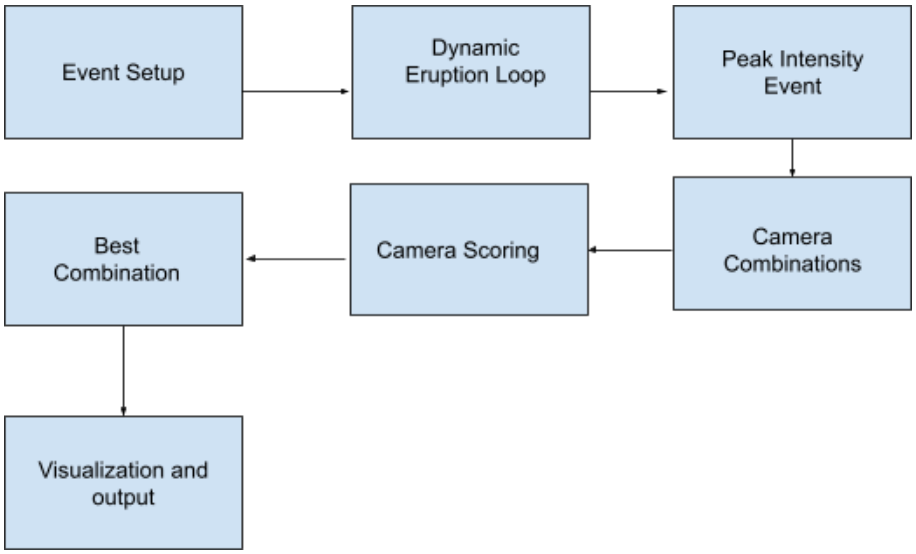



Figure 7.7: Procedure for volcanic eruption experiment

4. **Camera Combinations**: For every iteration, three cameras are generated at random positions, the combination or system of these three cameras are used for observation.

5. **Camera Scoring Function**: A scoring function was defined to evaluate the camera configurations according to their positions and FoV. The scores are calculated by taking into account the visibility of the eruption.

6. **Best Combination**: The performance of each combination was assessed based on the scoring function.

7. **Visualization and Output**: Graphs of intensity vs time of eruption were plotted. To visualize the performance of cameras the normalized combined scores are plotted for each iteration.

**Result Analysis**

Figure 7.6, shows sequential frames of volcanic eruption simulation in a blender environment along with its corresponding intensity graph. It illustrates how by tracking the intensity changes over the frames, it is possible to determine crucial information and instantaneous changes of a dynamic event. As in the case of volcanic eruption, observing changes in intensity graph provides information regarding the instance the eruption occurred. By using such an approach it is possible to find the optimal position to place cameras to observe the desired event.

Figure 7.9, plots scores of every system, for volcanic eruption event in the simulation environment. The plot illustrates which of the camera systems were able to capture the event with more precision. As we can see from the graph, system-2 performed the best and had cameras at optimal positions for observing the dynamic event of volcanic eruption.

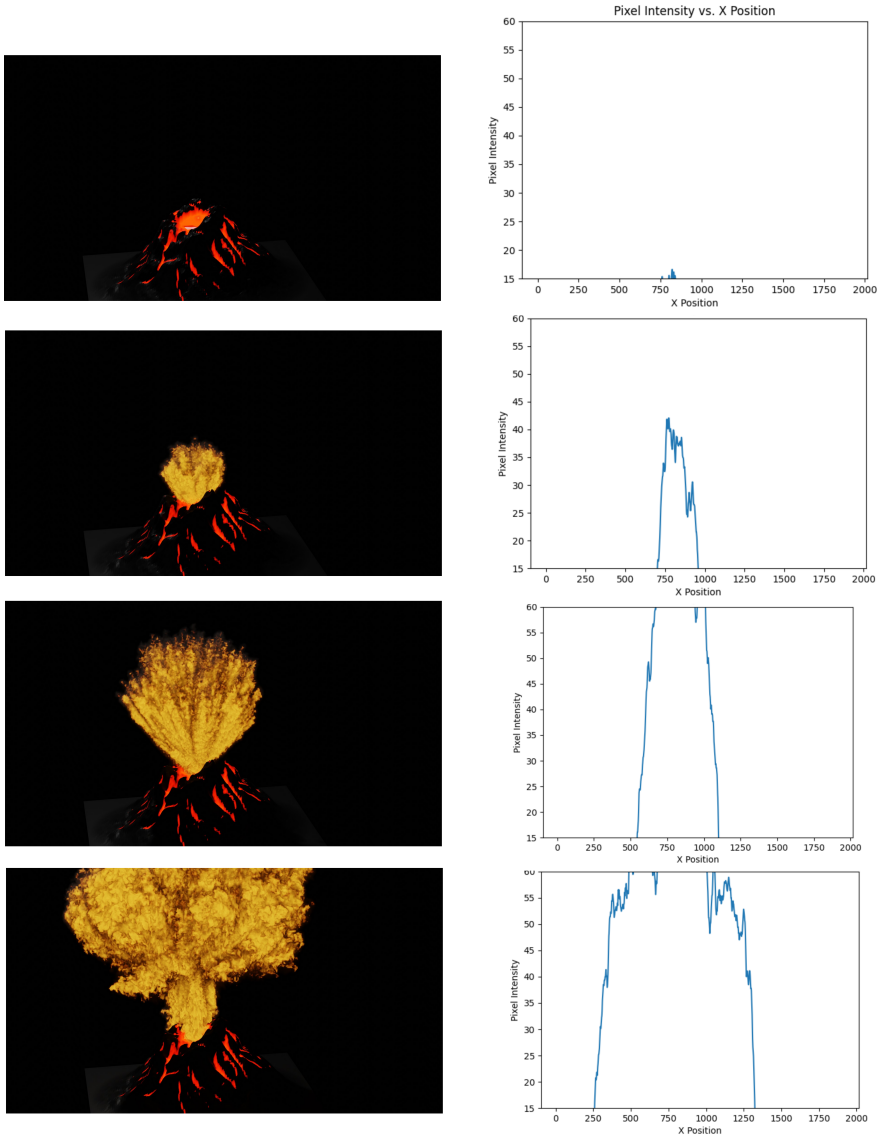



Figure 7.8: Sequential frames of volcanic eruption simulation in Blender environment, with relative plume intensity plots

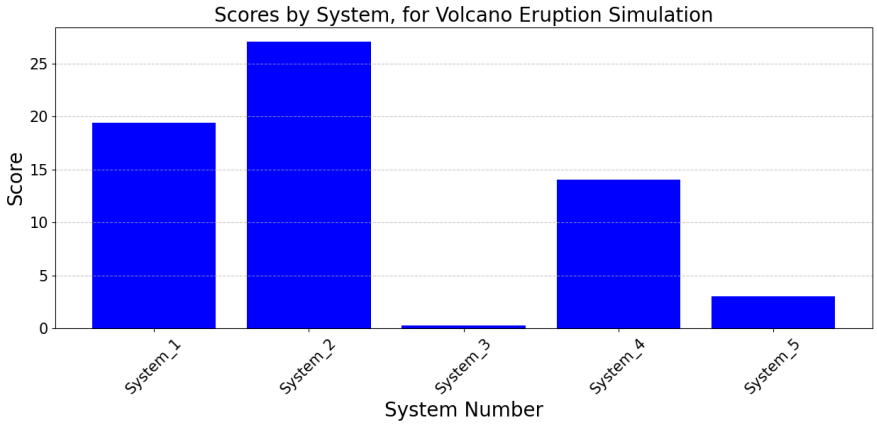



Figure 7.9: Scores of different systems for the volcanic eruption experiment.

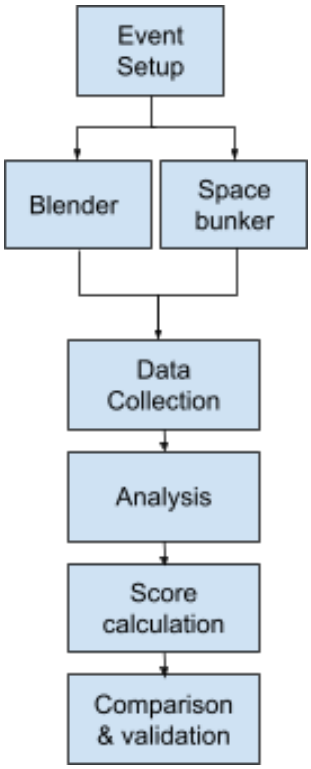



Figure 7.10: Procedure for experiment 03

## 7.3 Experiment 03: Comet Model

**Experimental setup and Outcomes**

The experiment 03 aimed to simulate and examine a comet flyby event at a particular instant of its trajectory by recreating the comet 67p model in a simulation blender environment and imitating the same in a physical environment, spacebunker. The main goal was to observe and evaluate camera configuration for optimal observation. Every campaign included data collection experiments from both simulation and physical environment, within similar conditions or constraints. The iterative approach for designing and optimizing our experiments helped us identify the outliers and determine the optimal solution for our problem.

1. **Event Setup**: The Model for comet 67p was set up in the simulation and physical environment. The generated sample coordinates were simulated in blender and then in spacebunker. The setup included the comet model within the spacebunker with various camera setups.

2. **Sampling Coordinates**: Volumetric sampling of chaotic systems, were used to sample coordinates to be used for experiments. These sampling methods generated camera coordinates within a defined boundary.

3. **Data Collection**: The cameras were placed according to the sampled coordinates and the images were captured and rendered from spacebunker and blender respectively. These image datasets from the physical and simulation environment assisted in analyzing visibility and coverage of the comet from different viewpoints.

(a) Camera-1 (b) Camera-2 (c) Camera-3

Figure 7.11: Target detected in simulation (Blender) rendered images from three-camera setup, System-3, Campaign-01

(a) Camera-1 (b) Camera-1 (c) Camera-3

Figure 7.12: Target detected in Images captured during experiment in physical environment (spacebunker) from three-camera setup, System-3. Campaign-01.

4. **Analysis**: An extensive analysis was conducted to evaluate the efficiency and capability of every camera configuration and to find the optimal system for the observation of the event. The analysis involved image processing with computer vision techniques to evaluate the visibility or intensity and coverage area of the target comet in images.

5. **Score Calculation**: A score function was defined, using the coverage area and the distance between the target and the cameras.(Algorithm 2) The scores were calculated for a three camera system, where the combined scores of the cameras were analyzed.

### 7.3.1 Results for Campaign-01

Images were captured and rendered for each camera within every system. The same setup of systems were used in both simulation and analogue environments for data gathering.

#### Result Analysis

The figure 7.11, contains images rendered from the simulation environment from a three camera setup, system-3. While figure 7.12, shows the images captured from spacebunker experiment where pictures were clicked through the same setup as in simulation environment but with transformed coordinates. These images are from simulation and spacebunker experiments, captured using the same camera system

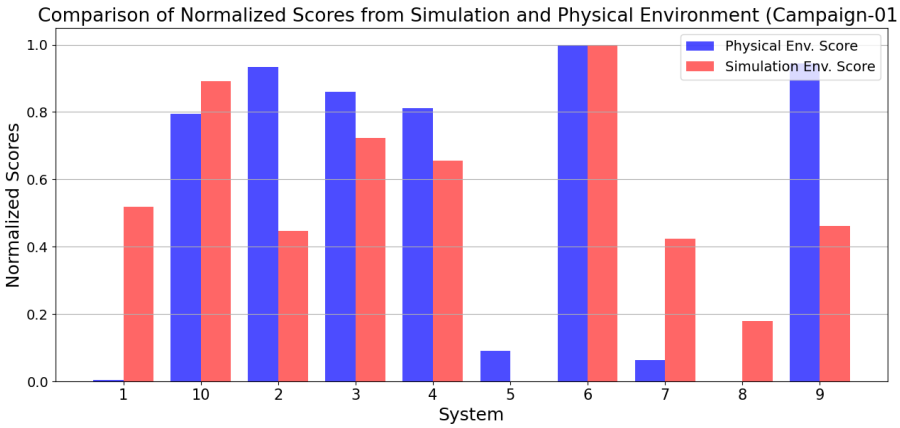



Figure 7.13: Plot showing comparison between the physical (spacebunker) and Simulation (Blender) environment scores over their corresponding camera Systems, for Campaign-01

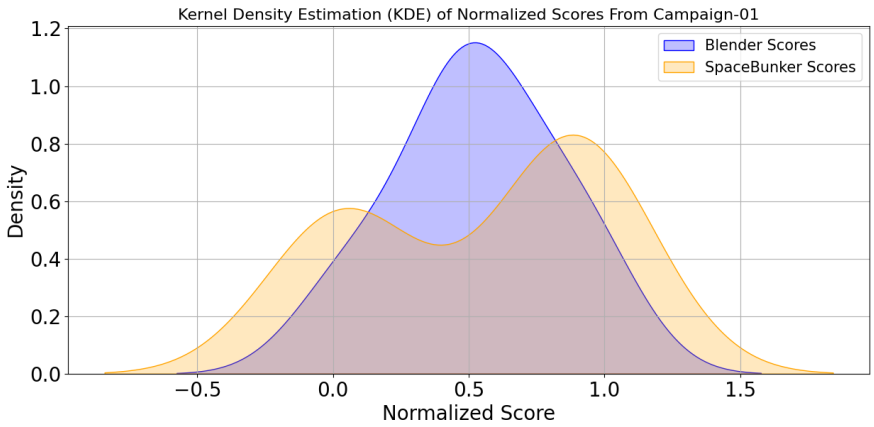



Figure 7.14: KDE of Normalized scores for simulation (Blender) and physical (spacebunker) environment, for campaign-01.

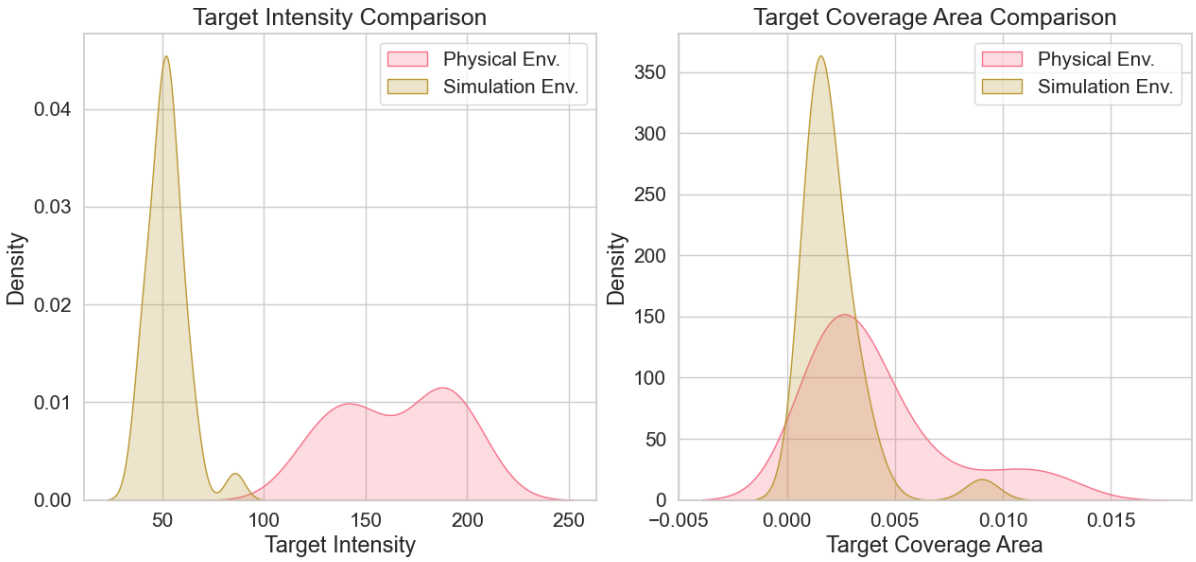



Figure 7.15: KDE plots for intensity and target coverage comparison of physical and simulation images, for campaign-01

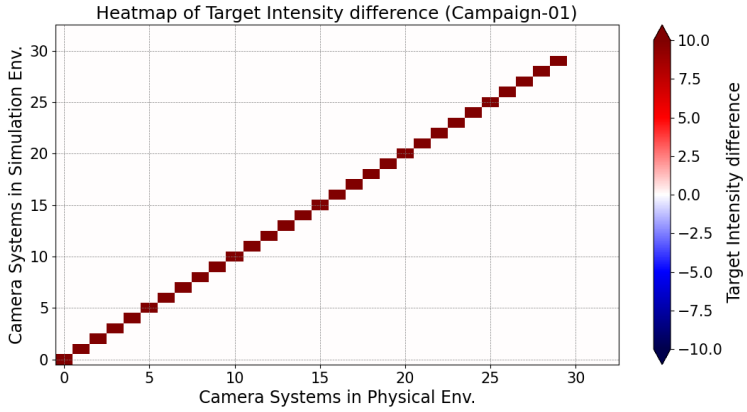



Figure 7.16: Heatmaps for intensity difference between the physical and simulation data, for campaign-01

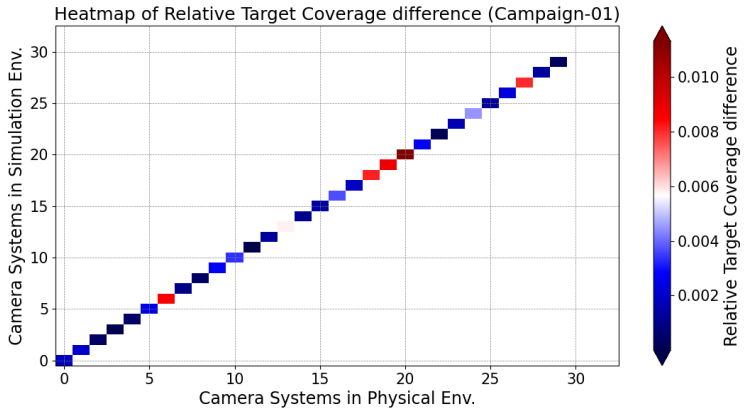



Figure 7.17: Heatmaps for target coverage difference between the physical and simulation data, for campaign-01

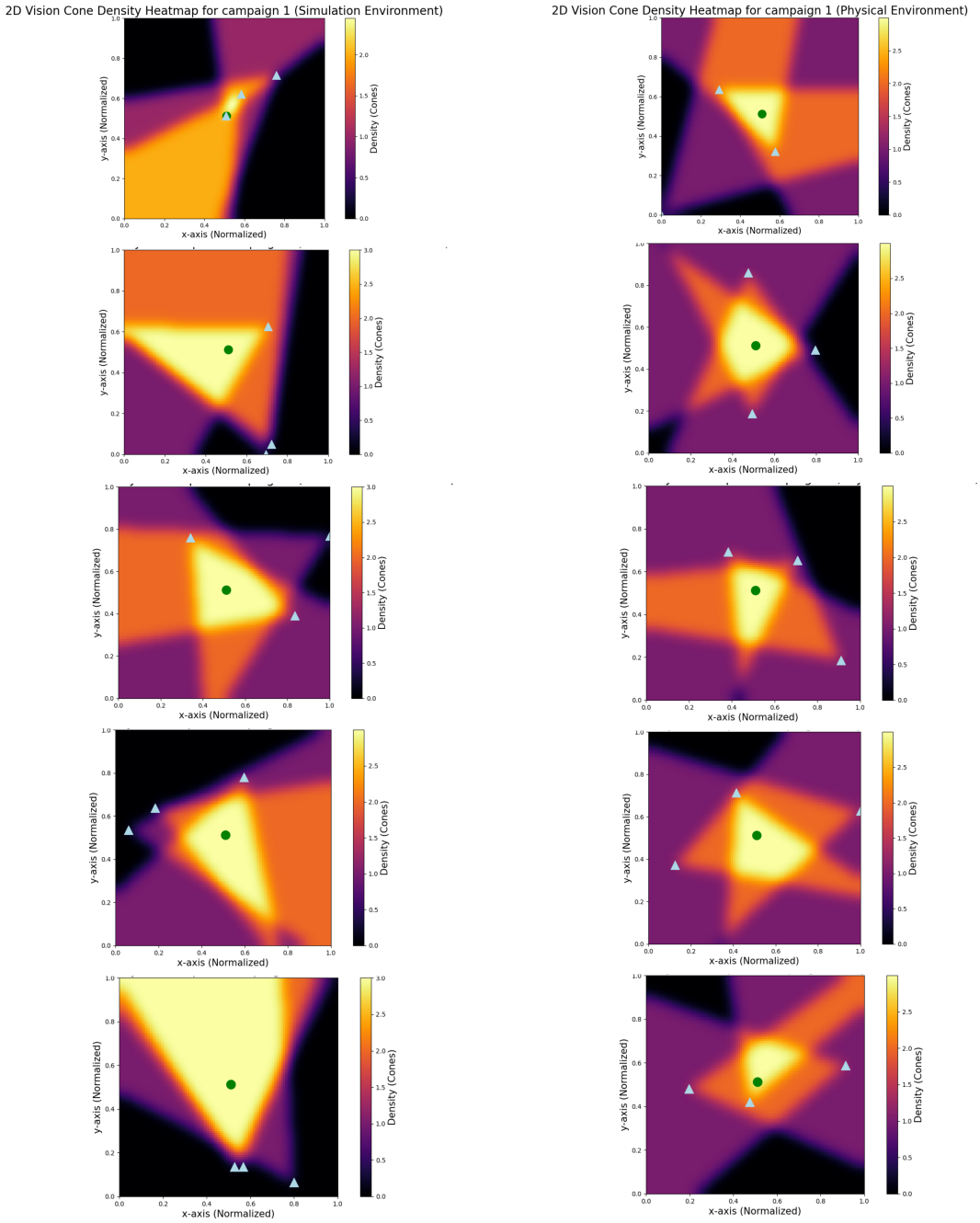



Figure 7.18: 2D heatmap showing vision cone density for simulation environment on the left, and Physical environment on the right. For five systems from campaign-01

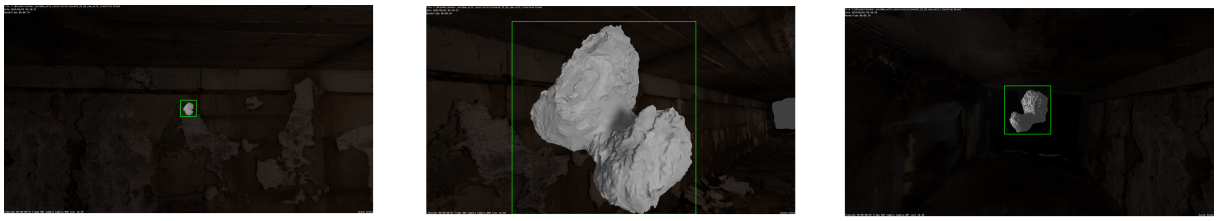
(System-3) updated with bounding boxes around the detected target object.

Figure 7.13, depicts, Kernel density estimation (KDE) of normalized scores from spacebunker and Blender environment. The plot compares distribution of scores obtained from these two environments, by comparing the distribution of the normalized scores we can analyze the extent of similarity between these two distributions. The overlapping areas indicate regions where the scores from both experiments have similar scores. The total overlapping area between the scores distribution from blender and bunker environment is approximately 22.53 %. The KDE plot between the blender and bunker scores assists in validating the approach by comparing the similarity between the distribution of the resultant scores of different systems.

Similarly the KDE plots for intensity and target area coverage (Figure 7.15) illustrates how these parameters are distributed for data gathered through spacebunker and blender environments.Through these KDE plots we note that there is minimal intersection between the intensity values of the target in the images captured from bunker and blender experiments. While the coverage area of the target has about more overlapping area between the data gathered from both environments.

The heatmap of intensity error (Figure 7.16) depicts the disparity in the intensity of the target between the data gathered from the spacebunker and the blender experiment. From the plot its clear that the difference between the intensity of the target has been high, between the images captured in simulation and physical environment.The x-axis labels the camera number in the bunker environment and the y-axis is the cameras in the blender environment, for Campaign-01. Unlike the Intensity difference heatmap, the Heatmap of target coverage area (Figure 7.17)has multiple ranges of error which indicates variation in the target coverage area through simulation and physical experiment. Through the heatmap plot it is prominent that the difference between the coverage area is very low for most of the cameras, while cameras like camera-7, 19, 20 and 28 have the highest discrepancy between the target coverage area observed from spacebunker and blender environment.

The 2D vision cone density heatmap (Figure 7.18) illustrates density of vision cones for each system of 3-camera setup. The cameras are placed using coordinate samples generated randomly. Vision cones represent the FoV of a camera, and vision cone density defines how many vision cones are present per unit area of an environment. Illustrating vision cones from each camera in a system helps in understanding the possible coverage areas within the observation window. Higher density indicates better coverage while low density could lead to blindspots or gaps in the observation. The above heatmaps, shows vision cone density for five camera systems in both, simulation and physical environment. The map demonstrates how by changing the positions of the cameras in an environment, the coverage area changes. This shows the importance of camera placement for optimal observation of an event.



(a) Camera-1 (b) Camera-2 (c) Camera-3

Figure 7.19: Target detected in simulation (Blender) rendered images from three-camera setup, System-3, Campaign-02

(a) Camera-1 (b) Camera-1 (c) Camera-3

Figure 7.20: Target detected in images captured during experiment in physical environment (spacebunker) from three-camera setup, System-3. Campaign-02.

### 7.3.2 Results for Campaign-02

**Result Analysis**

Figure 7.19, shows images from 3-camera setup in the simulation environment, while figure 7.20, shows images captured from the same camera system (System-3), with transformed coordinates, in a physical environment or the spacebunker. The images also illustrate the target object detection by showing a bounding box around the target. The plot (Figure 7.21) illustrates comparison between the normalized scores from simulation and physical environment. The plot displays how different or similar the scores of simulation and physical experiments are, for particular camera systems. We can observe that the highest scoring camera system is the same in both environments. The optimal camera system for observation of the target is System-3 in both simulation and physical environment.

The KDE plot of normalized scores (Figure 7.22) shows how the scores are distributed for both simulation and physical environment. We can see that the area of overlap between the score distribution of blender and spacebunker is higher than the one acheived by Campaign-01. This indicates that the cases where the same camera systems in both environment have similar scores, is higher for the experiment of Campaign-02 rather than Campaign-01.The overlapping area between simulation and physical environment scores KDE, from Campaign-02 corresponds to 27.30% ; which is approximately 5% more than the resulting overlapping area from Campaign-01. Similarly the KDE plots, Figure 7.23, for the target intensity and coverage area show better results than the ones obtained through Campaign-01.

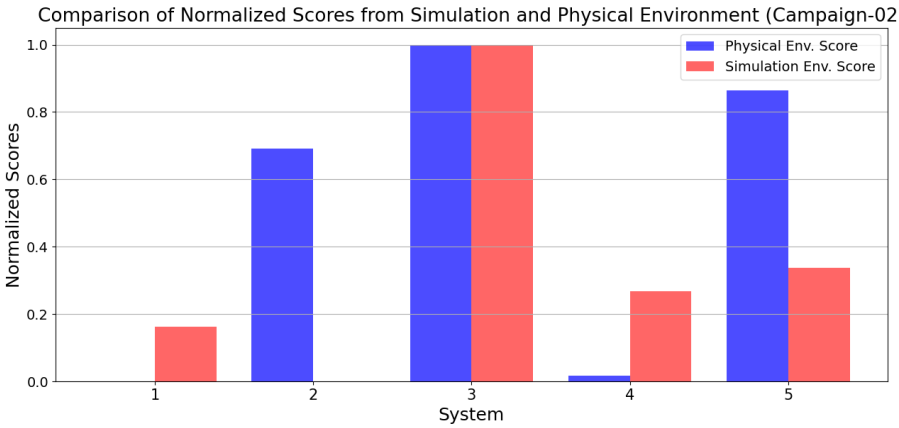



Figure 7.21: Plot showing comparison between the physical (spacebunker) and Simulation (Blender) environment scores over their corresponding camera systems, for Campaign-02

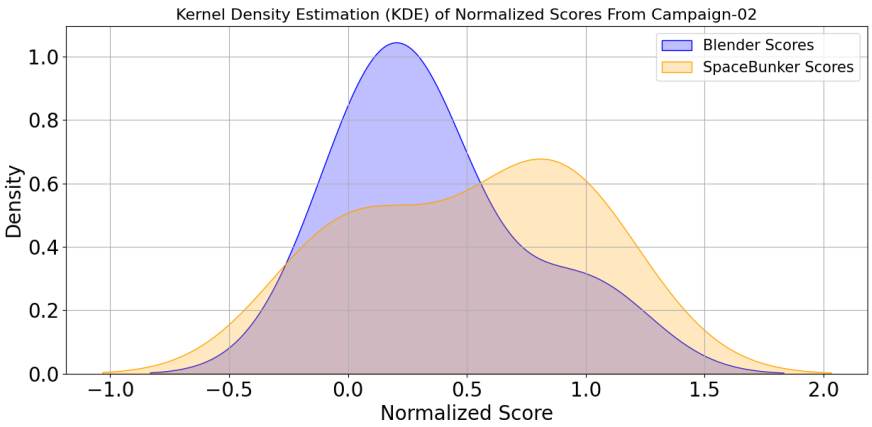



Figure 7.22: KDE of normalized scores from simulation (Blender) and physical (spacebunker) environment data, for campaign-02.

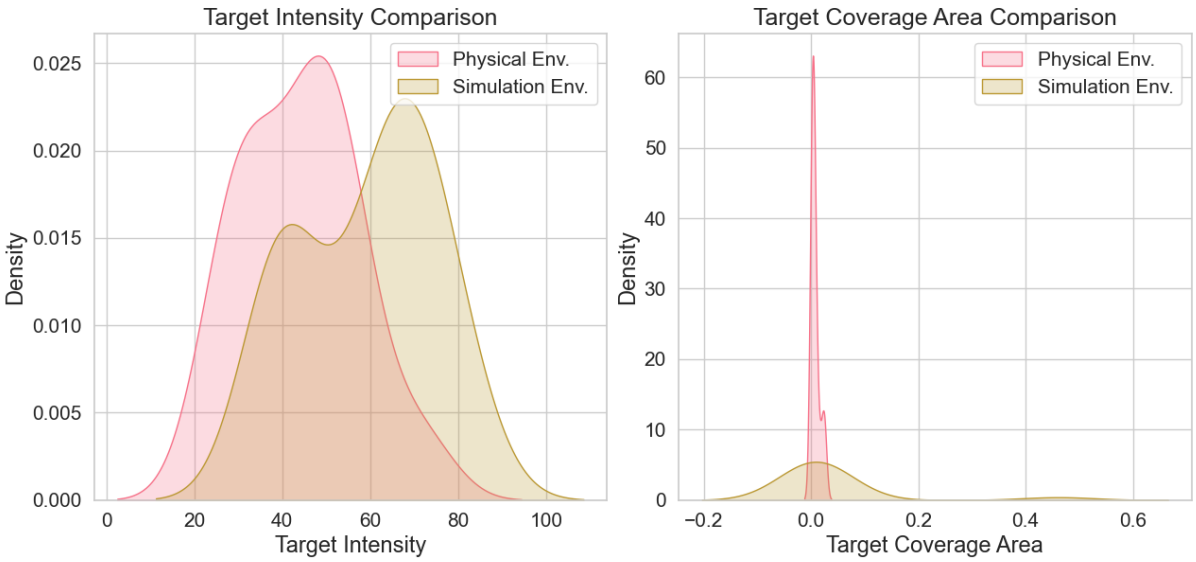



Figure 7.23: KDE plots for intensity and target coverage comparison of physical and simulation images, for campaign-02

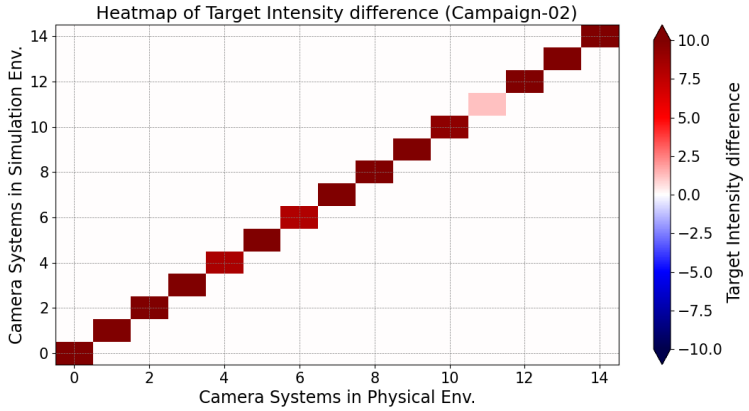



Figure 7.24: Heatmaps for intensity difference between the physical and simulation data, for campaign-02

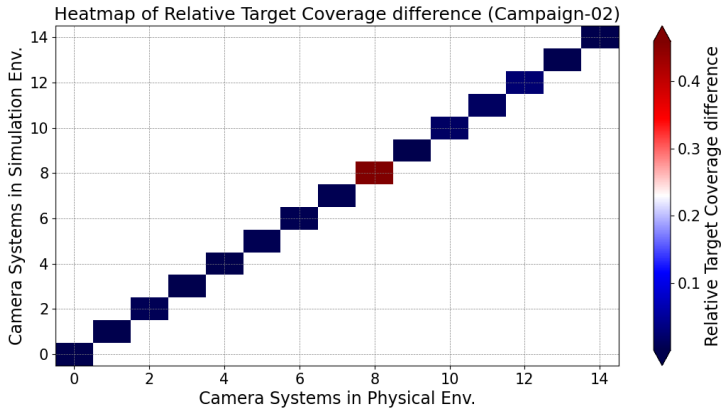



Figure 7.25: Heatmaps for target coverage difference between the physical and simulation data, for campaign-02

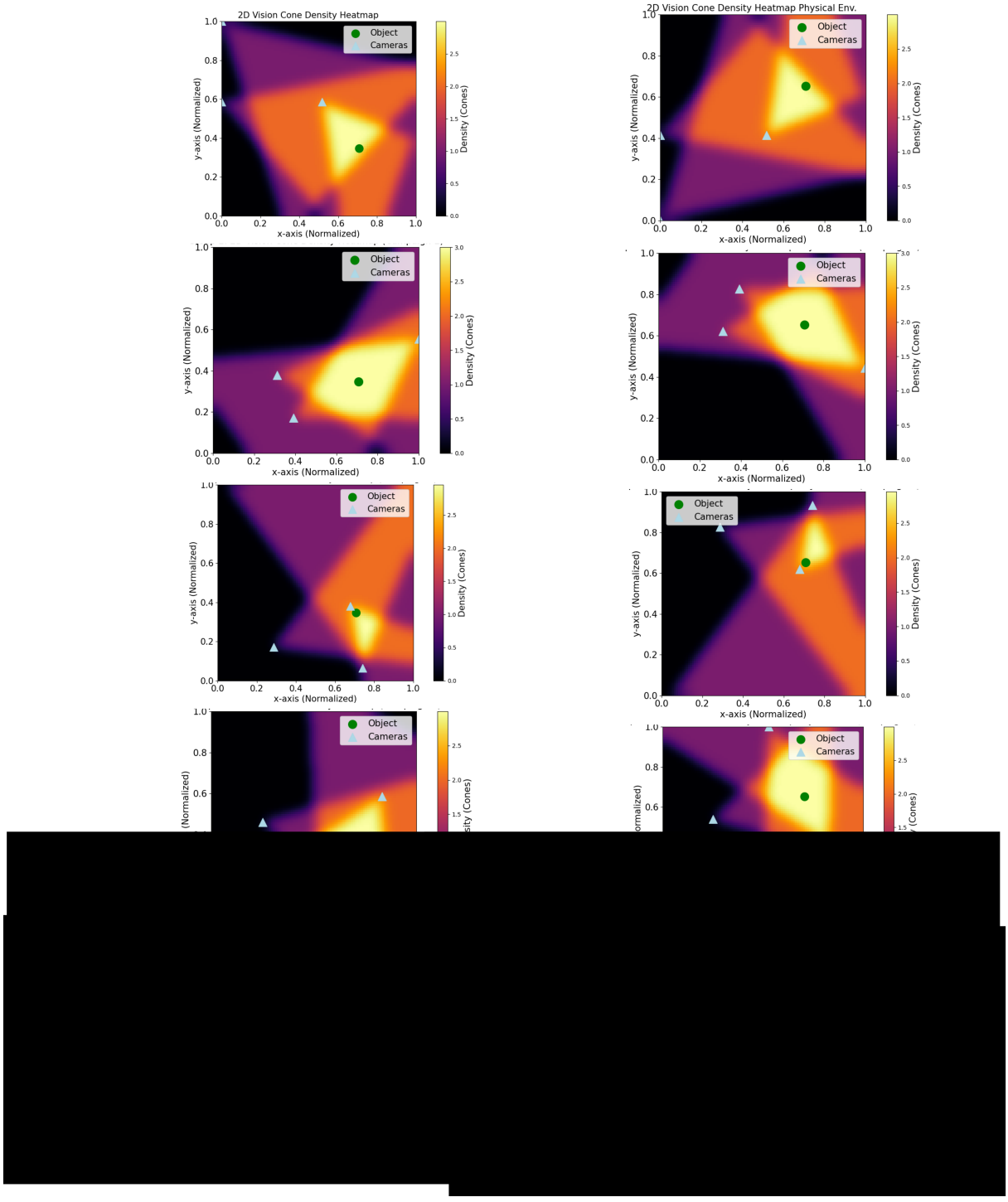



Figure 7.26: 2D heatmap showing vision cone density for simulation environment on the left, and Physical environment on the right. For five systems from campaign-02

The difference in the target intensity and the coverage area between the same cameras in different environments, is also illustrated through the heatmaps (Figure 7.24). The heatmap for the coverage area, shows the difference between the coverage area calculated in the blender and spacebunker environment. It is evident that the target coverage area has minimal error for most of the cameras, except camera-9 which shows high discrepancy in the results between both environments. The intensity error heatmap shows that there is high error or difference in the resultant intensity values of the target between the simulation and physical environment for the same cameras, except for camera-12 which has less error compared to other cameras.

The 2d vision cone density heatmap (Figure 7.26) illustrates the positional information of the cameras, in Campaign-02. The coverage capabilities have been displayed through the FoV of each of the cameras. The density of the vision cones are exhibited through the overlapping FoV of each camera in a system.

# 8 Conclusion

An approach was demonstrated for optimal agent placement for dynamic event observations. By leveraging computational geometry, computer vision techniques and optical sciences a methodology was developed to identify optimal agent configuration for observing a dynamic event. The development of a scoring function for cameras on the basis of intersection areas between camera's FoV and target object, the camera system effectiveness was quantitatively assessed for determining the optimal camera system. Throughout the experiments a three-camera setup has been used for making observations of the events. The number of cameras were maintained at a minimum, to adhere to the balance between capturing comprehensive data and minimizing resource utilization. By using only a necessary number of cameras, we ensured that our observation system remains efficient and cost-effective. Though it is probable that using additional cameras will increase the coverage, they will also increase the complexity and use of additional resources.

Through multiple campaigns or experiments, a comparison was made to showcase the feasibility of the approach for defining the optimal coordinates for the agent/camera systems for monitoring dynamic events. The approach focused on determining the coverage of the target through placing cameras at various points in a given environment with boundary conditions defined. The experimental outcomes discussed in the thesis demonstrated utilization of the simulation environment for generating a dynamic event for experiments which can be validated through equivalent tests in an analogue environment.

# 9 Further Contributions

The research aims to contribute towards advancement in dynamic event observation and optimal positioning of camera systems. There are several areas which can further the use-case for such an approach, few such areas have been discussed as following:

- **Integrating Machine learning**
  Target detection and tracking could be improved by integrating machine learning algorithms. This could generate adaptable systems, which are not constrained by variations in the observational environment such as changing lighting conditions. This could also extend the suitability of the approach for multiple-target observation.

- **Autonomous Optimization**
  Incorporating autonomous calibration algorithms, to enable cameras to adjust the parameters like focus and exposure autonomously based on real time feedback from the environment. Self optimization to enhance configurations to improve performance.Adding mechanism for self diagnosis to detect and resolve issues onboard the observing agent.

- **Multiplane exploration**
  Adding multiple observation planes for better coverage of the event. The expanded spatial coverage could improve the observation of crucial events from multiple viewpoints by minimizing blind spots. Using multiple planes could enable switching perspective for better observation which in turn could assist in scene reconstructions, depth estimation to generate 3D representations of models and environments.

- **Upcoming Space Mission**

  - Comet Interceptor [19]
    European Space Agency, Space Mission for observing Long period Comets flyby. The robotic spacecraft could use such an approach to determine the optimal positions for observing comet flyby, and capturing crucial events without loss of important and sensitive information.

# Acknowledgements

I would like to express my gratitude to my very able supervisor **Mr. Aditya Savio Paul**. His constant guidance and support, throughout the tenure has been prime in completion of the work. From technical suggestions to full-filing bureaucratic proceedings, his unwavering and continuous support was encouraging and motivating, which gave me more faith in research and helped me make progress with it every single day.

My heartfelt gratitude to the **University of Tartu**, and **Tartu Observatory** for providing me with the research-oriented environment and infrastructure that made the thesis a possibility.

I extend my appreciation and regards to my parents, who have continuously inspired and nurtured my aspirations.I am grateful to my siblings for being my pillars of strength throughout this journey. A special thanks to friends and colleagues for critical insights, new learning and fun.

The research work explored the disciplines of space engineering. Throughout the research, my respect for scientists and engineers have grown significantly because of their dedication towards advancing knowledge and creating new technologies.

# Bibliography


[1] Michael F A'Hearn. "Comets: looking ahead". en. In: *Philos. Trans. A Math. Phys. Eng. Sci.* 375.2097 (July 2017), p. 20160261.

[2] Randa Almadhoun et al. "Multi-Robot Hybrid Coverage Path Planning for 3D Reconstruction of Large Structures". In: *IEEE Access* 10 (2022), pp. 2037–2050. DOI: 10.1109/ACCESS.2021.3139080.

[3] M.T. Berg, M.J. Kreveld, and M.H. Overmars. *Computational Geometry: Algorithms and Applications*. Jan. 2008. ISBN: 978-3-540-77973-5. DOI: 10.1007/978-3-540-77974-2.

[4] Andreas Bircher et al. "Receding horizon path planning for 3D exploration and surface inspection". In: *Autonomous Robots* 42 (Feb. 2018). DOI: 10.1007/s10514-016-9610-0.

[5] *Blender(Version 3.6)*. 2023. URL: https://www.blender.org.

[6] Dominique Bockelée-Morvan. "An overview of comet composition". en. In: *Proc. Int. Astron. Union* 7.S280 (June 2011), pp. 261–274.

[7] Rowan Border and Jonathan D. Gammell. "Proactive Estimation of Occlusions and Scene Coverage for Planning Next Best Views in an Unstructured Representation". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 4219–4226. DOI: 10.1109/IROS45743.2020.9341681.

[8] Prateek Chhikara. *Understanding Morphological Image Processing and Its Operations*. 2022. URL: https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756.

[9] combine. *Image processing in Sympathy for Data*. 2018. URL: https://combine.se/blog/object-recognition-without-deep-learning/.

[10] Estonian Research Council. *Common European Research Classification Scheme (CERCS) fields of research*. URL: https://www.etis.ee/Portal/Classifiers/Index/26.

[11] Brian Curless and Marc Levoy. "A Volumetric Method for Building Complex Models from Range Images". In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2* (1996). URL: https://api.semanticscholar.org/CorpusID:260380609.

[12] Ronald Daly et al. "Successful Kinetic Impact into an Asteroid for Planetary Defense". In: (Mar. 2023). DOI: 10.48550/arXiv.2303.02248.

[13] Tung Dang, Christos Papachristos, and Kostas Alexis. "Autonomous exploration and simultaneous object search using aerial robots". In: Mar. 2018, pp. 1–7. DOI: 10.1109/AERO.2018.8396632.

[14] Jeffrey Delmerico et al. "A comparison of volumetric information gain metrics for active 3D object reconstruction". In: *Auton. Robots* 42.2 (Feb. 2018), pp. 197–208.

[15] "Chapter 7 - Electrical heating fundamentals". In: *The Efficient Use of Energy (Second Edition)*. Ed. by I.G.C. DRYDEN. Second Edition. Butterworth-Heinemann, 1982, pp. 94–114. ISBN: 978-0-408-01250-8. DOI: https://doi.org/10.1016/B978-0-408-01250-8.50016-7. URL: https://www.sciencedirect.com/science/article/pii/B9780408012508500167.

[16] Newell Martin Edward. "The utilization of procedure models in digital image synthesis". In: (), p. 100. URL: https://collections.lib.utah.edu/ark:/87278/s6hh73p6.

[17] Uğur Murat Erdem and Stan Sclaroff. "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements". In: *Computer Vision and Image Understanding* 103.3 (2006). Special issue on Omnidirectional Vision and Camera Networks, pp. 156–169. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2006.06.005. URL: https://www.sciencedirect.com/science/article/pii/S1077314206000671.

[18] ESA. *Asteroids*. URL: https://www.esa.int/Science_Exploration/Space_Science/Asteroids.

[19] ESA. *Comet Interceptor*. URL: https://www.cosmos.esa.int/web/comet-interceptor.

[20] ESA. *ROSETTA IMAGE ARCHIVE*. 21 June 2018. URL: https://sci.esa.int/web/rosetta/-/60432-rosetta-image-archive-complete.

[21] ESA. *Rosetta, ESA's comet-chaser*. URL: https://www.esa.int/Science_Exploration/Space_Science/Rosetta.

[22] ESA. *SHAPE MODEL OF COMET 67P/C-G*. URL: https://sci.esa.int/web/rosetta/-/54728-shape-model-of-comet-67p.

[23] B Gundlach, M Fulle, and J Blum. "On the activity of comets: understanding the gas and dust emission from comet 67/Churyumov–Gerasimenko's south-pole region during perihelion". In: *Monthly Notices of the Royal Astronomical Society* 493.3 (Feb. 2020), pp. 3690–3715. ISSN: 0035-8711. DOI: 10.1093/mnras/staa449. eprint: https://academic.oup.com/mnras/article-pdf/493/3/3690/32894595/staa449.pdf. URL: https://doi.org/10.1093/mnras/staa449.

[24] E. Horster and R. Lienhart. "Approximating Optimal Visual Sensor Placement". In: *2006 IEEE International Conference on Multimedia and Expo*. 2006, pp. 1257–1260. DOI: 10.1109/ICME.2006.262766.

[25] Itseez. *Open Source Computer Vision Library*. https://github.com/itseez/opencv. 2015.

[26] Nursuriati Jamil, Tengku Mohd Tengku Sembok, and Zainab Abu Bakar. "Noise removal and enhancement of binary images using morphological operations". In: *2008 International Symposium on Information Technology*. Vol. 4. 2008, pp. 1–6. DOI: 10.1109/ITSIM.2008.4631954.

[27] Fan Jiang et al. "Distributed Optimization of Visual Sensor Networks for Coverage of a Large-Scale 3-D Scene". In: *IEEE/ASME Transactions on Mechatronics* 25.6 (2020), pp. 2777–2788. DOI: 10.1109/TMECH.2020.2993573.

[28] Geraint H Jones et al. "The Comet Interceptor mission". en. In: *Space Sci. Rev.* 220.1 (Jan. 2024), p. 9.

[29] Rico Jonschkowski and Oliver Brock. "Towards Combining Robotic Algorithms and Machine Learning: End-To-End Learnable Histogram Filters". In: (). URL: https://www.static.tu.berlin/fileadmin/www/10002220/Publications/Jonschkowski-16-IROS_WS.pdf.

[30] Richard M. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*. Ed. by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Boston, MA: Springer US, 1972, pp. 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9. URL: https://doi.org/10.1007/978-1-4684-2001-2_9.

[31] Julien Kritter et al. "On the optimal placement of cameras for surveillance and the underlying set cover problem". In: *Applied Soft Computing* 74 (2019), pp. 133–153. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2018.10.025. URL: https://www.sciencedirect.com/science/article/pii/S1568494618305829.

[32] Phone Thiha Kyaw et al. "Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem". In: *IEEE Access* 8 (2020), pp. 225945–225956. DOI: 10.1109/ACCESS.2020.3045027.

[33] Phone Thiha Kyaw et al. "Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem". In: *IEEE Access* 8 (2020), pp. 225945–225956. DOI: 10.1109/ACCESS.2020.3045027.

[34] Laëtitia Matignon and Olivier Simonin. "Multi-Robot Simultaneous Coverage and Mapping of Complex Scene - Comparison of Different Strategies". In: *AAMAS 2018 - 17th International Conference on Autonomous Agents and Multiagent Systems - Robotics Track*. Ed. by M. Dastani et al. Stockholm, Sweden: ACM, July 2018, pp. 559–567. URL: https://hal.science/hal-01726120.

[35] Alan T. Murray et al. "Coverage optimization to support security monitoring". In: *Computers, Environment and Urban Systems* 31.2 (2007), pp. 133–147. ISSN: 0198-9715. DOI: https://doi.org/10.1016/j.compenvurbsys.2006.06.002. URL: https://www.sciencedirect.com/science/article/pii/S0198971506000573.

[36] Kourosh Naderi, Joose Rajamäki, and Perttu Hämäläinen. "RT-RRT*: a real-time path planning algorithm based on RRT*". In: *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*. MIG '15. Paris, France: Association for Computing Machinery, 2015, pp. 113–118. ISBN: 9781450339919. DOI: 10.1145/2822013.2822036. URL: https://doi.org/10.1145/2822013.2822036.

[37] NASA. *Deep Impact Mission, NASA*. URL: https://www.jpl.nasa.gov/missions/deep-impact.

[38] JOSEPH O'ROURKE. "Art Gallery Theorems and Algorithms". en. In: (1987).

[39] OpenCV. *Morphological Transformations*. URL: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html.

[40] Michael Otte and Emilio Frazzoli. "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning". In: *The International Journal of Robotics Research* 35.7 (2016), pp. 797–822. DOI: 10.1177/0278364915594679. eprint: https://doi.org/10.1177/0278364915594679. URL: https://doi.org/10.1177/0278364915594679.

[41] Aditya Savio Paul. *volumetric sampling for event cognizance*. 2024.

[42] Aditya Savio Paul and Michael Otte. "Simultaneous Motion Replanning and Gravity Model Refinement near Small Solar System Bodies". In: *AIAA Journal* 20 (Sept. 2023). DOI: 10.2514/1.I011200.

[43] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing*. Boston, MA, 2008.

[44] Ragesh K Ramachandran, Zahi Kakish, and Spring Berman. "Information correlated Levy walk exploration and distributed mapping using a swarm of robots". In: (Mar. 2019). arXiv: 1903.04836 [cs.RO].

[45] Andrew Rivkin and Andrew Cheng. "Planetary defense with the Double Asteroid Redirection Test (DART) mission and prospects". In: *Nature communications* 14 (Mar. 2023), p. 1003. DOI: 10.1038/s41467-022-35561-2.

[46] David Salomon. *The computer graphics manual. 2 Volumes*. Jan. 2011. ISBN: 978-0-85729-885-0. DOI: 10.1007/978-0-85729-886-7.

[47] Frank Y. Shih. "Image Segmentation". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 1389–1395. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_1011. URL: https://doi.org/10.1007/978-0-387-39940-9_1011.

[48] *Tartu Observatory Space Mission Simulation Center*. URL: https://kosmos.ut.ee/en/space-bunker.

[49] "Towards endogenous mapping of small solar system bodies during multi-agent rendezvous." In: 2021. URL: https://iafastro.directory/iac/paper/id/65229/ext/appendix/IAC-21,A3,IP,57,x65229.pdf.

[50] Nicholas Mario Wardhana, Henry Johan, and Hock Seah. "Enhanced waypoint graph for surface and volumetric path planning in virtual worlds". In: *The Visual Computer* 29 (Oct. 2013). DOI: 10.1007/s00371-013-0837-x.

[51] Chinmay Wyawahare. *Object Detection Techniques in Computer Vision*. 2020. URL: https://medium.com/swlh/object-detection-techniques-in-computer-vision-7c169771fb15.

[52] A. V. Zakharov et al. "Physical processes leading to surface erosion and dust particles dynamics of airless bodies". In: *Physics of Plasmas* 29.11 (Nov. 2022), p. 110501. ISSN: 1070-664X. DOI: 10.1063/5.0117833. eprint: https://pubs.aip.org/aip/pop/article-pdf/doi/10.1063/5.0117833/16625264/110501_1_online.pdf. URL: https://doi.org/10.1063/5.0117833.

# Appendices

**Comparison between Campaign-01 and Campaign-02**

For comparing the results obtained from Campaign-01 and Campaign-02, a density plot for normalized differences between the simulation and blender environment scores, with respect to each campaign, were plotted (Figure 9.1). It is apparent from the graph that the distribution of the normalized score differences is more dense around the low values, for Campaign-02. On the other hand, for the Campaign-01 the distribution has two peaks with higher difference in the scores between the environments. This indicates that the Campaign-02 produced more consistent results with smaller variations between simulation and physical environment scores.

The box plot (Figure 9.2) was used for comparing the mean differences between the blender and bunker scores for both techniques. The plot provides overall measure of how much the scores differ on average between the two environments for each technique.The mean difference between the simulation and physical environment, when the coordinates for camera placement were generated by Campaign-01 is 0.16 and for Campaign-02, 0.02.The lower mean difference depicts that Campaign-02 generates configuration which yields scores that are more consistent through both simulation and physical environment.

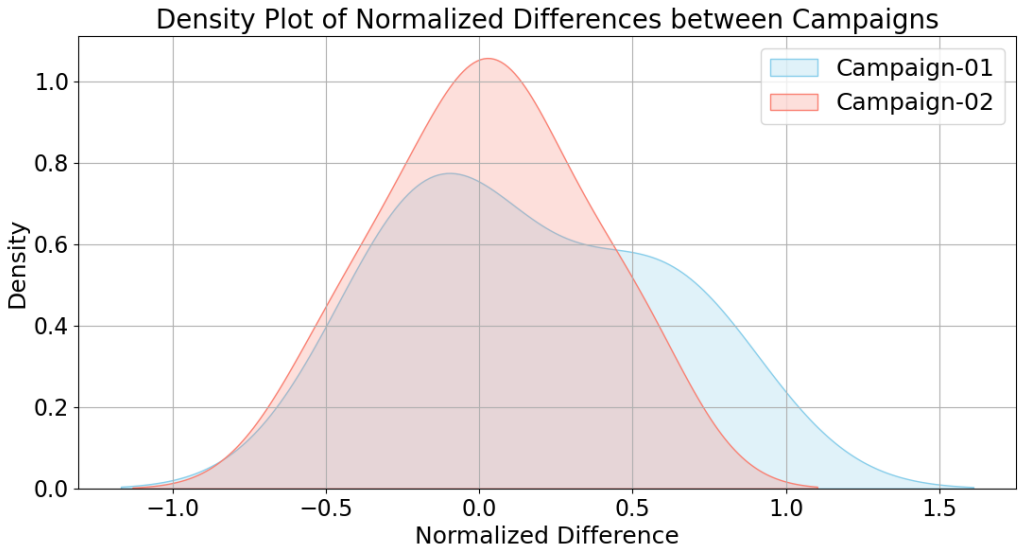



Figure 9.1: KDE Plot for normalized difference of scores between simulation and physical environment for both Campaign and Campaign2.

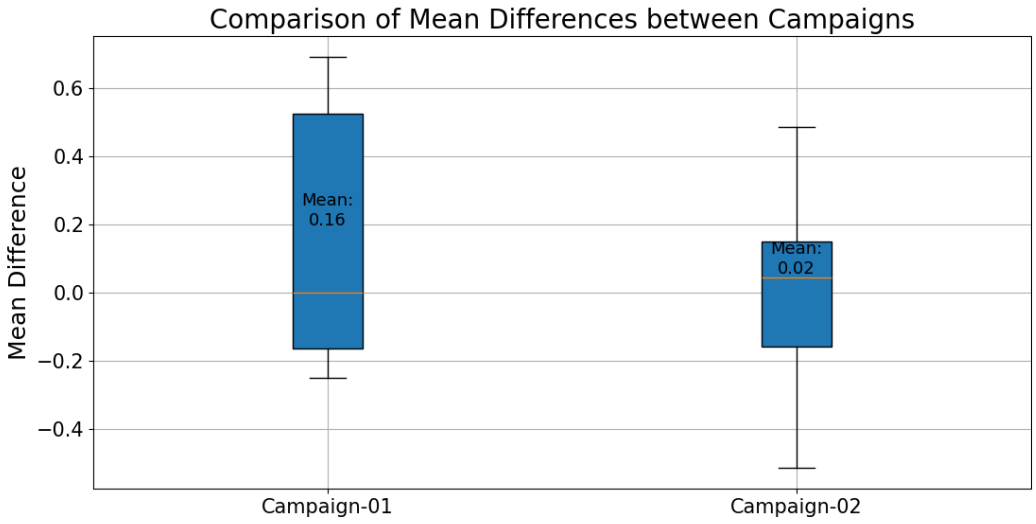



Figure 9.2: Mean difference plot, illustrating the mean difference between the Simulation and physical environment scores for both Campaign-01 and Campaign-02.

# Non-exclusive licence to reproduce thesis and make thesis public

I, Naghma Afreen

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   **"Optimal agent positioning for dynamic event monitoring and analysis."**

   supervised by Aditya Savio Paul

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

*Naghma Afreen*
**20.05.2024**