Universität
Augsburg
University

Bachelor's Thesis

Improving Crowdsourcing-based QoE Studies for Point Clouds via Interaction Incentives

Andreas Wilms

First Reviewer:	Prof. Dr. Michael Seufert
Second Reviewer:	Prof. Dr. Elisabeth Andre
Supervisor:	Prof. Dr. Michael Seufert, Filip Simonovski M. Sc.
Date:	September 30, 2024

published on
Chair of Networked Embedded Systems and Communication Systems
Prof. Dr. Michael Seufert
Faculty of Applied Computer Science
Augsburg University
D–86135 Augsburg, Germany
`https://www.uni-augsburg.de/informatik`

Contents

1. Introduction

Point clouds serve as the digital backbone of many technologies, such as autonomous driving, applications in manufacturing, and virtual reality (VR). Especially within the field of VR, point clouds are significant as they provide precise data to generate highly realistic environments. This precise nature allows users to enjoy a truly immersive experience, opening up new opportunities. For instance, the manufacturing industry benefits from point clouds by creating accurate 3D representations of machinery and assembly lines. This capability enables trainees to engage in VR simulations, allowing them to practice operating equipment and performing maintenance tasks without the risks and costs associated with real machinery.

In the time frame between 2020 and 2023, the number of VR and AR users grew from 83.7 million to 110 million in the US [1]. Meanwhile, the global VR market was valued at 12 billion USD in 2022 and is expected to pass 22 billion USD by 2025 [2]. While VR is regarded as the most important innovation in the gaming sector [3, 4], many other industries are being disrupted by its widespread impact. It is expected that 38% of the healthcare sector will be influenced by VR, Manufacturing 21%, and Education 28%, where VR is already outperforming traditional methods, particularly in training and skills development [4]. Despite these disruptive advancements, 26% of users still report poor user experience, and 24% cite a lack of quality content as barriers for adoption [3].

Realistic VR applications require a significant amount of point cloud data, which usage is often limited by transmission bandwidth or end-user hardware processing capabilities. This situation leads researchers and developers to implement reduction strategies to reduce point cloud size. User evaluations of each strategy are essential, because technical parameters only partially predict user satisfaction. Executing those evaluations through crowdsourcing has high potential due to its scale, diversity and cost-effectiveness. However, the approach faces challenges in its methodology, particularly regarding how to make point cloud evaluations feasible for crowdsourcing and ensuring data reliability.

A web-based crowdsourcing study on the Quality of Experience (QoE) of point clouds was conducted with participants recruited from the MicroWorkers[1] platform. In the study, users navigated freely within 3D point clouds and provided evaluations of their quality. The study logic was built upon the jsPsych[2] library, and the Potree[3] library made the point clouds viewable in the browser. An incentive mechanism in the form of an element, which can be a letter or a number placed within the clouds, is incorpo-

[1] https://www.microworkers.com/
[2] https://www.jspsych.org/
[3] https://github.com/potree/potree

rated to test its influence on user engagement. To evaluate this, various configurations related to the incentive mechanism, such as element sizing, quantity, and positioning, are implemented. Additionally, two user-instruction scenarios are investigated for their influence. All of this was tested on two stimuli, each reduced by two different reduction strategies, across three levels of degradation.

This thesis investigates the feasibility of using crowdsourcing-based studies to evaluate the QoE of point clouds. It also explores methods to improve the methodology of interactive crowdsourcing studies. Additionally, the thesis examines whether implementing an incentive mechanism, such as embedding letters or numbers for searching within point clouds, can enhance participant engagement. It seeks to determine the most effective way to implement this mechanism to improve participant interaction. Finally, the study investigates whether this implementation can enhance the QoE evaluation in crowdsourcing studies involving point clouds.

Beginning with a deep dive into the theoretical background of the topic and review of related work in Chapter 2, the thesis then outlines the methodology and design of the study in Chapter 3. The results of the study are presented in Chapter 4, followed by a conclusion and outlook in Chapter 5.

2. Background and Related Work

First in section 2.1, the key concepts and technologies that form the foundation for this research are presented. It covers point clouds, QoE, crowdsourcing, and the MicroWorkers platform, as well as the jsPsych library for web-based behavioral experiments. Subsequently, in section 2.2, existing research related to crowdsourcing-based QoE studies for point clouds is reviewed, exploring the current state of the art in this field.

2.1. Background

In this section, the fundamental concepts and technologies that underpin this research are presented. The discussion begins with an overview of point clouds 2.1.1, followed by an exploration of QoE 2.1.2, and an examination of crowdsourcing 2.1.3. The section concludes with an overview of the MicroWorkers platform 2.1.4 and the jsPsych library for web-based behavioral experiments 2.1.5.

2.1.1. Point Clouds

In the realm of 3D modeling and spatial data representation, point clouds play a crucial role. They offer detailed and accurate digital replicas of physical objects or spaces, capturing minute details that are essential for various applications [5].
A point cloud is a set $S = \{p1, p2, ..., pn\}$ of points that collectively describe the shape and surface of a 3D object or scene. Each point p typically contains at least three coordinates (x, y, z) defining its position in 3D space, such that $S = \{p1, p2, ..., pn | p \in R^3\}$. Additional properties like curvature or color information (r, g, b) may be included, extending the dimensionality of each point's representation.[6]

Acquisition of Point Cloud

There are several methods for acquiring point clouds, each with its own advantages and limitations. The most common methods include:

- **Light Detection and Ranging (LiDAR):** Widely used in real-time applications, LiDAR operates by emitting laser pulses and measuring the time it takes for the light to reflect off objects and return to the sensor. This "time of flight" data is used to calculate precise distances. The resulting collection of measurements forms a point cloud that represents the scanned area in three dimensions.[6]

- **Structure from motion (SfM):** Often used interchangeably with the term photogrammetry, SfM is a technique that utilizes a series of 2D images taken from different angles to reconstruct a point cloud, as illustrated in Figure 2.1.[6]

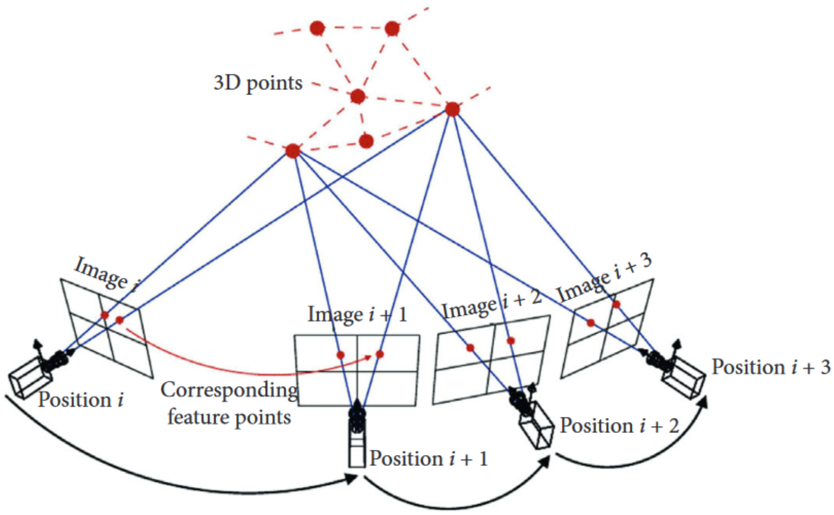

Figure 2.1.: Structure from motion [6]

Applications of Point Clouds

Point clouds build the digital foundations for numerous industries by providing precise and detailed 3D models. These digital representations are particularly valued for their accuracy and versatility. In agriculture, point clouds enable efficient crop monitoring and management. The aerospace and defense sectors utilize them for detailed simulations and analyses. In construction and architecture, point clouds assist in planning, visualization, and ensuring structural accuracy. The entertainment industry, especially in VR and augmented reality (AR), relies on point clouds for creating immersive experiences. Additionally, the automotive industry uses point clouds to develop high-definition maps and enhance autonomous driving by enabling real-time perception and accurate object detection and classification.[5]

Challenges in Point Cloud Processing

Due to the limitations of three-dimensional scanning technology, multiple point clouds are typically captured from different viewpoints, each with its own coordinate system. The overlap region between consecutively retrieved point clouds can vary in size, from

large to small. Most existing solutions calculate the possible transformations by determining correspondences based on these common regions. Therefore, processing varying overlap during point cloud registration remains a significant challenge.[6]
Capturing devices may also introduce noise, missing data and variations in accuracy, especially with low cost sensors. This circumstances demand filtering and registration algorithms in order to achieve a high-quality point cloud model.[6]
Subsequently, the sheer volume of data in point clouds poses a challenge for storage, transmission and processing. The large size of point clouds requires efficient compression techniques to reduce the data size while maintaining the quality of the model.[7]
The combination of filtering, registration and compression algorithms leads to a high computational cost, which can be a bottleneck for real-time applications.

3DTK and CloudCompare: Point Cloud Processing Software

After capturing the point clouds, they are processed using specialized software such as the 3D Toolkit[1] (3DTK) and CloudCompare[2].
3DTK excels in advanced processing tasks, particularly in the registration and alignment of complex, large-scale point cloud datasets. It handles the entire workflow from importing raw data, preprocessing, and aligning multiple scans (registration) to exporting the processed data using various compression techniques.[8]
In Addition, CloudCompare is a versatile open-source tool designed for both processing and analyzing point cloud data. It provides a comprehensive suite of features for importing, visualizing, editing, and exporting point clouds, making it an essential resource and a valuable addition to 3DTK for working with 3D data.[9]

2.1.2. Quality of Experience (QoE)

As services and systems become more functionally sophisticated, evaluating their quality has grown increasingly complex due to the expanding number of dimensions involved. Some applications involve a few technologies and sensory dimensions, while others combine numerous technologies and experiences, making quality assessment more challenging. This subsection explores QoE multifaceted nature and the complexities of ensuring high-quality experiences in modern systems.[10]

Definition and Scope

QoE is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current

[1] https://www.slam6d.sourceforge.io
[2] https://www.cloudcompare.org

state.[10]
QoE is a highly subjective concept that extends beyond the traditional metrics of quality of service (QoS). QoS is primarily concerned with the technical performance of telecommunication services, networks, and applications. It measures specific technical parameters such as bandwidth, latency, and packet loss to assess the system's efficiency and reliability. However, while QoS focuses on these objective technical metrics, QoE encompasses the user's perception and overall satisfaction with the service. QoE considers a broader range of factors, including usability, accessibility, and emotional responses to the service, providing a more comprehensive understanding of the user experience. This broader scope makes QoE a more inclusive and holistic measure compared to QoS.[10]

Influence Factors

QoE is influenced by several interrelated factors categorized into Human, System, and Contextual influencing factors.[10]
Human factors encompass user characteristics that significantly influence their perception and experience. These factors include demographic and socio-economic background, physical and mental constitution, and emotional state. System factors dictate the technical quality produced by an application or service. These factors encompass content-related factors, media-related factors, network-related factors, and device-related factors. Contextual factors describe the situational aspects of the user's environment, such as physical context, temporal context, economic context, task context, social context, and technical and information context.[10]

2.1.3. Crowdsourcing

Crowdsourcing is a distributed problem-solving and production model that involves outsourcing tasks to a distributed group of people or community (a crowd) through an open call. Crowdsourcing leverages the collective intelligence, expertise, and skills of the crowd to solve complex problems, generate ideas, or complete tasks that would be difficult or expensive for a single entity to accomplish.[11]
The concept of crowdsourcing is not new, but the advent of the internet and social media has significantly expanded its scope and potential. Famous examples would be the online encyclopedia wikipedia[3] which acquires collective knowledge from volunteers worldwide, or the creation of the Toyota Logo by a crowd of designers.[11]
In web-based crowdsourcing, tasks are typically posted on online platforms where individuals can sign up to complete them. These platforms provide a marketplace for requesters to post tasks and for workers to select and complete them. After completing a task, workers are compensated based on the conditions set by the requester.[11]

[3] https://www.wikipedia.org

A Framework for Crowdsourcing

The crowdsourcing process can be divided into the stages Initializing, Implementation and Finalization as illustrated in Figure 2.2.
The initialization phase of crowdsourcing contains important preparation steps before tasks are assigned to the crowd. In the beginning, the requester must design the task effectively and estimate the required work. In cases of complexity, a divide-and-conquer approach may be employed to break tasks into manageable subtasks. It is essential to ensure that each task's execution does not interfere with others. Furthermore, the requester must design incentives that attract crowdsourcing workers to sign up for the job based on the offered rewards.[11]
Once all results from the crowd are collected, it is essential to refine them by distinguishing answers from cheaters using various quality control methods. In the final step, once all results from the crowd are collected, it is time to clean the data by eliminating cheaters using various quality control methods. Once the preparation steps are complete, the requester seeks a suitable crowd for the task at hand. Various crowdsourcing platforms, such as Amazon Mechanical Turk[4] or MicroWorkers[5], can be utilized for this purpose. By carefully selecting a fitting crowd, quality can be ensured. After this process, the answers are aggregated and analyzed which concludes to the final output of the crowdsourcing.[11]

Figure 2.2.: Framework of crowdsourcing [11]

[4]https://www.mturk.com/
[5]https://www.microworkers.com

Incentives in Crowdsourcing

Incentives are the driving force behind crowdsourcing, motivating participants to contribute their time, effort, and expertise to complete tasks. They can be split into monetary and non-monetary categories, with monetary incentives being the most common and preferred form of reward.[11]

Monetary Incentive Model

Research shows that work quality in crowdsourcing is linked to payment. However, higher payments can encourage speed over quality, as workers may prioritize earnings. Underpayment can cause "task starvation." Additionally, paying only after task completion can lead to exploitation without guaranteed compensation. On the other hand, guaranteed payment may reduce motivation due to free-riding. Therefore, requesters must balance compensation to ensure fair pay and high-quality work.[11]

Non Monetary Incentive Model

Monetary incentives are not always the most effective motivators. Non-monetary incentives can be equally, if not more, effective in motivating crowdsourcing workers, sometimes leading to higher quality work due to the intrinsic motivation they offer.[11] Here are two examples of non-monetary incentives:

1. **Natural Incentives:** Making tasks more enjoyable through gamification, competition, or opportunities to learn new skills can be a powerful motivator. A successful example is Duolingo's[6] language learning platform. Users learn new languages by translating texts and earn points and badges for their progress. This not only enhances their language skills but also contributes to the platform's development and the translation of texts.[11]

2. **Solidary and Moral Incentives:** Motivated by a sense of helping others during times of crisis or by the desire to gain reputation within a community for better future opportunities, workers may be more willing to contribute their time and effort to crowdsourcing tasks.[11]

2.1.4. MicroWorkers

MicroWorkers[7] is a global crowdsourcing platform that connects employers and workers for small, task-based work, commonly referred to as microtasks. Founded in 2009 and based in Dallas, Texas, the platform has experienced significant growth, offering

[6] https://www.duolingo.com
[7] https://www.microworkers.com

businesses and individuals ways to outsource repetitive or time-consuming tasks to a distributed workforce.[12]
These tasks typically include data entry, content moderation, web research, survey participation, and app testing. MicroWorkers hosts a wide array of tasks, allowing employers from various industries to find suitable workers for their specific needs, while workers can choose tasks that match their skills and preferences. The platform operates worldwide, giving employers access to a global talent pool, ensuring tasks can be completed around the clock and often at competitive rates.[12]
Quality control is maintained through a rating and feedback system, where workers receive ratings based on their performance and employers can review and approve work before payment. This helps maintain high standards and reliability.[12]

2.1.5. jsPsych: A Library for Web-Based Behavioral Experiments

jsPsych[8] is an open-source JavaScript library specifically designed to facilitate the creation and administration of behavioral experiments within a web browser. Tailored for researchers, jsPsych eases the process of programming and deploying online experiments, thereby reaching a broader and more diverse participant pool.[13]
The library provides an extensive range of plugins for various experimental needs, including stimulus presentation, response collection, and data recording. These plugins are well-documented and tested, ensuring a robust foundation for different experimental paradigms. Additionally, jsPsych allows for significant flexibility and customization, enabling researchers to create a wide variety of experiments, from simple reaction time tasks to complex multi-stage designs. Researchers can also develop their own plugins to further extend the library's functionality.[13]
Experiments in jsPsych are structured around a timeline, which is a sequence of events that occur during the study. Each event is represented by a trial, encompassing stimulus presentations, response collections, and data recording events. This timeline structure is highly flexible, allowing researchers to precisely define the order and timing of events in their experiments.[13]
Researchers can build experiments using a combination of HTML, CSS, and JavaScript, and these experiments can be executed on any device with a web browser, including desktops, laptops, tablets, and smartphones. Furthermore, jsPsych provides useful tools for data collection and analysis, making it an viable solution for conducting web-based behavioral experiments, particularly well-suited for crowdsourcing.[13]

2.2. Related Work

This section reviews existing research related to crowdsourcing-based QoE studies for point clouds and explores the current state of the art in this field. It covers studies on

[8]https://www.jspsych.org/

crowdsourcing, QoE evaluation, and point cloud processing, highlighting the challenges and opportunities in this domain.
The subjective evaluation of compression techniques for point clouds is essential for optimizing visual quality in applications that require data reduction. Seufert et al. [14] conducted an assessment of subjective quality in rendered images from point clouds, comparing two compression methods, octree-based and projection-based, across three levels of degradation. Their study used four diverse test stimuli, named "Chapel," "Church," "Human," and "Text," to obtain a range of different content types. By employing a crowdsourcing approach, the researchers collected user evaluations of the rendered images through the browser. The findings consistently demonstrated a user preference for the projection-based compression method over the octree-based approach, despite both techniques achieving similar compression ratios. This thesis takes a step in a different direction by evaluating point clouds directly in their native format, giving participants the opportunity to freely navigate the point clouds within the browser. Seufert et al. proved the feasibility using crowdsourcing as an viable assessment method for evaluating images. On the other hand, this thesis explores the feasibility of using such an approach for its case and searches for ways to improve its method and participants engagement with point clouds by incorporating an incentive mechanism.
In line with previous research, Alexiou et al. [15] examine how different compression techniques affect the subjective human perception of point clouds. Their study involved evaluating five point clouds at four levels of degradation using the methods of Gaussian noise and octree-pruning. They found that while existing objective metrics are reasonably effective at assessing Gaussian noise distortions, they often fall short in accurately predicting the perceptual quality of compression-related artifacts. Additionally, the effectiveness of these metrics in evaluating octree-pruning distortions was found to be content-dependent. Next, Zhang et al. [16] introduce a subjective quality evaluation model designed to assess differences between original and processed 3D color models. Their study, which involved evaluating point clouds under various degradations, found that changes in resolution correlated almost linearly with human perception, while color alterations exhibited a non-linear relationship. Notably, shape modifications were found to have a more significant impact on perceived quality than color changes, even when similar noise levels were applied. Unlike this thesis, the evaluations in the studies by Alexiou et al. and Zhang et al. were laboratory-based, and participants did not navigate freely in the first person through the point clouds. By exploring a different evaluation approach, this thesis aims to contribute to improved quality assessments of point clouds, helping future studies leverage the advantages of crowdsourcing, such as scale and diversity.
Another important aspect of point cloud applications is the streaming of 3D content, which necessitates efficient compression techniques to balance data size and visual quality. Van der Hooft et al. [17] address this by using the V-PCC coder to encode point clouds at five different rates, then computing objective image and video quality metrics by comparing video sequences from both the original and degraded point cloud data. Their findings demonstrated a strong correlation between some objective quality measures and subjective assessments. Similarly, Dumic et al. [18] focused on six point clouds,

applying G-PCC and V-PCC compression techniques at five different degradation levels. Their study revealed strong inter-laboratory correlation in subjective evaluations, illustrating the consistency of human perception across different viewing conditions.
The work of Wang et al. [19] sheds a light on another aspect of maximizing QoE of point cloud experiences. Wang et al. propose a QoE-driven point cloud streaming system that utilizes tile-based transmission to maximize the user's QoE while reducing transmission redundancy. To optimize the rate adaptation process, Wang et al. framed it as a multiple-choice knapsack problem, which they then converted into a submodular function maximization. This transformation allowed them to apply a greedy algorithm, which provided theoretical performance guarantees. Their empirical evaluations demonstrated that this novel rate adaptation algorithm outperforms existing methods, leading to enhanced visual quality and improved transmission efficiency. In a related work, Han et al. [20] also built a system based on perspective projection and managed to achieve an average data usage reduction of 40% (up to 80%) while maintaining high QoE.
All the previously mentioned studies underscore the significance of effective reduction strategies and quality assessment in various domains. They complement this thesis in the pursuit of finding the balance between QoE and the reduction of point clouds. Except for [14], they present a research gap in the evaluation approach, which shows a strong dependence on laboratory-based evaluations. Many studies rely primarily on professional or academic participants, such as those affiliated with universities [21, 16, 17, 15], which may not represent the general population. Some studies do use demographically balanced samples [18, 20], but this approach is not universal. Additionally, the number of participants in each conducted study is in the low double digits. This underscores the need for more diverse and inclusive testing groups to ensure the validity of research findings and achieve larger scales.
To the best of my knowledge, no study, except for [14], has employed crowdsourcing to evaluate point cloud quality. Additionally, no thesis has explored the feasibility of crowdsourcing-based evaluations of QoE in point cloud studies, where participants can engage directly by navigating freely within the point clouds. This underscores the significant potential for further exploration in this area.

3. Methodology

The methodology chapter describes the study design, stimuli and technical execution of achieving an interactive crowdsourcing study for the QoE evaluation of point clouds. First, it describes the stimuli used in Section 3.1. Secondly, the incentive method is presented in Section 3.2, followed by the sequential comparison method in Section 3.3. Next, the study design is presented in Section 3.4. Lastly, the technical implementation of the study is detailed in Section 3.5, covering the use of the jsPsych library and Potree for data collection and point cloud visualization, respectively.

3.1. Stimuli

All the point clouds described below are the same as those used by Seufert et al. in [14]. This study utilizes stimuli from the 3D scan repository [22], specifically the models *22.bremen_city*, presented from a top-down view as depicted in Figure 3.1, and *26.randersacker*, shown in Figure 3.2. These models are also referred to as Bremen and Randersacker, respectively, throughout the thesis.
In addition to these two original point clouds, six reduced versions of each include three octree reductions and three projection-based panorama reductions are used. The octree reductions were achieved by applying different voxel sizes and random subsampling of points within each voxel. The voxel sizes were 4, 8, and 30 cm, corresponding to the reductions OCV4, OCV8, and OCV30, respectively. Additionally, the three projection-based panorama reductions were employed to create datasets approximately equivalent in size to the octree-reduced datasets. The reduction names R3600x1000, R2400x667, and R1200x333 refer to panorama sizes of 3600×1000, 2400×667, and 1200×333 pixels, respectively. In total, this results in 14 point clouds, providing a comprehensive set of data for analysis. A comparison is shown in Table 3.1.
The movement area of each point cloud is reduced to contain only its main features. This area is indicated by the red circle, while the red cross marks the users starting position, as shown in Figures 3.1 and 3.2.

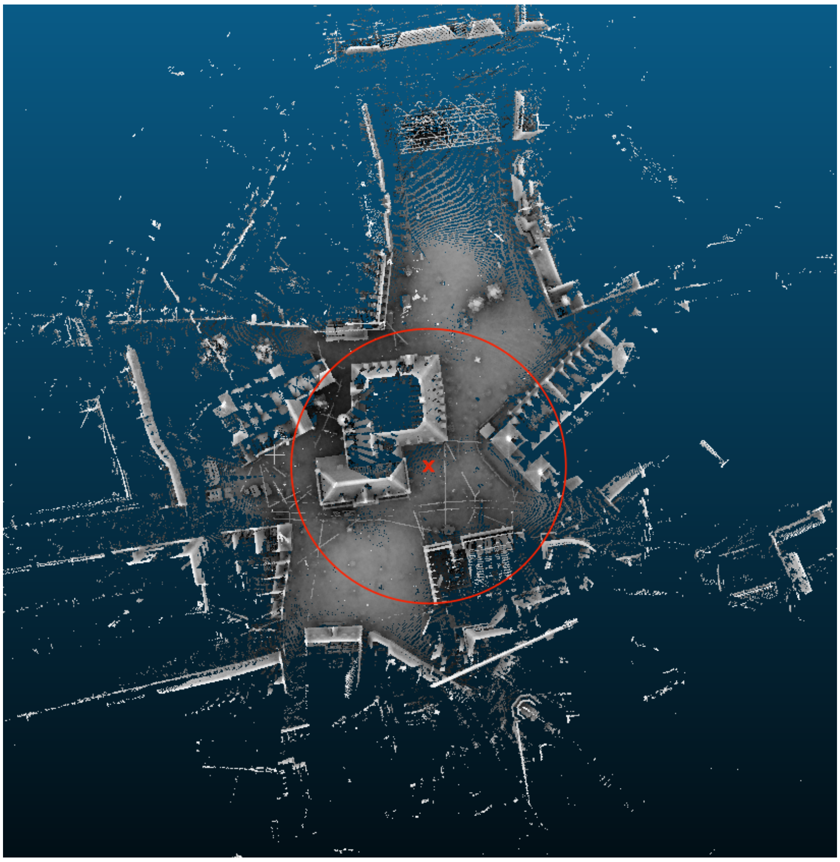

Figure 3.1.: Top view of 22.bremen_city with navigation limits

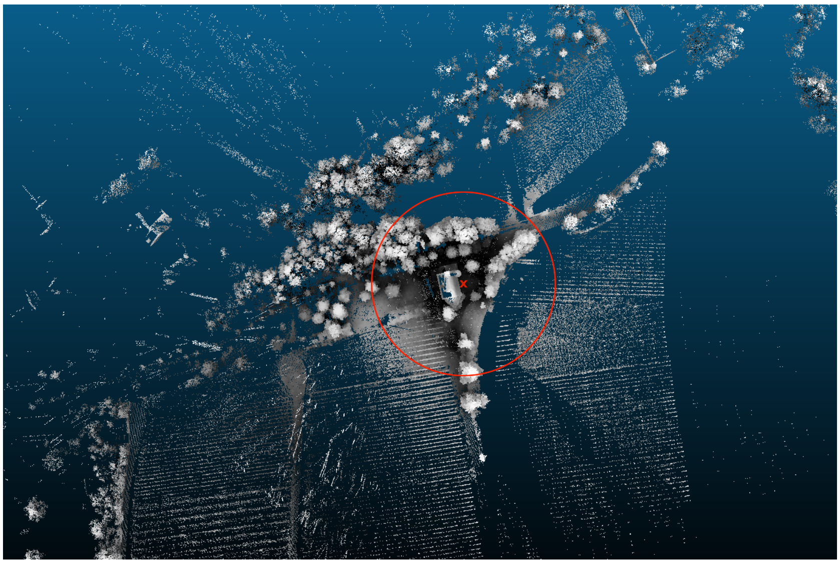

Figure 3.2.: Top view of 26.randersacker with navigation limits

Table 3.1.: Comparison of point clouds and reductions [14]

	bremen_city		randersacker	
Reduction	Data Size	Points	Data Size	Points
Original	10.346 MB	215.652.400	11.375 MB	194.754.633
OCV4	1.587 MB	35.985.142	769 MB	16.075.352
OCV8	657 MB	14.913.197	345 MB	7.196.991
OCV30	93 MB	2.113.011	89 MB	1.851.189
R3600x1000	1.629 MB	34.228.877	1.383 MB	29.008.210
R2400x667	727 MB	15.288.306	619 MB	12.983.283
R1200x333	183 MB	3.846.119	156 MB	3.275.927

3.2. Incentive Method

In order to enhance participant engagement and improve the quality of feedback, this study integrates a validation mechanism. A blue 3D element, in the form of either a letter or a number, is placed within the point clouds (see Figure 3.3). Participants must locate and identify it during the evaluation to confirm their participation. The study also tests various parameters, such as the size, placement, and quantity of elements, to determine the optimal configuration for maximizing participant engagement and evaluation quality. For Randersacker, six locations near the chapel were chosen, and for Bremen, eight locations were selected, evenly distributed. The exact coordinates and other parameters, such as sizes, are described in detail in the Technical Implementation section 3.5.

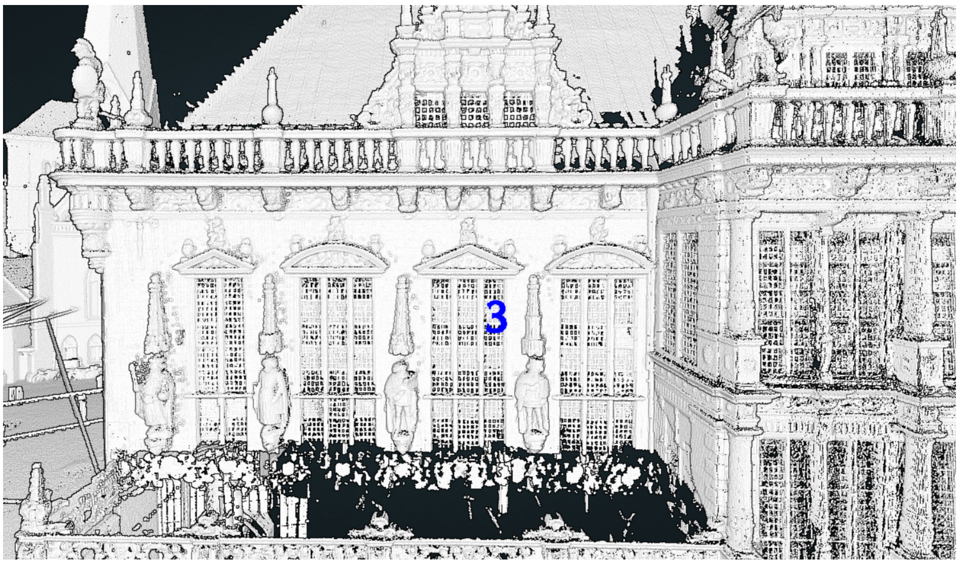

Figure 3.3.: Element placement in the point cloud

3.3. Sequential Comparison

The sequential comparison method is the evaluation mechanism used to assess the point clouds. It follows the sequence displayed in figure 3.4.
Initially, participants view an original point cloud from either Randersacker or Bremen, with elements placed in it based on the tested incentive condition. This is displayed along with a proceed button and a 10-second countdown. The button activates after the countdown, allowing participants to move to the next stage.
Participants then encounter a rating page where they assess the quality of the original point cloud using a 5-point Absolute Category Rating (ACR) scale, ranging from "bad" to "excellent", and validate the element.
Next, participants view a reduced version of the previous point cloud. The incentive mechanism is also incorporated here as before. A 10-second viewing period is provided before they can advance.
Finally, on the last rating page, participants evaluate the difference between the original and reduced version using a 5-point ordinal scale, from "no difference" to "huge difference". They also assess the quality of the reduced point cloud using the 5-point ACR scale and validate the element.

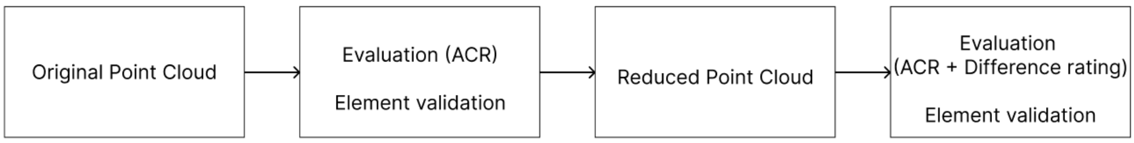

Figure 3.4.: Sequential comparison process

3.4. Study Design

This section brings together all the components to create the final design used in this study. The sequence is shown in Figure 3.5.
Initially, participants receive instructions on how to conduct themselves during the study, which include avoiding actions such as reloading the page, zooming, or switching tabs. Then they are introduced to the study's topic and given a general background about point clouds, VR and their importance. This is followed by information and instructions about the task they will perform. In the task, users will be required to traverse four distinct evaluation blocks following the sequential comparison method (see 3.3). Each block contains point clouds from either Randersacker or Bremen. Two blocks feature Randersacker, and two blocks feature Bremen. The order is randomized, either presenting two Randersacker blocks followed by two Bremen blocks, or vice versa.
After the information and instructions phase, participants have to complete a trial run

to familiarize themselves with the interface. The trial run features a simplified version of the study task, allowing participants to practice navigating the 3D environment and complete a test assignment. To ensure participants fully grasp how to use the interface, the trial is structured as an infinite loop. Participants must successfully complete the task before advancing, thereby demonstrating their proficiency with the interface.
Following the trial run, the main study begins, with participants advancing through the four evaluation blocks.
Upon completing the study, participants are invited to provide feedback and comments about their experience. Finally, participants receive a unique code for payment redemption via the MicroWorkers[1] platform.

Figure 3.5.: Abstract study design

3.5. Technical Implementation

For the technical implementation of the study, the jsPsych[2] library was used to manage the study flow and data collection. The point clouds were formatted using 3DTK[3] and visualized with Potree[4], a web-based point cloud viewer. The study was hosted on Augsburg University's server, which features an SSL certificate to ensure secure data transmission. The backend code was developed in PHP, and the study was hosted on an Apache server. Docker containers were utilized to manage the Apache server and handle data volume mounting efficiently.
Participants were recruited through the MicroWorkers platform and provided with a code to redeem their payment upon completion of the study.

3.5.1. Study Design with jsPsych

The study design described in Section 3.4 is implemented using jsPsych. To facilitate this, a *survey.html* file initializes the jsPsych library, along with the necessary CSS styles and essential plugins. The core logic and flow of the study are managed by the *survey.js*

[1] https://www.microworkers.com/
[2] https://www.jspsych.org/
[3] https://slam6d.sourceforge.io/
[4] https://github.com/potree/potree

file, which is loaded by *survey.html*. Leveraging the jsPsych library, the timelines discussed in Section 3.4 are constructed by organizing the blocks into a jsPsych timeline object, which is then executed sequentially.

Between the various tested incentive conditions the instructions get minor adjustments and are manually created to accommodate the current condition tested. On the other hand, the evaluation blocks are generated dynamically based on the conditions due to the vast variations in possible configurations. For instance, in the sequential comparison method, one original point cloud can be paired with six different reductions, resulting in six unique blocks. With two original point clouds, this expands to a total of twelve blocks. Testing three different element sizes results in 36 blocks, and adding two element quantity tests leads to a total of 60 blocks. Future test conditions can further increase the number of blocks.

To avoid hardcoding these blocks, a hierarchical structure was implemented, allowing for dynamic generation. This approach enhances flexibility and reusability, while reducing code duplication, complexity, and potential errors.

The hierarchical structure is organized into three levels of abstraction. At the highest level, functions are defined to generate the structure of the study blocks as a jsPsych timeline object. For example, in a sequential comparison, the function specifies the order of events, such as presenting the original point cloud, followed by a rating page, displaying the reduced point cloud, and concluding with a final rating page. Additionally, it specifies the conditions to be tested, such as using test stimulus "Bremen" with an element size of "*small*".

The middle tier of this hierarchy consists of an HTML page that is loaded by the function level. This page is responsible for displaying the point cloud window, not the point cloud itself, in a format suitable for the browser, as well as for managing the user interface. It also manages data collection for user interactions within the point cloud. In this study it builds the interface for the sequential comparison method.

At the lowest tier, an HTML page, loaded into the middle tier, handles the rendering of point clouds and configures specific stimuli settings. This includes adjustments for movement restrictions, starting positions, element sizing, element quantity and movement speed. Each stimulus has its own page with the appropriate calibration of parameters.

For example, if we want to add a new stimulus and a new evaluation method, the only steps required are to create a new stimulus page with the appropriate parameter calibration and a new mid-tier page with the updated interface. At the top tier, we can now call them to dynamically create block variations.

Once the study blocks are generated, they are added to the main timeline object, which is then executed sequentially. This method allows for easy modifications to the study design, as changes can be made at the function level without affecting the underlying structure, ensuring consistency and adherence to the predefined design pattern.

3.5.2. Point Cloud Visualization with Potree

In order to make a web-based crowdsourcing study feasible, the point clouds needed to be visualized in a 3D environment for the browser. Potree[5], a web-based point cloud viewer, was chosen for this purpose. Potree is an open-source project that provides a framework for visualizing point clouds in a web browser. It is based on the WebGL library Three.js[6] and offers a variety of features for rendering and interacting with point clouds.

Preparing Point Clouds for Potree

To utilize the point clouds in Potree, they first needed to be converted into a compatible format using Potree Converter 1.6[7]. This converter requires point clouds to be in the `.pts` format. Since the original point clouds are in the `.3d` format, which is not supported, the 3DTK[8] toolkit was employed to perform the conversion. The conversion process involved using the `exportPoints` tool from 3DTK to transform the point clouds into the `.pts` format. This was done using default settings to ensure that the integrity of the point cloud data was maintained throughout the conversion process. Following the conversion, the point clouds were imported into Potree Converter to generate the necessary files for use with the Potree Viewer.

Configuring Potree

To optimize performance and usability, Potree was configured with tailored settings, as detailed in table 3.2. These configurations ensure a consistent visual experience across all point clouds. In addition to the shared visual settings applied across all point clouds, each point cloud has its own specific configurations, including starting position, movement restrictions, movement speed, and element sizing. To achieve the design choices mentioned in section 3.4 and illustrated in Figures 3.1 and 3.2, the following settings in table 3.3 were used. Element placement was enabled by utilizing Potree's foundation on Three.js. First, the font *Optima Regular* is loaded using `THREE.FontLoader`. Once the font is loaded, a `THREE.TextGeometry` object is created for the specified text, configuring properties such as size and height. A `THREE.MeshBasicMaterial` is then applied to the text, setting its color to blue. Finally, a `THREE.Mesh` combines the text geometry and material into a 3D text object, which is then added to the scene where Potree has already loaded the point cloud. In Randersacker, six positions were used for the elements, while in Bremen, eight positions were employed. These positions, along with their corresponding coordinates and labels, are shown in Table 3.4.

[5]https://github.com/potree/potree
[6]https://threejs.org/
[7]https://github.com/potree/PotreeConverter/releases/tag/1.6
[8]https://slam6d.sourceforge.io/

Table 3.2.: Viewer and material settings affecting visualization

Setting	Description
setFOV(90)	Sets the field of view to 90 degrees.
setPointBudget(1*1000*1000*1000)	Allows rendering of up to 1 billion points at once.
useHQ = true	Enables high-quality rendering for enhanced visuals.
material.size = 1	Sets the material size to 1.
Potree.PointSizeType.ADAPTIVE	Adjusts the point size based on distance and screen density.
setNavigationMode(Potree.FirstPersonControls)	Enables first-person navigation with A/W/D/S key for movement and the mouse for rotation.

Table 3.3.: Point cloud specific settings

Setting	Bremen	Randersacker
Starting Position (x, y, z)	(-22, 8, 4)	(3.5, -10.6, 1)
Movement Radius	70	35
Movement Speed	14	7
Element Size	0.9	0.3
Looking Direction (x, y, z)	(-90, 50, 0)	(0, -90, 0)

Table 3.4.: X, Y, Z positions for stimuli in Bremen and Randersacker

Position	Bremen (X, Y, Z)			Randersacker (X, Y, Z)		
pos1	13	13.5	17	-6	-7	0.6
pos2	31	50	2	-4.5	-18	4
pos3	-65	-7	9	-13	-33	-0.5
pos4	-40	-68	3	-6	-14.5	-0.5
pos5	-35	64	4	8	-0.4	1
pos6	-5	76	1	1.7	-8	7.3
pos7	-18	30	0			
pos8	-15	-27	0			

4. Evaluation

The evaluation chapter presents the results of the conducted experiments. The first section, 4.1, focuses on the validation mechanism by evaluating the influence of different incentive sizes in subsection 4.1.1 and influences of quantities in subsection 4.1.2. In the second section, 4.2, the impact of different wordings on the evaluation results is analyzed.
To ensure the quality of the evaluation, only the first submitted results of the workers were used, and if workers participated in multiple experiments, only their first participation was considered.

4.1. Mechanism

The validation mechanism detailed in the methodology chapter, specifically in section 3.2, was thoroughly tested across various scenarios, including different sizes and quantities. This comprehensive testing aimed to optimize exploration, engagement, evaluation reliability, and success rates. It assessed the mechanism's effectiveness in enhancing user interaction and ensuring accurate, consistent evaluations. By applying the mechanism under diverse conditions, the testing identified the most effective configurations and clarified its impact on user experience and result reliability.

4.1.1. Influence of Incentive Sizes

To assess the impact of varying incentive sizes, three distinct incentive sizes were employed across separate study runs. As shown in Table 3.3, the element sizes are 0.9 for Bremen and 0.3 for Randersacker. These sizes are selected based on the relative sizes of the inspected point clouds, with the Bremen point cloud being larger than the Randersacker point cloud, hence the larger size applied to Bremen. Furthermore, only one element is displayed per point cloud, as the primary focus is on the impact of incentive size. These sizes were categorized as *small*, *medium*, and *large*, with the *medium* size serving as the base size. The *small* size is defined as one-third of the medium size, while the *large* size is three times the medium size. For Bremen, the sizes are *small* = 0.3, *medium* = 0.9, and *large* = 2.7. Similarly, for Randersacker, the corresponding sizes are 0.1, 0.3, and 0.9 respectively for *small*, *medium*, and *large*. The study runs were conducted using the sequential comparison method described in section 3.3. A run was

considered successful if participants correctly identified at least five elements and did not navigate away from the survey tab more than four times.
Table 4.1 shows the relationship between incentive size and evaluation results across different study runs. The "Passed Tab Filter" column indicates the number of participants who demonstrated engagement by remaining on the survey tab. Those who stayed within the filter were compliant with the established participation rules. The numbers in the "Successful" column represent participants who successfully completed the study by correctly identifying at least five elements and adhering to the rules. The percentage in the "Success %" column represents the portion of users who completed the study successfully out of the participants who adhered to the rules. The "Avg. Time" column shows the average time spent by participants, measured in minutes.
For the *small* size, 26 participants were involved. Of these, 20 passed the tab filter, and 6 successfully completed the study. For the *medium* size, 37 participants participated, with 28 passing the tab filter and 15 successfully completing the study. Finally, for the *large* size, 35 participants were involved, with 30 passing the tab filter and 24 successfully completing the study. In percentage terms, the success rates were 30% for the *small* size, 54% for the *medium* size, and 80% for the *large* size. For the *medium* The average time spent for the *small* size is 8.9 minutes, for the *medium* size is 9.9 minutes, and for the *large* size is 10.2 minutes.

Table 4.1.: Incentive sizes and success rates

Incentive	Participants	Passed Tab Filter	Successful	Success %	Avg. Time
Small	26	20	6	30%	8.9 min
Medium	37	28	15	54%	9.9 min
Large	35	30	24	80%	10.2 min

The results indicate a linear increase in success rates with increasing incentive sizes, with the *large* size yielding the highest success rate. This trend can be attributed to the enhanced visibility of larger incentives, making them easier to locate. Furthermore, the time spent is similar across the different sizes, particularly between the *medium* and *large* sizes, suggesting that participants spend approximately the same amount of time, yet are able to locate elements more efficiently with larger incentives.
To evaluate whether there is a dependency on the stimulus in the success rates across different sizes, the success rates for each point cloud and incentive size among successful participants were analyzed. Table 4.2 shows the success rates for each point cloud and incentive size. The success rates are 79% for Bremen and 71% for Randersacker with the *small* size, 85% for Bremen and 85% for Randersacker with the *medium* size, and 93% for Bremen and 91% for Randersacker with the *large* size. The percentages in different incentive size categories are similar for both point clouds. Firstly, this indicates that the sizing for both clouds is appropriate, given the similarity in success rates. Secondly, it suggests that success rates are independent of the stimulus, as they remain consistent

Table 4.2.: Success rates by element size and stimuli

Incentive Size	Stimuli	Success Rate
Small	Bremen	79%
	Randersacker	71%
Medium	Bremen	85%
	Randersacker	85%
Large	Bremen	93%
	Randersacker	91%

across different sizes for the same point cloud. This implies that the size of the incentive is the primary factor influencing success rates, rather than the stimulus itself.

Movement Analysis

The movement data of successful participants was examined to determine how incentive size influences exploration patterns. Figures 4.1 and 4.2 display the heatmaps corresponding to the various incentive sizes and point clouds. On the left side in each figure, the heatmap for the *small* incentive size is shown, in the middle for the *medium* size, and on the right for the *large* size. These heatmaps are generated by aggregating the movement data of all successful participants for each point cloud and incentive size. Movement is defined as changing the position of the user in the point cloud. This includes changes along the x, y, and z axes, as well as rotations. Each heatmap visualizes the point cloud on a 2D plane, with the x-axis representing the x coordinate and the y-axis representing the y coordinate. The heatmaps use color-coding to indicate the density of participant movement in a given area, with color intensity reflecting the percentage of movement density. A color scale, along with its corresponding density percentage, is provided on the right side of each figure. To facilitate direct comparison between the heatmaps, movement density is expressed as a percentage, accounting for the varying number of successful participants across different incentive sizes.
All heatmaps indicate a higher density of participant movement in the spawn area. The Randersacker heatmaps display a consistent pattern across all incentive sizes, with the highest movement density concentrated in the center of the point cloud, particularly where the chapel is situated. Similarly, the Bremen heatmaps reveal a uniform pattern across all incentive sizes, though the movement density is more widely dispersed across the point cloud.
For the *small* incentive size, the heatmaps show a greater density of participant movement outside the spawn area compared to those for the *medium* and *large* incentive sizes. There is no major difference in movement density and distribution between the *medium* and *large* sizes, indicating that participants explore the point cloud similarly with these two sizes. This is consistent across both point clouds, suggesting that the

small incentive size encourages broader exploration, while the *medium* and *large* sizes lead less dispersed movement.

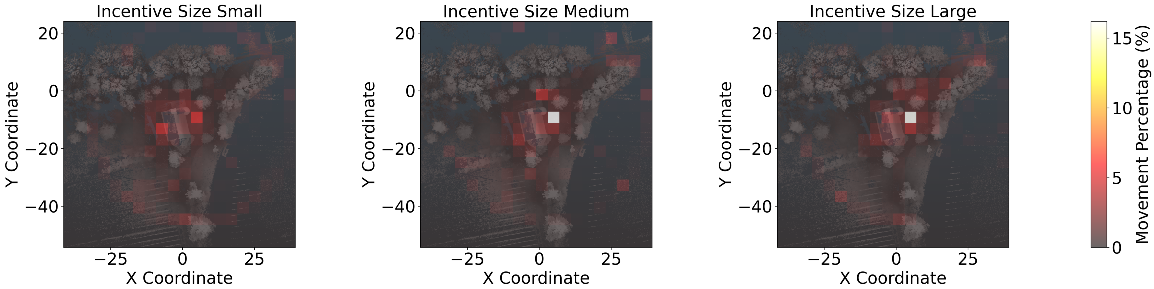

Figure 4.1.: Heatmaps for different incentive sizes (Randersacker)

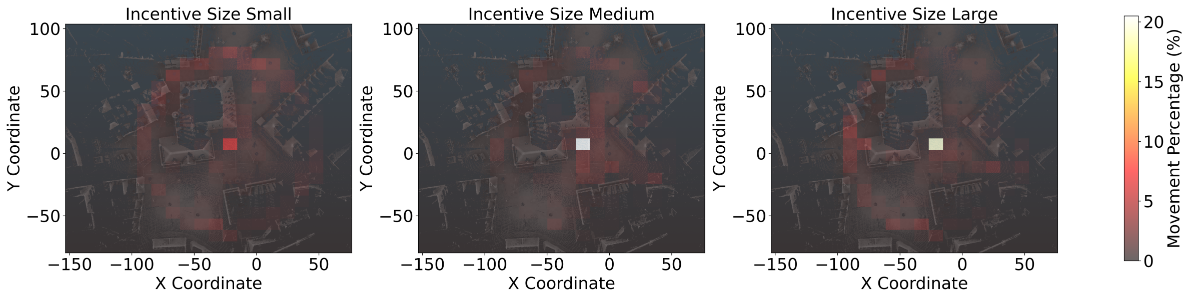

Figure 4.2.: Heatmaps for different incentive sizes (Bremen)

Figure 4.3 provides a detailed overview of the average movement per successful participant for each incentive size. On the x-axis, each point cloud is represented by three bars, corresponding to the *small*, *medium*, and *large* sizes, each accompanied by a confidence interval. The bars go from left (*small*) to right (*large*), with colors dark violet for *small*, grey for *medium*, and yellow for *large*. The legend is located in the top right corner. The y-axis illustrates the average movement, calculated by dividing the total movement by the number of participants. This representation effectively conveys the relationship between point cloud sizes and participant movement. When examining the movement values, there is a noticeable decline in average movement per participant as the incentive size increases. For Randersacker, the average movement decreases from 1760 (*small*) to 1393 (*medium*) and further to 1065 (*large*). Similarly, for Bremen, the movement drops from 1868 (*small*) to 1563 (*medium*), but interestingly, there is a slight increase to 1667 (*large*). The reason for the discrepancy is likely due to the statistical variance. However, the general decline from *small* to *medium* is evident in both locations. The decrease in movement suggests that participants require less exploration to locate elements with larger incentives, as indicated by the Randersacker values and mostly supported by the Bremen values.

The extreme upper confidence level of the size *small* suggests that some participants showed the need to move an extraordinary amount, indicating difficulty in locating the

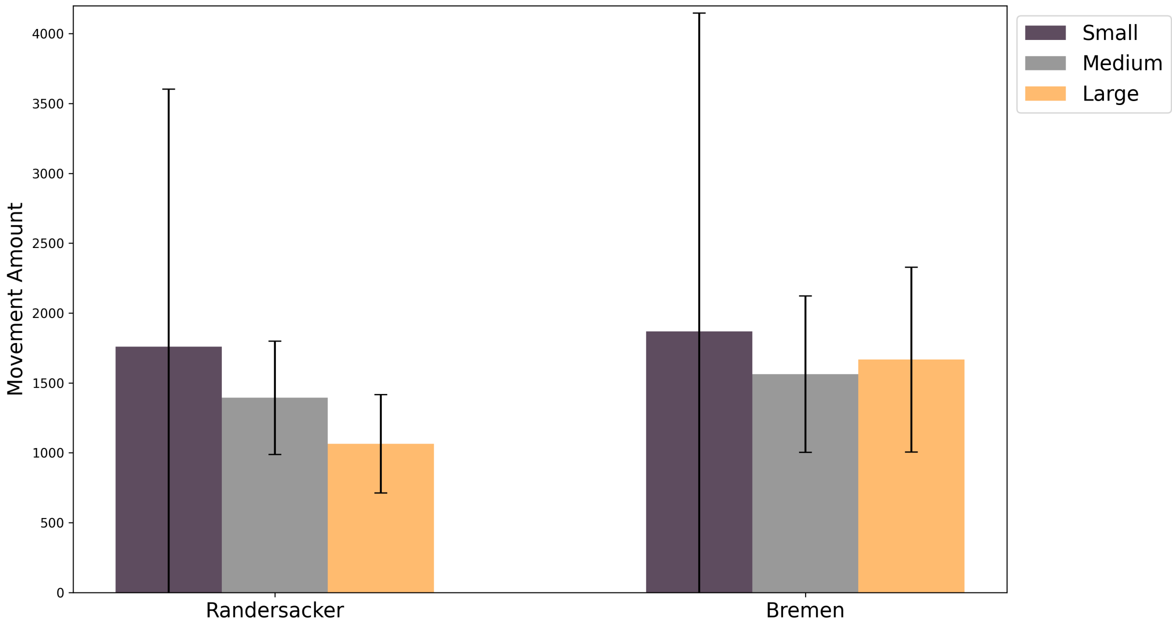

Figure 4.3.: Average movement per cloud and size

element. This extrem is also likely attributed due to the small sample size, leading to higher variance in the data. The confidence intervals for the *medium* and *large* sizes are more consistent and show a decrease in movement with an increase in size. Outliers, as previously mentioned, are Bremen's *medium* and *large* sizes, where the confidence intervals overlap nearly identically. This suggests that the differences in movement between these sizes in Bremen are not statistically significant, showing a different pattern compared to Randersacker, where movement patterns vary.

Incentive Positioning

To further explore how incentive size influences the positioning of incentives, heatmaps for various incentive locations were analyzed. The figures are available in the appendix, Chapter A.1. Each figure presents multiple heatmaps for different incentive positions within the Bremen and Randersacker point clouds, corresponding to the three incentive sizes. The heatmaps are generated like those described in the movement analysis, but in addition with a green cross marking the position of the incentive. In Randeracker, there were used 6 different positions, mainly around the chapel, while in Bremen, 9 different positions were used, spread across the point cloud.
The individual position heatmaps reveal the same pattern to the combined position heatmaps, with the key difference being that participant movement density tends to be more concentrated around specific incentive positions. This tendency is stronger for the *medium* and *large* incentive sizes compared to the *small* size. For instance, in the *medium* incentive size for Bremen, positions 1 and 4 demonstrate the concentration ef-

fect most clearly. At these positions, participants exhibit a notable increase in focused movement, indicating a strong attraction to the elements present. When we examine the same positions within the *large* incentive size, we observe a similar concentration effect. However, the level of movement around the elements is significantly reduced. This trend suggests that while larger incentives enhance focus on specific elements, they may also discourage exploration of the surrounding areas due to their increased clarity and ease of interpretation.

In Randersacker, the concentration effect is less pronounced due to the closer distribution of positions around the chapel. However, a similar pattern can still be observed, particularly at position 4 in the *large* size and positions 2 and 4 in the *medium* size, where the concentration effect becomes more noticeable.

Notably, these findings appear to be independent of content stimuli, indicating a robust effect of incentive size and positioning across different point cloud environments. This suggests that the size of the incentive plays a pivotal role in shaping exploration patterns. Larger incentives tend to encourage more concentrated movement around specific elements, while simultaneously reducing overall exploration, with the largest incentives leading to the most significant decrease in exploratory behavior.

Along with the heatmaps, the success rates for each incentive position and size were analyzed across all participants, who passed the tab filter, to determine whether success is influenced by positioning. Figures 4.4 and 4.5 present the success rates within the Randersacker and Bremen point clouds, respectively.

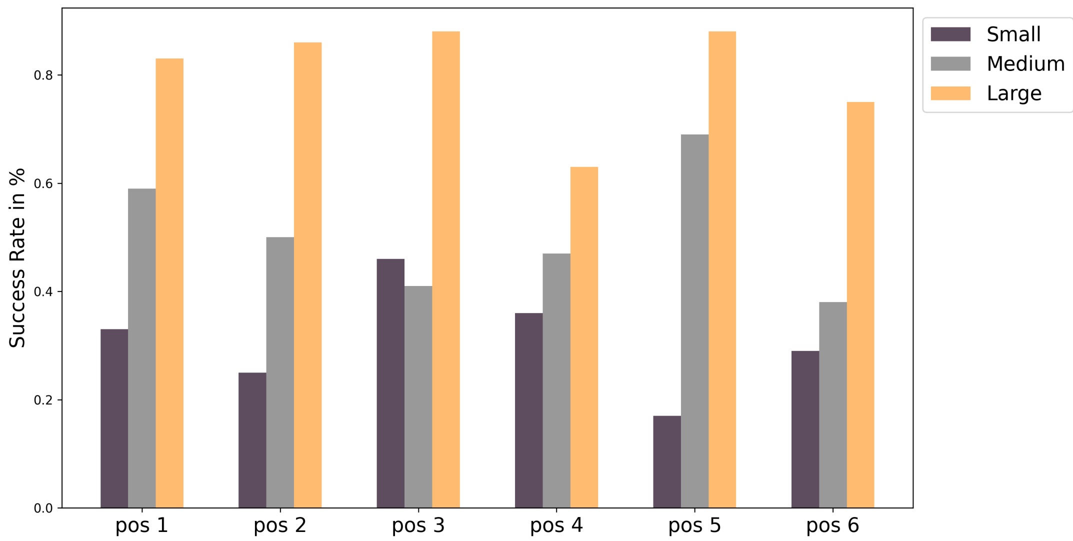

Figure 4.4.: Success rate by element size and positioning for Randersacker

The x-axis shows the name of the incentive positions, while the y-axis indicates the success rate in percentage. Each position has three bars representing the incentive sizes. The leftmost bar is dark violet for *small*, the middle bar is grey for *medium*, and the rightmost bar is yellow for *large*. A legend is found in the top right corner. The corresponding X, Y and Z coordinates of the positions are provided in methodology chapter

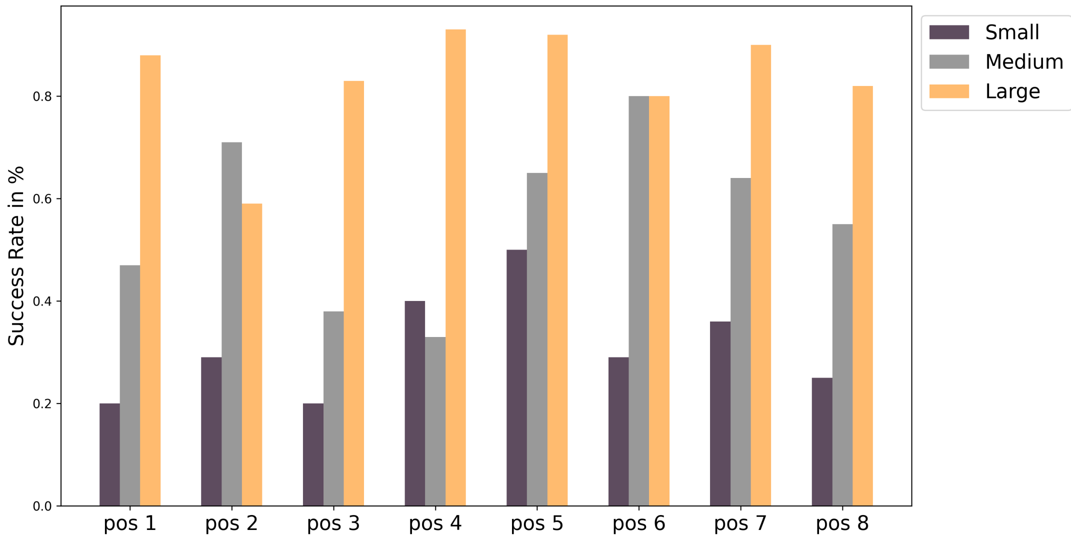

Figure 4.5.: Success rate by element size and positioning for Bremen

in table 3.4.
A distinct pattern emerges, with large elements consistently outperforming smaller ones and often achieving success rates above 80%, with the exception of position 2 and 6 in Bremen. Medium-sized elements typically fall in the middle range, while small elements generally have the lowest success rates, often below 40%. The success rates can vary notably based on the position within the point cloud. Positions like 3 in Randersacker and 4 in Bremen exhibit similar performance across the *small* and *medium* sizes. These cases, along with the noted outliers, highlight the significant impact that positioning can have on success rates. However, the size of the incentive remains the primary factor, showing a clear linear increase in success rates as the incentive size grows.

QoE ACR Rating

Finally, by bringing the QoE ACR rating into context, the impact of sizes on rating relibility can be assessed. Figures 4.7 and 4.6 show the QoE ACR rating for Bremen and Randersacker, respectively. The x-axis represents various point cloud reductions, including the original cloud. Each position on the x-axis is further divided into the three tested sizes, using the same color palette for the sizes as in the previous diagrams, with the legend located in the top right corner. The y-axis displays a scale ranging from 0, indicating no evaluations collected, to 1 (poor) and 5 (excellent), representing the average ACR ratings for each tested point cloud and size. Each position on the x-axis corresponds to a bar in the chart, which displays the average ACR rating along with a confidence interval.
The diagrams are based on ratings from successful participants, with the number of participants varying significantly by incentive size. Only 6 participants provided ratings for the *small* size, while 24 participants did so for the *large* size. Consequently, there is

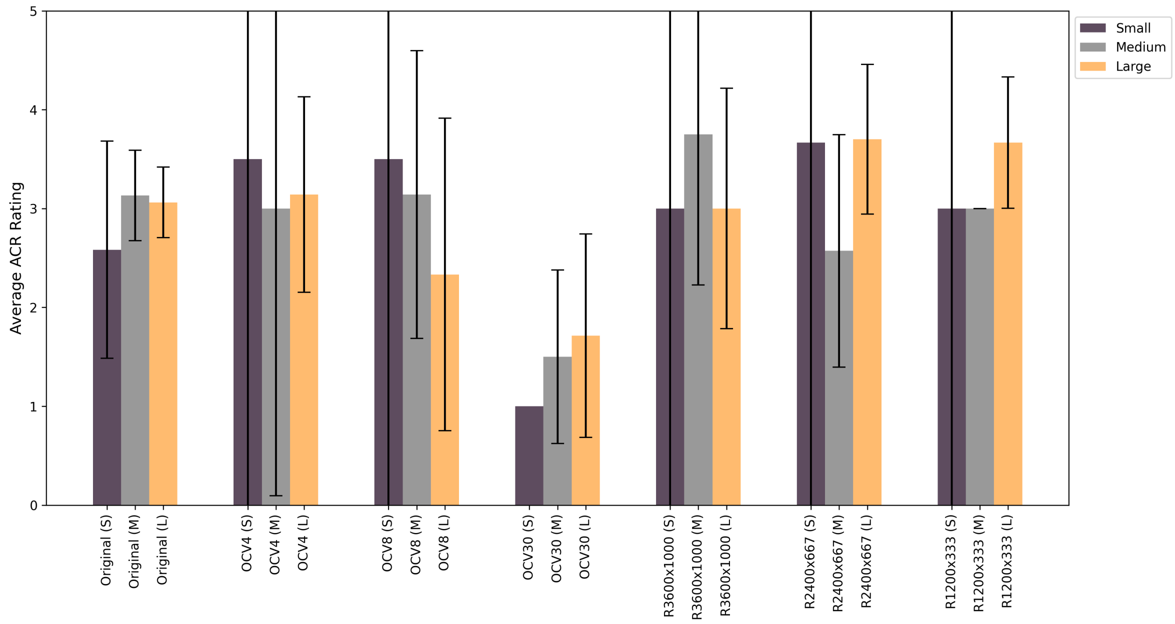

Figure 4.6.: QoE ACR rating for Randersacker by element size

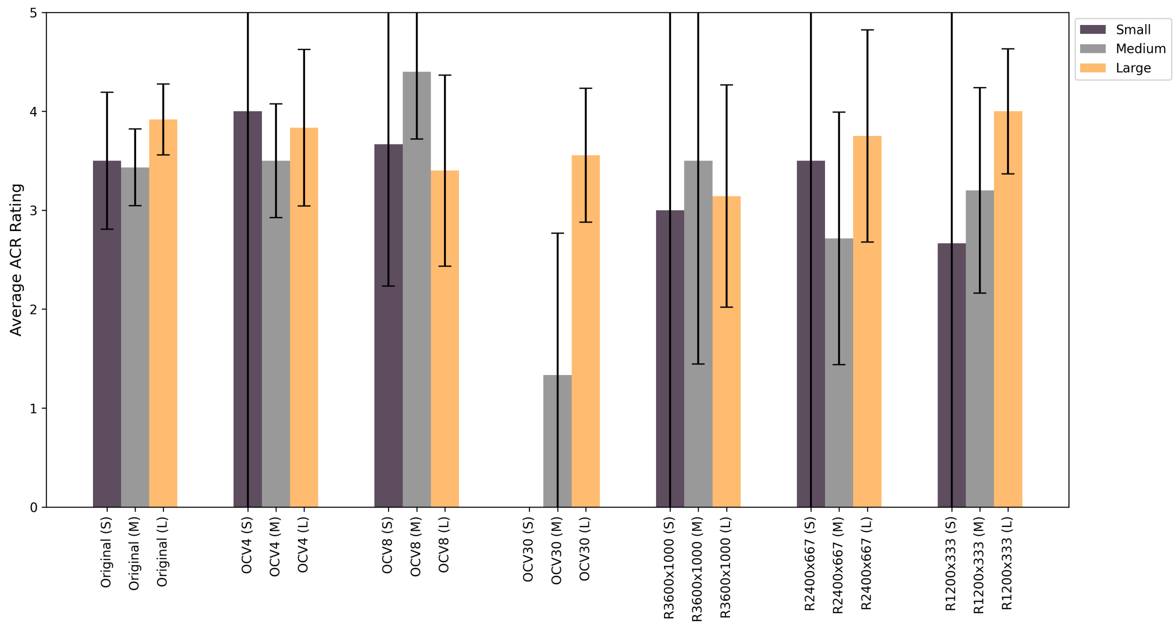

Figure 4.7.: QoE ACR rating for Bremen by element size

a higher statistical variance for the *small* size compared to the *large*. For the category OCV30 (S) in Bremen, there are no ratings available, as none of the 6 participants were randomly assigned that reduction level.
When comparing results across different sizes within a cloud category, a strong correlation emerges between ratings, reduction levels, and sizes, with the exception of Bremen

OCV30, where the *large* and *medium* sizes deviate from this trend. Minor variations occur like for example, in Bremen, the difference between the sizes in OCV4 and the Original is only 0.5, a pattern similarly observed across all other comparisons, including Randersacker. The high overlap in confidence intervals underscores this similarity. Stronger differences can also be seen, such as in Bremen, where the *medium* size for OCV8 approaches a rating of 4.5, while the *large* size falls below 3.5. Conversely, in the R2400x667 category for Bremen, the *large* size is closer to 4, while the *medium* size drops below 3. A similar pattern is observed in Randersacker, particularly in categories like R3600x1000 and R2400x667. Despite the variations in size ratings within each reduction category, the overall rating trends remain consistent. The most significant shifts in ratings occur when comparing the different stimulus variants, rather than within size categories. For instance, when transitioning from OCV8 to OCV30 in Randersacker, or when comparing OCV reductions in Bremen to projection-based reductions, notable differences emerge. This suggests that the reduction level of the stimulus is mainly impacting the results than the size of the incentive.

An analysis of the rating trends reveals a stimulus-dependent difference based on the reduction methods employed. In Bremen, both the OCV reduction method and the original cloud consistently receive higher ratings compared to their counterparts in Randersacker, across all incentive sizes. The ratings in Bremen typically hover around 3.5, while those in Randersacker are closer to 3, indicating a clear preference for the reduction methods used in Bremen. When examining projection-based reductions, the ratings in Bremen are slightly lower, nearing 3, compared to OCV reductions. In Randersacker, the ratings for both reduction methods are comparable, also hovering around 3, with the notable exception of the OCV30 reduction, which receives a particularly low rating of 1.5. This discrepancy highlights the potential shortcomings of the OCV30 method in this context. Furthermore, no significant correlation exists between the size of the incentive and the ratings. This suggests that the effectiveness of the reduction methods is more influenced by the characteristics of the stimulus itself than by the size of the incentive. Overall, these findings underscore the importance of considering stimulus properties when evaluating the effectiveness of different reduction techniques.

In conclusion, the data indicates that incentive size significantly affects both success rate and exploration, while positioning influences the focus of the exploration area. *Small* incentives encourage greater exploration of the point cloud, as evidenced by a higher percentage of participant movement beyond the spawn area and increased absolute movement. However, this increased exploration does not translate into a higher success rate, which remains at only 30%. In contrast, *large* incentives result in more focused exploration around specific positions, with less dispersed movement leading to a higher success rate of 80%. Furthermore, size seems to have no significant impact on the QoE ACR rating, with the reduction level of the stimuli and the stimuli itself having a more significant influence.

To balance exploration and success rate effectively, *medium*-sized incentives appear to be the optimal choice. By strategically placing these incentives, researchers can control the exploration area and increase success rates. This approach avoids the pitfalls associated with *small* incentives (low success rates) and *large* incentives (reduced exploration and excessive focus on specific elements). The differences observed between the Bremen and Randersacker clouds further underscore the importance of considering the specific layout and features of each point cloud when determining optimal incentive placement strategies.

4.1.2. Influence of different Incentive Quantities

To assess the impact of different incentive quantities on the evaluation results, three distinct incentive quantities were tested across separate study runs. Specifically, one, two, and three incentives per point cloud were examined, using a *medium* incentive size as detailed in subsection 4.1.1 and the sequential comparison method as detailed in section 3.3. Consistent with the size studies, a run was deemed successful if participants correctly completed at least five evaluations and navigated away from the survey tab no more than four times.

Table 4.3 shows the relationship between incentive quantity and evaluation results across different study runs. The table is structured like the size table, with the difference that the incentive quantities are displayed instead of the sizes.

For the one incentive quantity, 37 participants were involved. Of these, 28 passed the tab filter, and 15 successfully completed the study. For the two incentive quantity, 16 participants participated, with 12 passing the tab filter and 7 successfully completing the study. Finally, for the three incentive quantity, 16 participants were involved, with 9 passing the tab filter and 4 successfully completing the study. In percentage terms, the success rates were 54% for the one incentive quantity, 58% for the two incentive quantity, and 44% for the three incentive quantity. The average time spent for the one incentive quantity is 9.9 minutes, for the two incentive quantity is 14.1 minutes, and for the three incentive quantity is 13.2 minutes. The results indicate that the one and

Table 4.3.: Incentive quantities and success rates

Quantity	Participants	Passed Tab Filter	Successful	Success %	Avg. Time
One	37	28	15	54%	9.9 min
Two	16	12	7	58%	14.1 min
Three	16	9	4	44%	13.2 min

two incentive quantities yield similar success rates, while the three incentive quantity shows a noticeable decline. This trend may be attributed to the increased cognitive load and complexity of decision-making associated with multiple incentives. Supporting this assumption is the high number of filtered participants in the three-incentive group, which decreased from sixteen to nine, indicating a greater level of distraction. There is a notable increase in average time spent when transitioning from one to two incentives. However, the time spent remains similar between the two and three incentive groups. This suggests that participants require more time to complete the study with two incentives, but the time spent does not increase further with the addition of more incentives.

Movement Analysis

Next to the success rates, the movement data of successful participants was analyzed to determine how incentive quantity influences exploration patterns. Figures 4.8 and 4.9 display the heatmaps corresponding to the various incentive quantities and point clouds. The same heatmap generation process as described in subsection 4.1.1 was applied to the movement data of successful participants for each point cloud and incentive quantity. In the figures, the left side shows the heatmap for the one incentive quantity, the middle for the two incentive quantity, and the right for the three incentive quantity. It shows

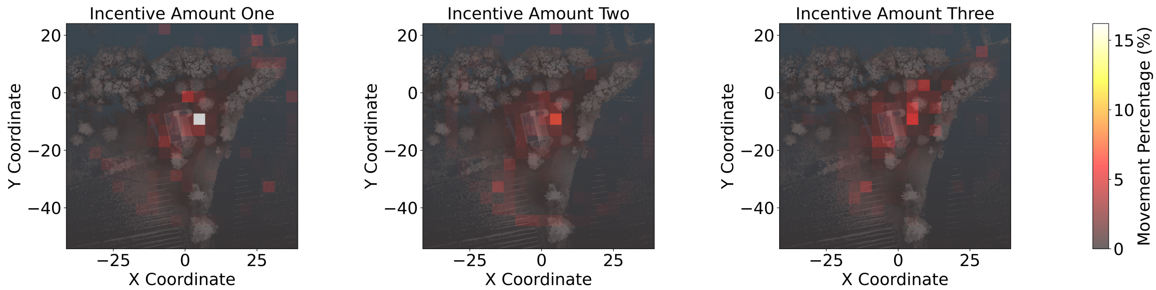

Figure 4.8.: Heatmaps for different incentive quantities (Randersacker)

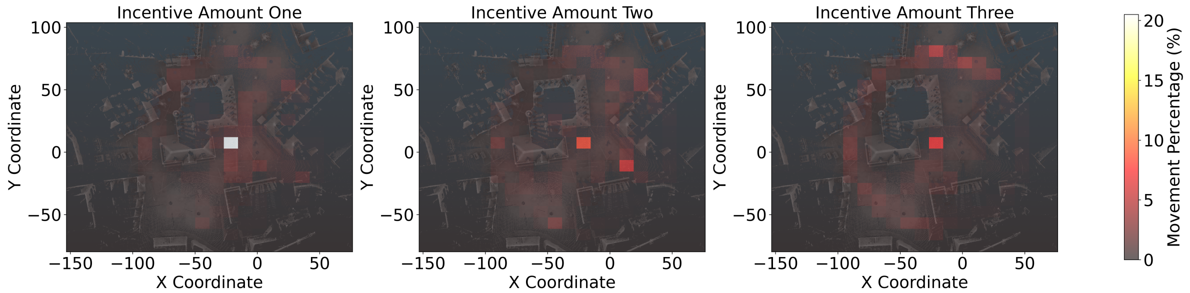

Figure 4.9.: Heatmaps for different incentive quantities (Bremen)

that higher incentive quantities correlate with increased movement density and more uniform exploration of the point cloud. This is evidenced by greater color coverage and more even color intensity distribution across the heatmap as incentives rise. This signals that participants explore more of the point cloud with higher incentive quantities, as they are incentivized to visit more areas to locate the elements. The difference becomes noticeable when transitioning from a single incentive to two incentives. However, when increasing the number of incentives from two to three, the exploration pattern remains quite similar, suggesting a threshold effect where additional incentives do not drastically

change the behavior. This effect is stimuli independent, as the same pattern is observed in both Bremen and Randersacker.
Figure 4.10 provides a detailed overview of the total movement and average movement per successful participant for each incentive quantity. It is constructed similarly to the previous diagram of the size analysis, but it is filled with data for the different quantities instead of sizes. The color mapping has been changed. Now dark violet stands for One, grey for Two, and yellow for Three, with the legend located in the top right corner.

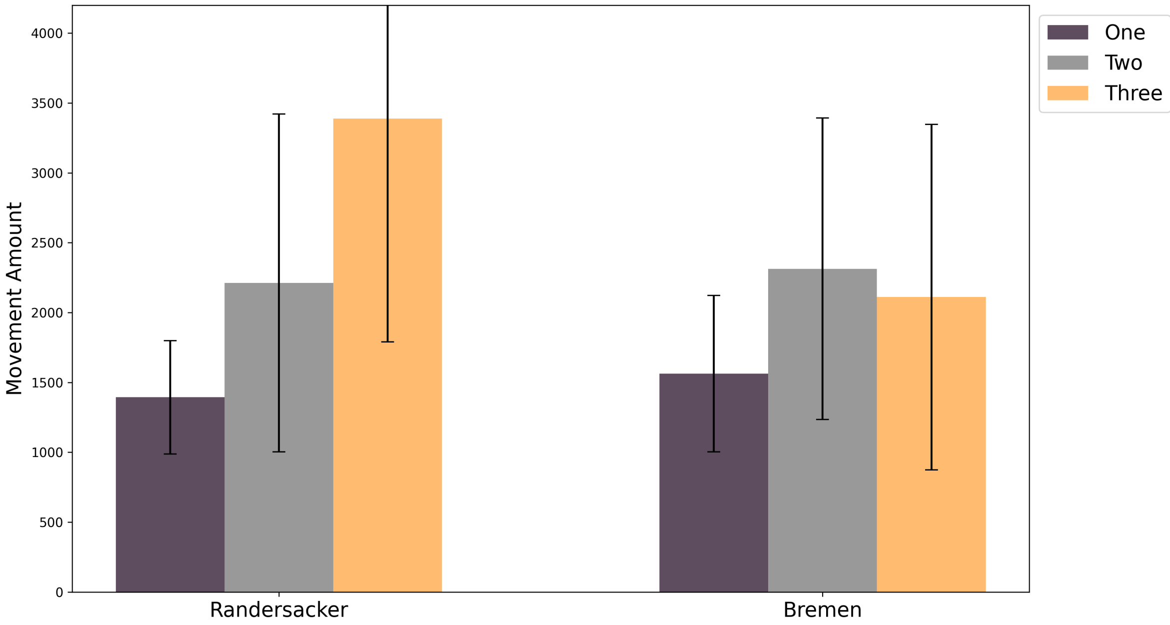

Figure 4.10.: Average movement per cloud and quantity

The results show a noticeable increase in average movement per participant as the incentive quantity rises. For Randersacker, the average movement increases from 1393 (One) to 2212 (Two) and further to 3389 (Three). Similarly, for Bremen, the movement rises from 1563 (One) to 2313 (Two) and slighty down to 2111 (Three). The discrepancy between the Two and Three incentive quantities in Bremen is likely due a statistical variance, as only 4 participants provided movement data for the Three incentive quantity. The general trend of increased movement with higher incentive quantities is consistent across both point clouds, suggesting that participants explore more of the point cloud with additional incentives.
However, all confidence intervals between the different quantities overlap slightly, though the trend still shows higher movement with higher quantities. For Bremen, the overlap between the Two and Three quantities is nearly identical, making it an outlier. This indicates that the differences in movement between the mentioned quantities in Bremen are not statistically significant, showing a different pattern compared to Randersacker. This is likely attributed to the small sample size and/or the different point cloud layout.

Incentive Positioning

An analysis of the impact of incentive quantity on success rates in the context of positioning is provided. However, due to the large number of possible combinations, there is not enough data per combination, and it would be beyond the scope of this thesis to analyze each influence on exploration. Therefore, only a limited number of combinations are addressed here. A full overview of the success rates for each captured combination of positions and quantities is provided in the appendix, Chapter A.2. Each table lists combinations of positions, appearances, and success rates for the different quantities and stimuli, sorted by success rate and indexed accordingly. Success is defined as correctly identifying all elements of the combination while adhering to the participation rules.
In Randersacker, with six possible positions and a quantity of two, the total number of combinations is 15. In contrast, Bremen, with eight positions, has 28 possible combinations. Furthermore, when considering a quantity of three incentives, the number of combinations increases to 20 for Randersacker and 56 for Bremen.
Significant differences in success rates can arise depending on the combinations of positions. For instance, in Randersacker, with a quantity of two, index twelve exhibits a poor success rate. Out of six appearances, only one was correct, resulting in a success rate of 17%. Similarly, low success rates are observed for index eighteen in Bremen with a quantity of two, and index seventeen in Bremen with a quantity of three. In contrast, index two (Randersacker quantity two) achieved a success rate of 100%, with four correct answers out of four attempts. Other high success rates include index one and three in Randersacker with a quantity of three, achieving 100% and 75% success rates, respectively.
Many combinations appeared only once, especially in the three quantity group, making it difficult to draw definitive conclusions. However, there are more appearances in the two quantity group, allowing for a more detailed analysis. The data clearly shows that the positioning combinations have a significant impact on the success rates, with certain combinations leading to higher or lower outcomes. Thus, it would be valuable to eliminate the less successful combinations and focus on the more successful ones.
Still, it is important to note that the impact of the combinations on exploration is not analyzed, and a balance between exploration and success rate is needed when selecting the optimal combination.

QoE ACR Rating

Finally, the QoE ACR rating was analyzed to determine the impact of incentive quantities on rating reliability. Figures 4.12 and 4.11 show the QoE ACR rating for Bremen and Randersacker, respectively. The diagrams are constructed like the size diagrams but based on the number of incentives studied. They are labeled as (1) for one incentive, (2) for two incentives, and (3) for three incentives. The color palette is the same, but the mapping is different and tailored to the quantities. The legend is located in the top right corner.

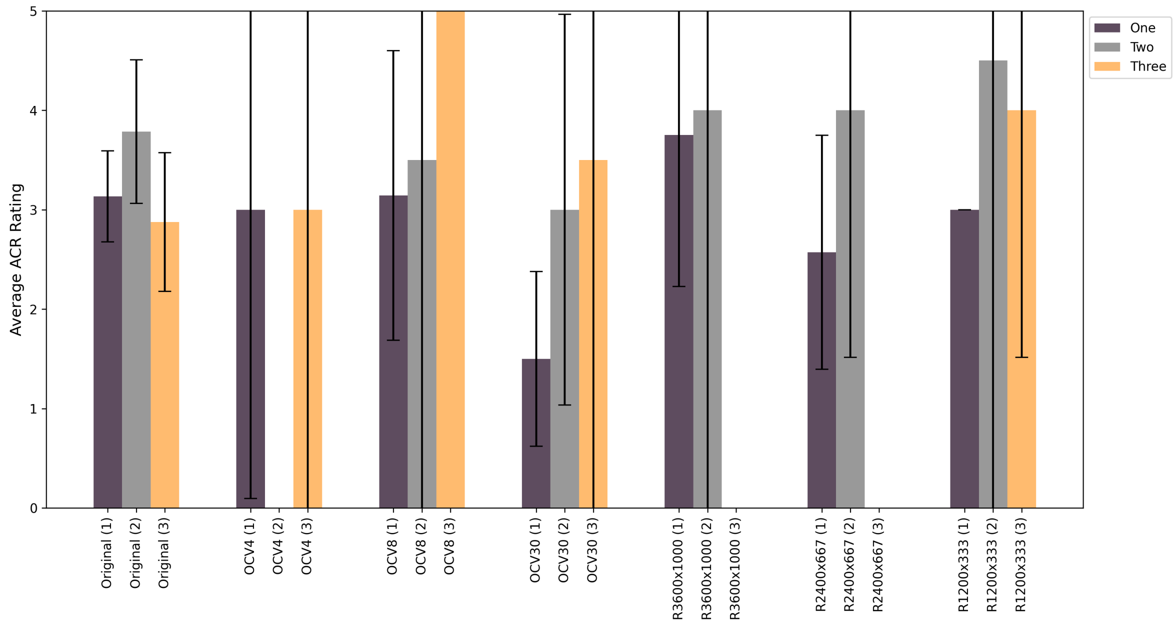

Figure 4.11.: QoE ACR rating for Randersacker by element quantity

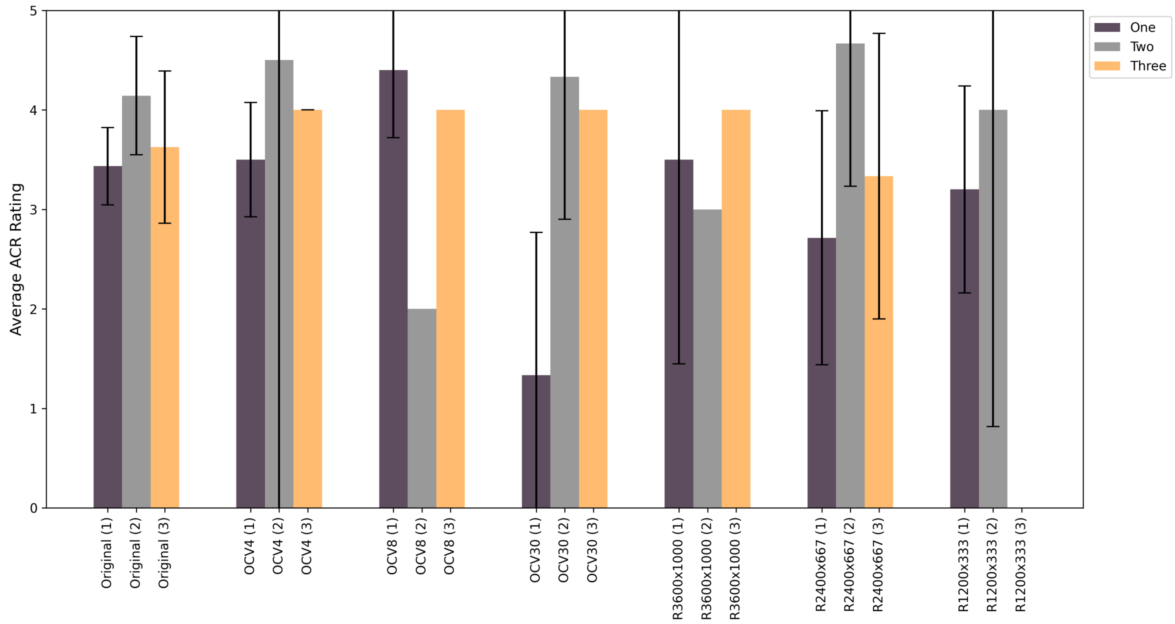

Figure 4.12.: QoE ACR rating for Bremen by element quantity

The results reveal significant inconsistencies between different quantities at the same reduction level. Extreme cases include the OCV8 reduction in Bremen, where the two-quantity is rated 2.0, while the one- and three-quantities are rated between 4.0 and 4.5. Other examples in this category are OCV30 and R2400x667 in Bremen, as well as OCV8, OCV30, and R1200x333 in Randersacker. Additionally, there is no data for

the R1200x333 three-quantity in Bremen, nor for the OCV4 two-quantity, R3600x1000 three-quantity, and R2400x667 three-quantity in Randersacker, as none of the seven participants in the two-quantity group or the four participants in the three-quantity group were randomly assigned to these reductions. The mentioned inconsistencies are likely due to the small sample size, which leads to high variance in the data, as evidenced by the large confidence intervals.
Where the most data is available at the original reduction level, due to the sequential comparison format, the ratings exhibit a similar pattern but with somewhat larger variance compared to the size ratings.
In Bremen, the one-quantity is rated below 3.5, while the two-quantity is rated at 4.2. In Randersacker, the variance is even greater, with the three-quantity rated around 2.75 and the two-quantity around 3.75.
A pattern emerges, not only for the originals, indicating that the highest variance occurs between the two-quantity and the other quantities, with the two-quantity frequently receiving the highest ratings across all reductions. This is also evidented by the confidence intervals, which are generally place higher for the two-quantity than for the other quantities. In Bremen, the two-quantity is rated the highest in five out of seven cases, while in Randersacker, this occurs in four out of seven. This suggests that the two-quantity is generally preferred, regardless of the reduction level and quantity does have an impact on ratings. However, the limited amount of data is insufficient to draw definitive conclusions.
Although there are inconsistency, a small trend remains visible, much like the size-based QoE ratings. OCV reduction is generally rated worse for Randersacker than for Bremen being in similar rating category as sizes. In contrast to the size-based QoE ratings, the quantity-based ratings in Randersacker show better scores for projection-based reductions, approaching 3.5. In Bremen, however, the ratings are worse, slightly falling below those for OCV reductions, similar to the findings in the size analysis. This suggests that despite the variability, the reduction level and the specific stimuli tested have a greater influence on the ratings than the quantity of incentives.
In conclusion, the data indicates that the number of incentives significantly impacts exploration and success rates, though it does not necessarily correlate with QoE ACR ratings. Since exploration increases with the number of incentives and the coverage remains similar for two and three incentives, the additional incentive does not significantly change the exploration pattern. Furthermore, the measured success rates are considerably higher for two incentives compared to three, indicating that two incentives are the optimal choice. Additionally, filtering by combinations can alter the success rates, but this must be balanced with exploration patterns.

4.2. Impact of Wording

To evaluate the impact of different wordings on the assessment results, two distinct sets of instructions were tested across separate study runs. In the first condition, referred to as *no emphasis*, participants received instructions that included information about the evaluation task and associated incentives. In the second condition, referred to as *emphasis*, participants received the same instructions as in the first condition, with an additional colored banner at the end of the instructions, just before proceeding to the trial run. The banner stated:

> **Important: While finding the hidden letters/numbers is part of the mechanism to validate the interaction, the primary focus is on evaluating the overall quality of the point cloud.**

Furthermore, after successfully completing the trial run, participants in the *emphasis* condition received another reminder before starting the main study, which stated:

> **While searching for control letters/numbers, remember that rating the cloud quality is the top priority.**

In both conditions, the standard instructions informed participants that the primary objective was to evaluate the point cloud quality, with the hidden elements serving as part of the validation mechanism. To ensure participants remained focused on the primary objective, the additional emphasis in the second condition was tested for its impact. Previous study runs all conducted under the *emphasis* condition, including tested varying sizes and quantities.
For each condition, the sequential comparison method was used, with the incentive size set to *medium* and the quantity set to one. A run was deemed successful if participants correctly identified at least five elements and did not navigate away from the survey tab more than four times.
Table 4.4 shows the relationship between emphasis and evaluation results across different study runs. In same manner as the previous tables, the table is structured with the difference that the emphasis conditions are displayed instead of sizes or quantities.
For the *emphasis* condition, 37 participants were involved, with 28 passing the tab filter and 15 successfully completing the study. For the *no emphasis* condition, 13 participants participated, with 11 passing the tab filter and 8 successfully completing the study. In percentage terms, the success rates were 54% for the *emphasis* condition and 72% for the *no emphasis* condition. The average time spent for the *emphasis* condition is 9.9 minutes, while for the *no emphasis* condition, it is 8.6 minutes. A significant difference

Table 4.4.: Emphasis and success rates

Emphasis	Participants	Passed Tab Filter	Successful	Success %	Avg. Time
Yes	37	28	15	54%	9.9 min
No	13	11	8	72%	8.6 min

in success rates is observed between the two conditions, with the *no emphasis* condition yielding a higher success rate of 72% compared to 54% for the *emphasis* condition. The average time spent is slightly lower for the *no emphasis* condition compared to the *emphasis* condition, suggesting that the additional emphasis may have contributed to the increased time spent. The reduced success rate in the *emphasis* condition may stem from the increased focus on evaluating the quality of the point cloud. This heightened emphasis likely diverted participants attention away from locating the hidden elements, which are critical for determining success in the study. As a result, participants may have spent more time assessing the point cloud and less time searching for these hidden elements, ultimately leading to lower success rates.

Movement Analysis

Figures 4.13 and 4.14 present the heatmaps for the *emphasis* and *no emphasis* conditions corresponding to the Randersacker and Bremen point clouds, respectively. These heatmaps were generated using the same method outlined in the preceding sections.

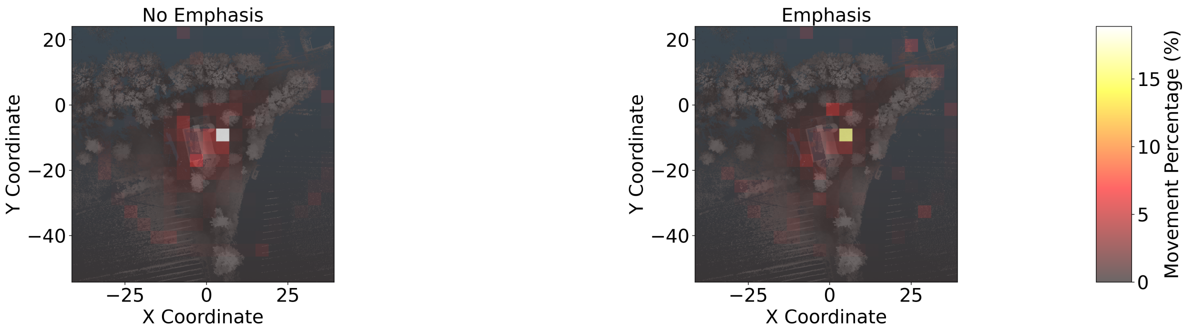

Figure 4.13.: Heatmaps for different emphasis conditions (Randersacker)

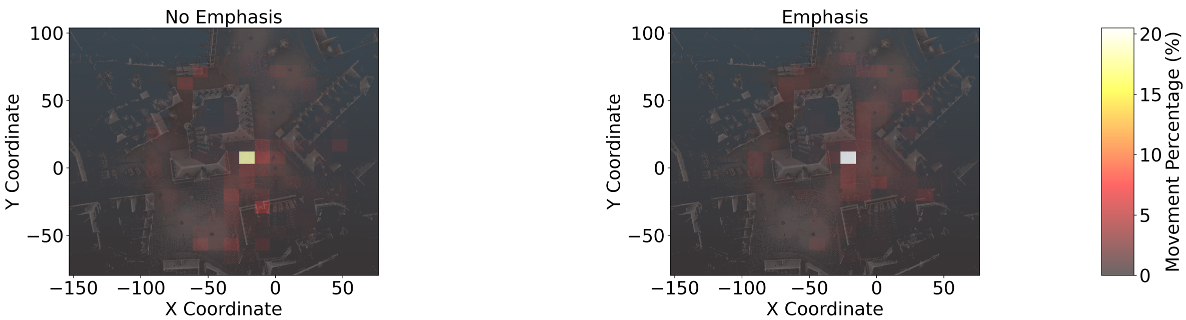

Figure 4.14.: Heatmaps for different emphasis conditions (Bremen)

While there are minor differences in movement distribution between the two conditions, the overall patterns remain consistent. The *no emphasis* condition for Bremen and the *emphasis* condition for Randersacker show slightly lower percentages of movement in the spawn area. However, the overall movement distributions are similar.
Thus, there is no significant difference that clearly indicates how emphasis affects the exploration pattern. Consequently, no connection exists between emphasis and exploration behavior with regard to movement.

Moving on to the average movement per successful participant, Figure 4.15 is constructed similarly to the previous diagrams, but it incorporates data for the different emphasis conditions instead of sizes or quantities. Furthermore, there are only two conditions, so the color mapping is different. Dark violet represents the *emphasis* condition, while yellow represents the *no emphasis* condition. The legend is located in the top right corner.
For Randeracker, the average movement is 1393 for the *emphasis* condition and 996 for the *no emphasis* condition. For Bremen, the movement is 1563 for the *emphasis* condition and 1721 for the *no emphasis* condition. The results reveal a significant decrease in average movement per participant in the *no emphasis* condition compared to the *emphasis* condition in Randersacker. In contrast, Bremen exhibits a slight increase in average movement under the *no emphasis* condition. This discrepancy may be attributed to the different point cloud layouts and features, which can influence exploration patterns. The confidence intervals are mostly overlapping for Bremen, while for Randersacker, they also overlap to a high degree, but not as much as in Bremen. This suggests that there is no significant difference in movement between the two conditions, indicating that the emphasis does not influence exploration behavior.

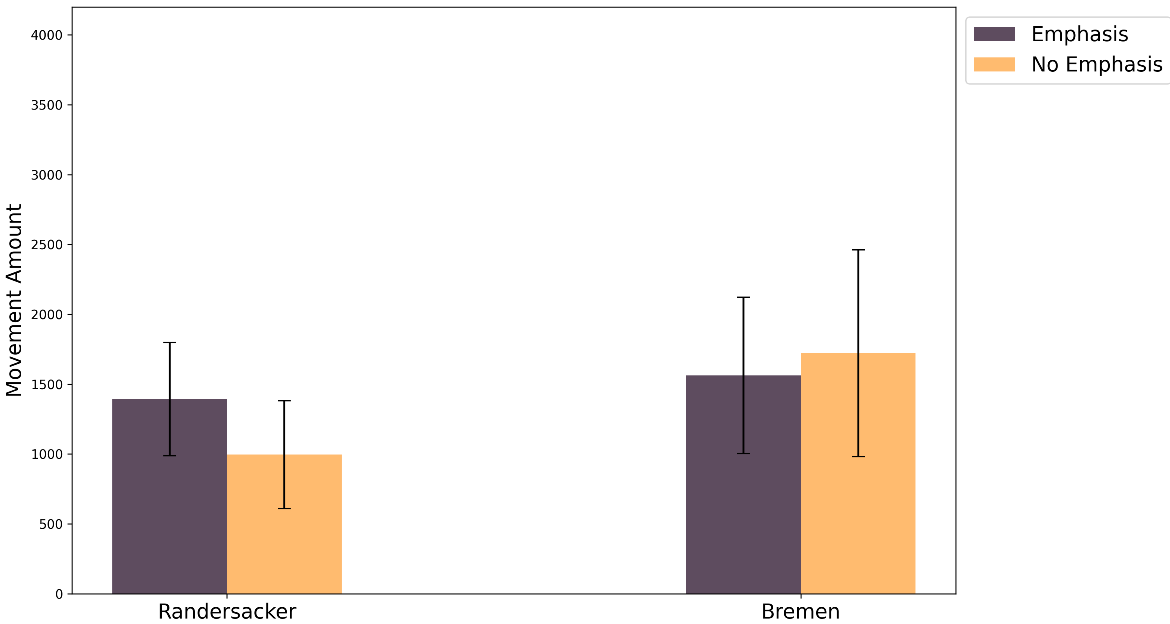

Figure 4.15.: Average movement per cloud and emphasis condition

QoE ACR Rating

For the QoE ACR rating, Figures 4.16 and 4.17 show the QoE ACR rating for Bremen and Randersacker, respectively. The diagrams are constructed similarly to the previous diagrams but based on the emphasis conditions. The (N) label represents the *no*

emphasis condition, while the (E) label represents the *emphasis* condition. Dark violet represents the *emphasis* condition, while yellow represents the *no emphasis* condition. The legend is located in the top right corner.
In Randersacker, the *no emphasis* condition consistently receives higher ratings than the *emphasis* condition, except for OCV30 and R3600x1000, where the ratings are similar. Ratings for the *emphasis* condition in the categories Original, OCV4, and OCV8 hover around 3, while those for the *no emphasis* condition are slightly above 4. For R3600x1000, both conditions are rated around 3.75, with *emphasis* slightly higher. The outlier OCV30 has an *emphasis* rating of 1.5 and a *no emphasis* rating of 1, which remain close to each other. Another outlier is R2400x667, for which the *no emphasis* rating is missing due to a lack of participants in that condition. The trend in Randersacker indicates that the *no emphasis* condition generally receives higher ratings than the *emphasis* condition, with a few exceptions that remain relatively close in value.
A similar pattern is observed in Bremen, where the *no emphasis* condition receives higher ratings in four out of seven categories. In the Original category, the *no emphasis* rating is approximately 3.5, while the *emphasis* rating is around 4.1. For R2400x667, both conditions yield ratings close to 3.5, and a similar trend is noted for R1200x333. Other categories, such as OCV8, show ratings of 4.5 for *emphasis* and 4.4 for *no emphasis*, while R3600x1000 has ratings of 3.5 for *emphasis* and 3.4 for *no emphasis*, reflecting only minor differences. The OCV30 category is an outlier, with a rating of 1.1 for *emphasis* and 4.5 for *no emphasis*, indicating a significant disparity.
Additionally, a stimulus-dependent trend was observed, similar to previous analyses, with Bremen's OCV reduction receiving higher ratings than Randersacker's OCV reduction. The projection-based reductions in Bremen are rated slightly lower compared to its OCV reductions, while in Randersacker, the ratings for projection- based reductions are similar to those of the OCV reductions.
The results indicate that the *no emphasis* condition generally yields higher ratings than the *emphasis* condition, with a few exceptions. This effect is particularly pronounced in Randersacker. In instances where this trend does not hold, the ratings remain close to each other, with only minor variations. The effect is also supported by the confidence intervals, which are generally higher placed for the *no emphasis* condition than for the *emphasis* condition. This pattern may be attributed to the additional emphasis in the *emphasis* condition, which could have influenced participants' perceptions of point cloud quality. The emphasis might have led participants to adopt a more critical view of the quality, thereby affecting their ratings. Overall, the results suggest that the *emphasis* condition may negatively impact the QoE ACR rating, as participants may have become more critical of the point cloud quality due to the heightened emphasis. *no emphasis* condition.
The only significant differences between the two conditions are in the success rate and the QoE ACR rating. The movement analysis does not reveal significant differences between the conditions, suggesting that emphasis does not influence exploration behavior. Given this context, the *no emphasis* condition appears to be the optimal choice for achieving higher success rates. The ratings are generally higher for the *emphasis* condition. However, they still reflect the same trends observed in the *no emphasis* condition.

This indicates that emphasis does not significantly alter the overall rating patterns. Therefore, using the *no emphasis* condition results in no losses and contributes to a greater number of successful participants.

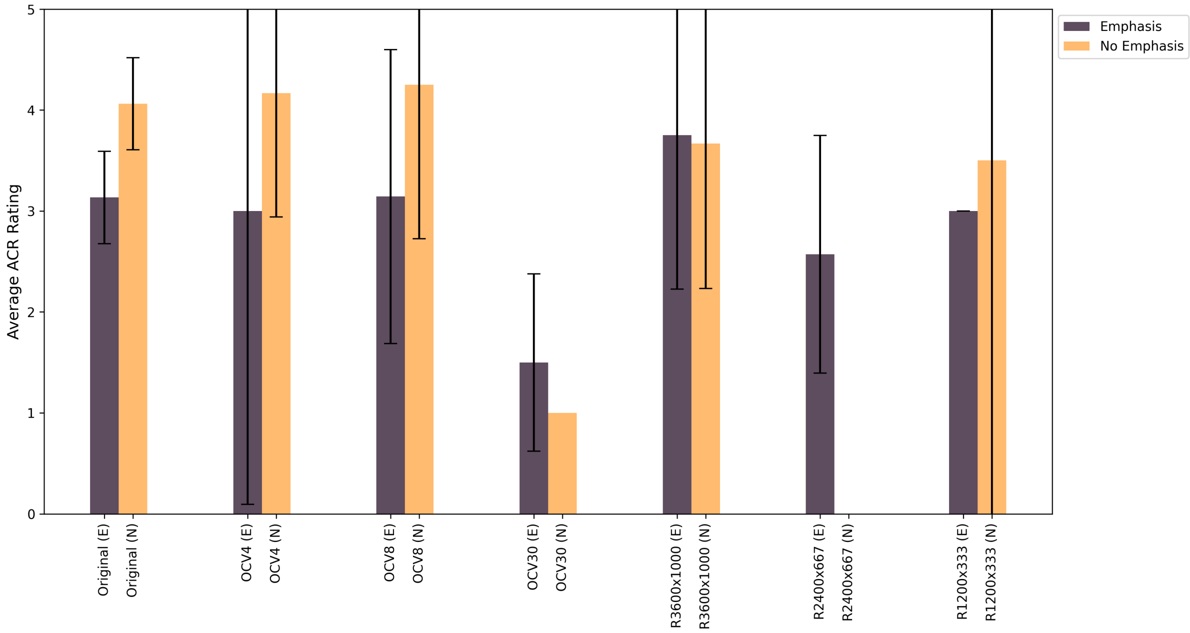

Figure 4.16.: QoE ACR rating for Randersacker by emphasis condition

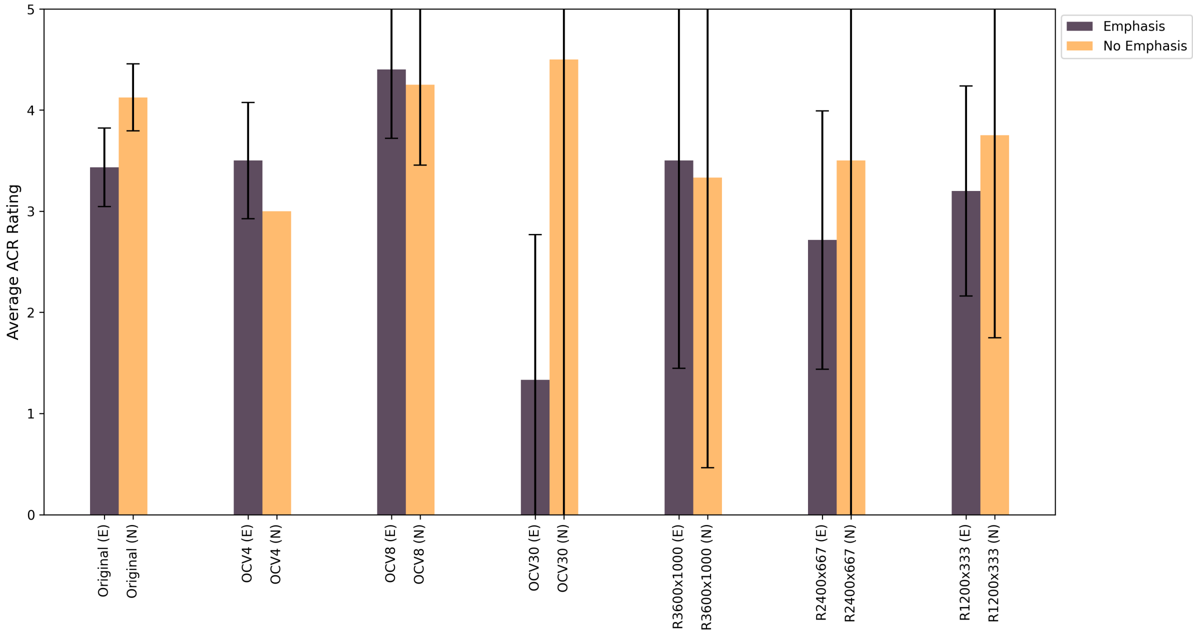

Figure 4.17.: QoE ACR rating for Bremen by emphasis condition

5. Conclusion and Outlook

There are many current and future applications, like in VR, autonomous driving and manufacturing, which depend on the data provided by point clouds. Especially in the case of realistic VR scenarios, point clouds contribute precise data for creating authentic environments. Success or failure of the adaptation of such technologies is decided by the user experience. Technical parameters such as resolution or jitter can only partially predict user satisfaction, thus there is a need to gather user feedback through studies. Most studies are conducted in controlled laboratory environments, which often lack scalability, cost-effectiveness, and diversity. Hence, crowdsourcing emerges as a promising alternative.

Nevertheless, crowdsourcing studies lack control over user behavior, which can result in unreliable data. Because of this, it raises the question of whether crowdsourcing-based QoE studies on point clouds are even practical, and if so, what can be improved? Studies in which participants can move freely within a 3D environment of point clouds have the distinct advantage of offering a more immersive experience than watching video clips or pictures. It is important to ensure real focus on the task, as well as to have quality exploration of the point cloud. Therefore, it is essential to integrate a mechanism that can guarantee this. Leading to the question of whether the implementation of a mechanism to enhance participant interaction can improve the QoE evaluation of a crowdsourcing study with point clouds?

In this thesis, a web-based crowdsourcing study was employed to evaluate the QoE of two-point cloud stimuli. It deals with the construction of a study design for crowdsourcing based studies and the implementation of mechanisms to improve user engagement and data reliability. In the study, users interacted with point clouds of varying reduction levels by freely navigating through them. Afterward, they rated the clouds individually and compared them to the same cloud at different reduction levels. The mechanism employed is a dynamically generated element, which can take the form of a letter or number. This element is placed within the point cloud, and users must find and name it during the evaluation, ensuring their active engagement. Three distinct element sizes and quantities, as well as varying positions and combinations, were evaluated to assess their influence on cloud exploration and data reliability. Additionally, the impact of wording in the instructions on the evaluation result was evaluated.

The analysis revealed a linear correlation between element size and success rates, with small elements achieving a success rate of only 30%, medium-sized elements reaching 54%, and large elements achieving 80%. Though with rising sizes, the cloud exploration focuses stronger around the element area, leaving parts of the cloud unseen. Additionally, positioning showed influence on success rates in some instances, but the dominating

factor is still size. The user evaluation showed no major influence from the element sizes. The results were primarily dependent on the reduction levels and stimuli.
Next, in the quantity analysis, users showed nearly identical performance when placing one (54%) or two (58%) elements into the cloud, but when placing three, the rate drops to 44%. The increase in quantity correlates with higher point cloud exploration, though the effect is mainly expressed when moving from one to two quantities. Moreover, the positional combinations of elements resulted in different success rates. User ratings revealed inconsistencies, which can be mainly attributed to low sample sizes, but the overall trend showed that, like in the size analysis, the main influence comes from the stimuli and reduction tested on ratings.
Finally, in the last condition, users got two different instructions. In one version, they were strongly advised on multiple occasions to focus on the quality ratings. In the other version, this emphasis was missing. This resulted in overall more positive ratings in the second condition (no emphasis) and more successful participants. Further differences weren't observed.
When bringing all the findings together, the best combination appears to be using no extra emphasis on quality ratings during instructions, combined with two medium-sized elements. This approach avoids pitfalls such as the low success rates of small elements, the use of three elements and the emphasis condition. In addition, it is benefiting from the increased exploration provided by the two-element condition, while avoiding having unseen areas in the large condition.
The implementation and methodology used demonstrated the feasibility of conducting a crowdsourcing-based study on QoE evaluation of point clouds.
In future work, the influence of different positional combinations on exploration behavior will be investigated. Additionally, comparing results with a study that lacks an incentive mechanism could provide further insights into the effect of such a mechanism. Furthermore, incorporating a more precise measurement tool for point cloud exploration may lead to a deeper understanding of exploration patterns. Finally, exploring alternative hiring and selection options for crowdsourcing participants could reveal their influence on rating and data reliability.

A. Appendix

A.1. Heatmaps for different Incentive Positions

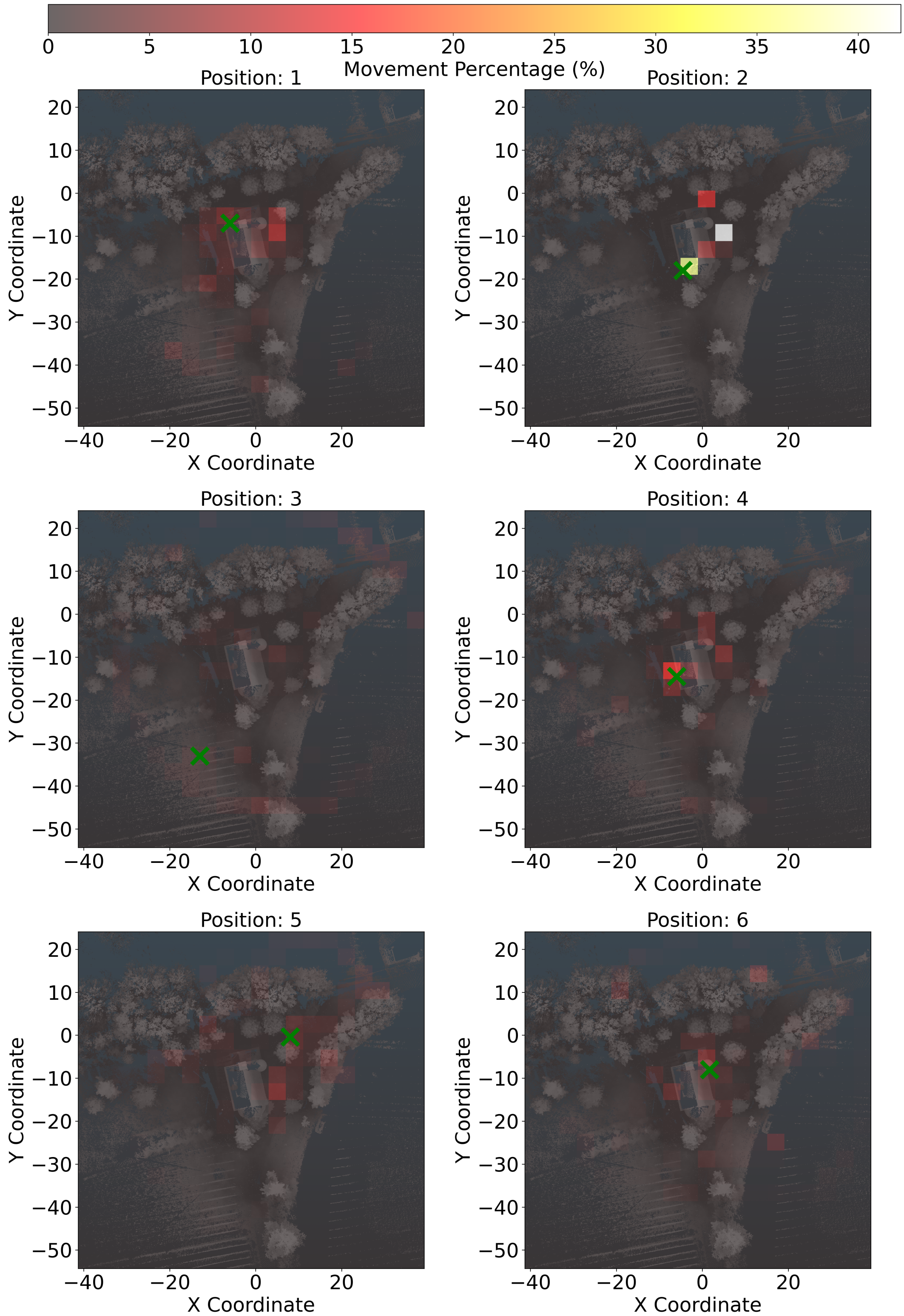

Figure A.1.: Heatmaps for different incentive positions Randersacker size *small*

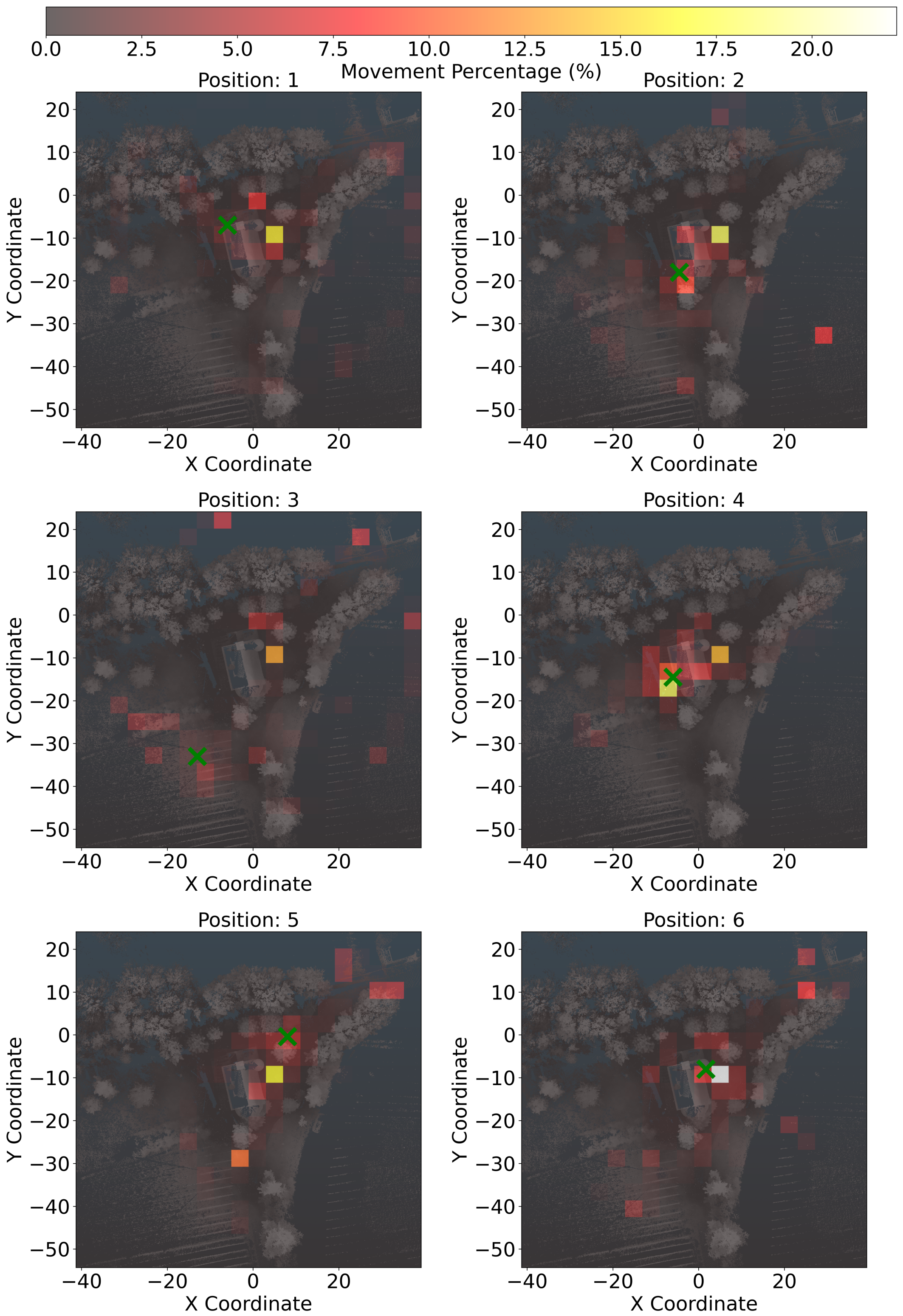

Figure A.2.: Heatmaps for different incentive positions Randersacker size *medium*

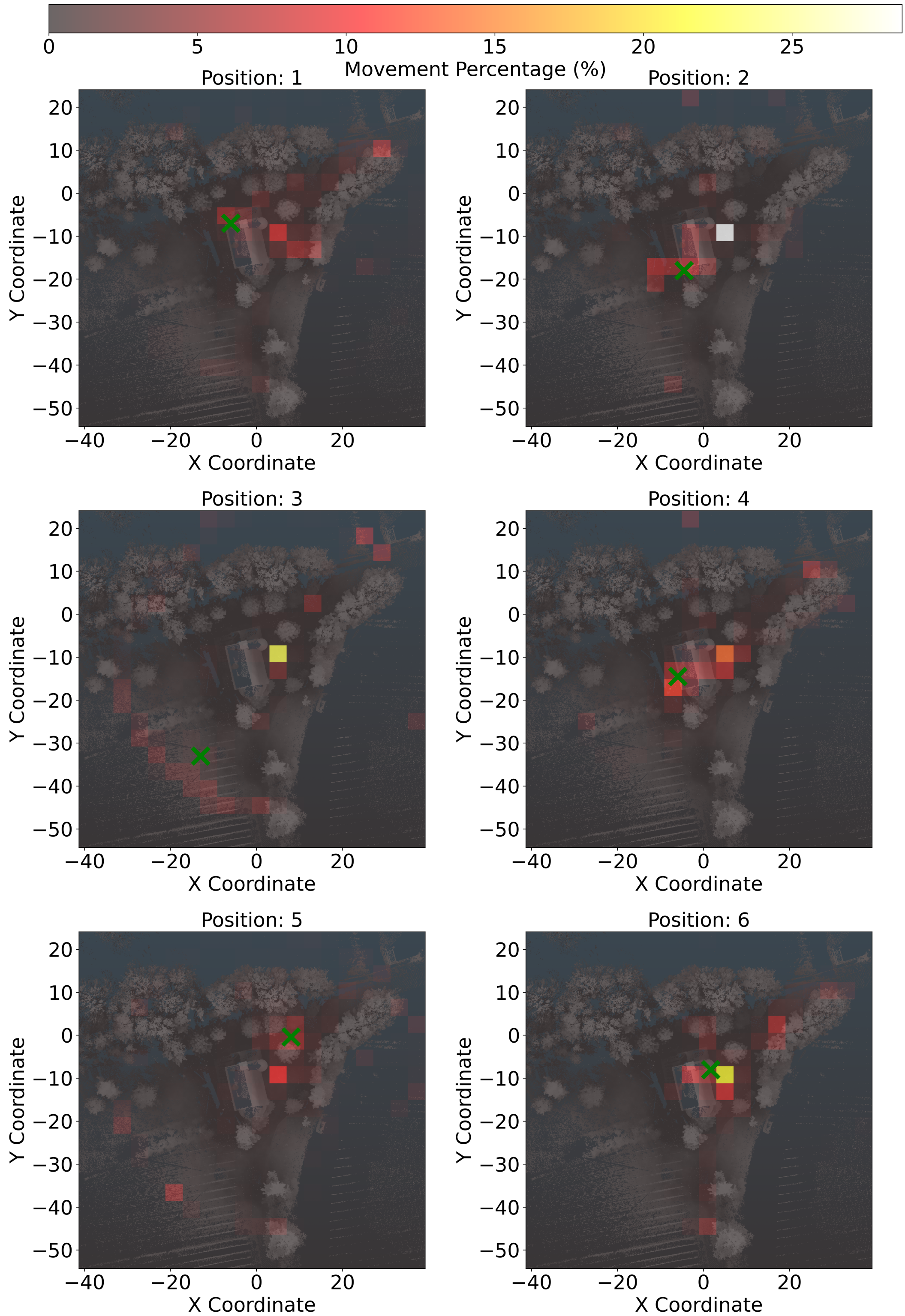

Figure A.3.: Heatmaps for different incentive positions Randersacker size *large*

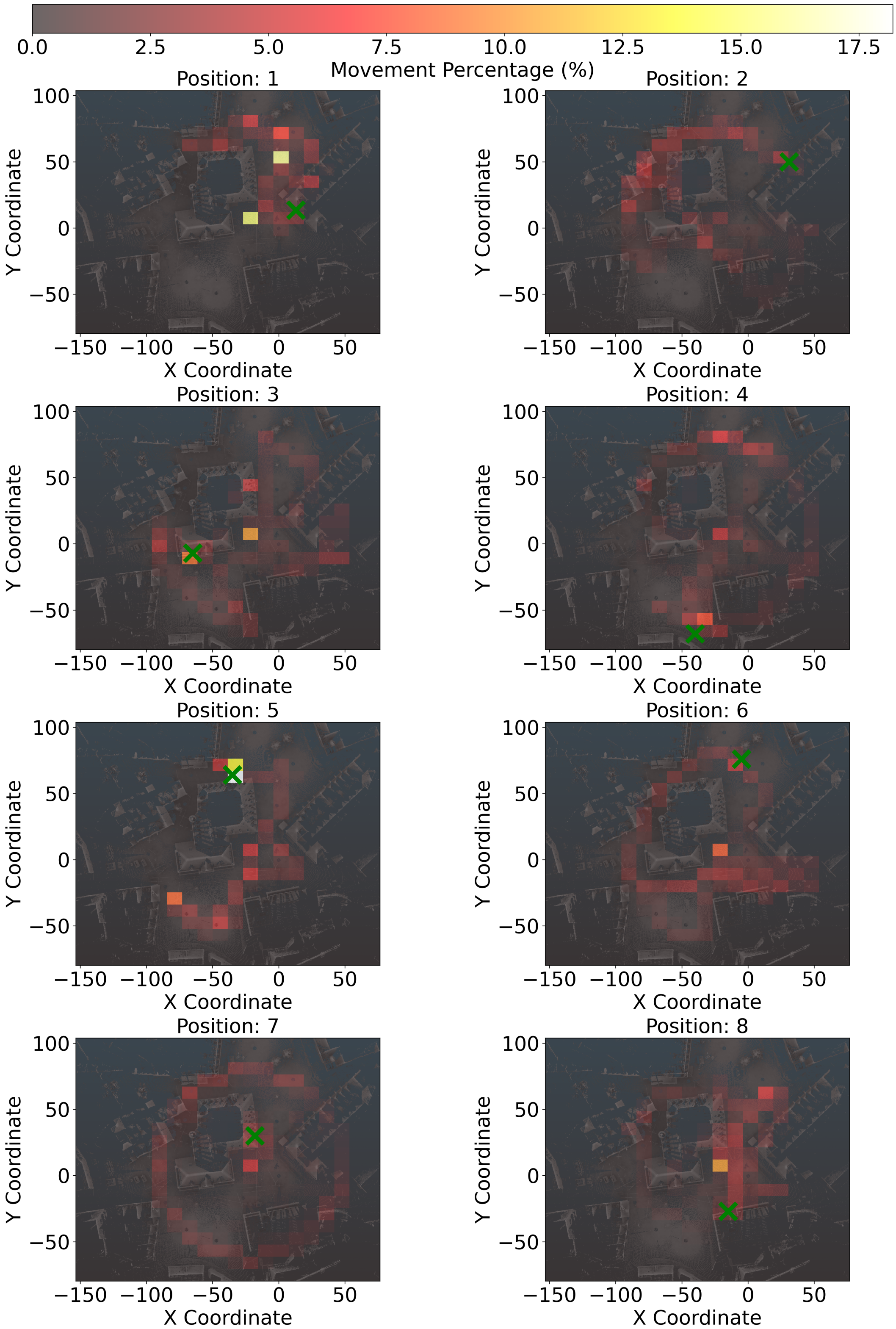

Figure A.4.: Heatmaps for different incentive positions Bremen size *small*

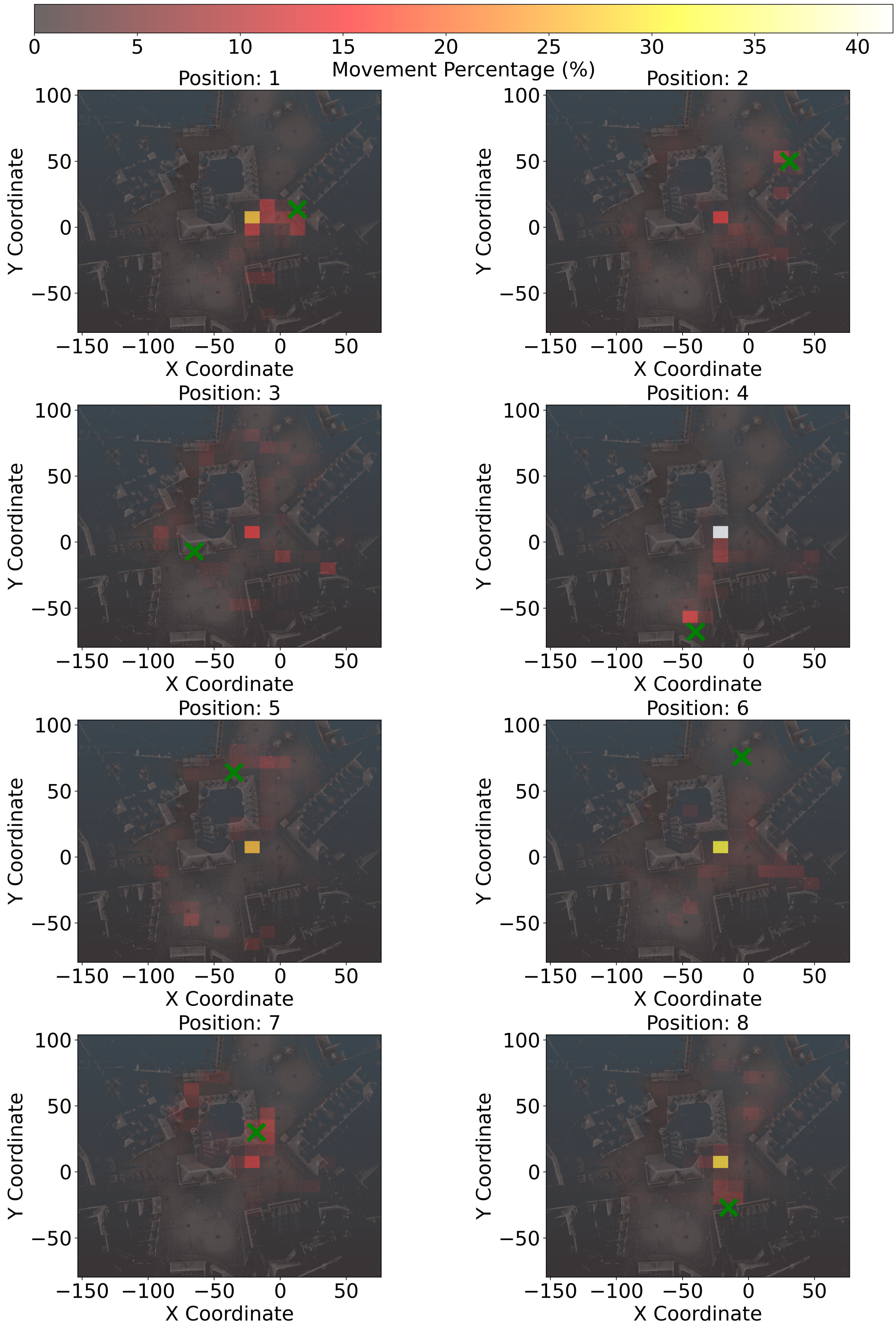

Figure A.5.: Heatmaps for different incentive positions Bremen size *medium*

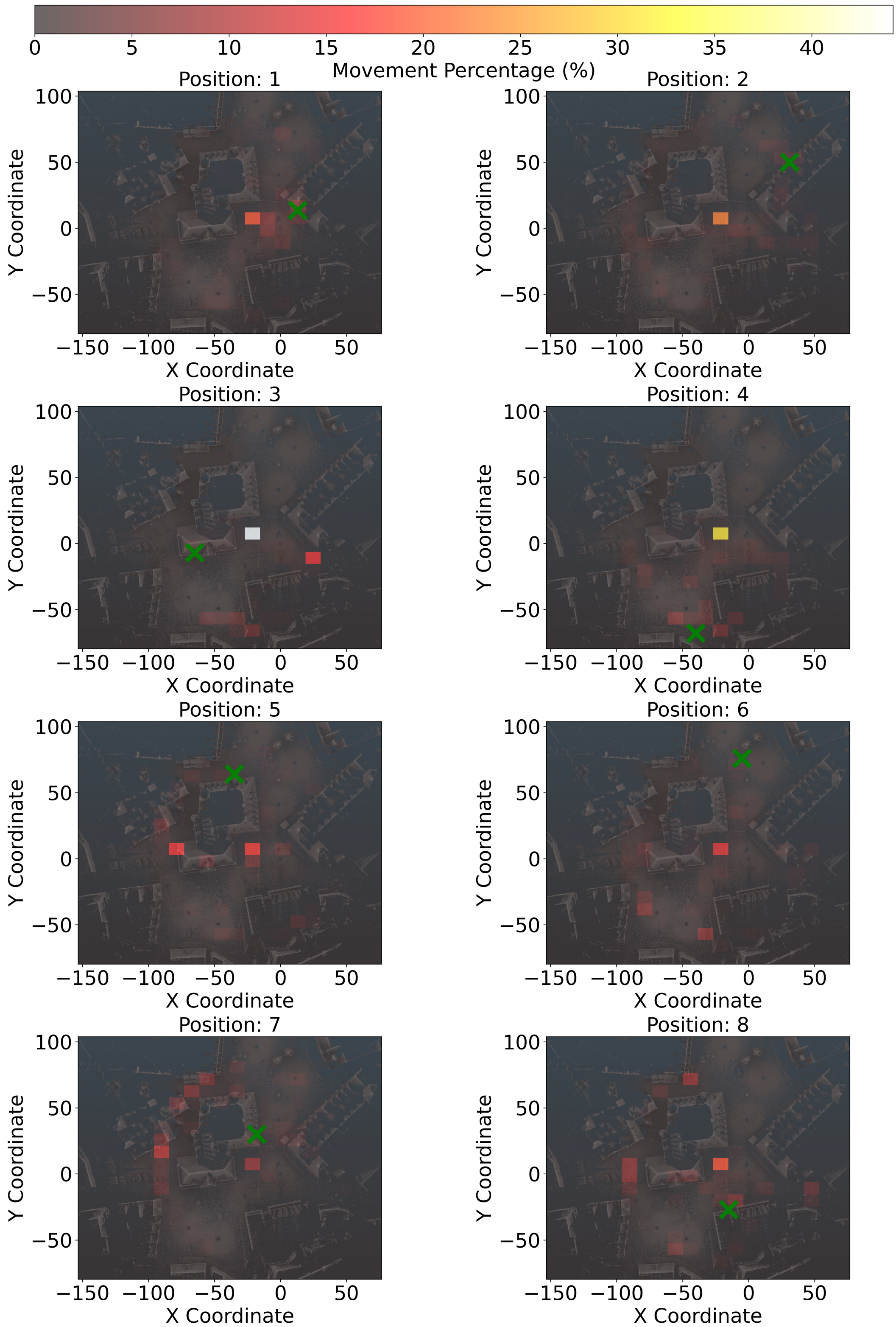

Figure A.6.: Heatmaps for different incentive positions Bremen size *large*

A.2. Positional Data per Permutation

Index	position	appearance	successfull	successrate
1	['pos 4', 'pos 6']	2	2	100.0
2	['pos 6', 'pos 3']	4	4	100.0
3	['pos 2', 'pos 4']	3	2	67.0
4	['pos 1', 'pos 3']	3	2	67.0
5	['pos 2', 'pos 3']	5	3	60.0
6	['pos 5', 'pos 3']	2	1	50.0
7	['pos 4', 'pos 3']	4	2	50.0
8	['pos 5', 'pos 4']	3	1	33.0
9	['pos 2', 'pos 5']	3	1	33.0
10	['pos 4', 'pos 1']	4	1	25.0
11	['pos 1', 'pos 5']	5	1	20.0
12	['pos 6', 'pos 2']	6	1	17.0
13	['pos 6', 'pos 1']	3	0	0.0
14	['pos 2', 'pos 1']	1	0	0.0

Figure A.7.: Positional data for each permutation Randersacker: quantity two

Index	position	appearance	successfull	successrate
1	['pos 1', 'pos 6', 'pos 2']	3	3	100.0
2	['pos 2', 'pos 6', 'pos 3']	2	2	100.0
3	['pos 3', 'pos 2', 'pos 4']	4	3	75.0
4	['pos 3', 'pos 4', 'pos 5']	3	2	67.0
5	['pos 5', 'pos 2', 'pos 6']	2	1	50.0
6	['pos 5', 'pos 4', 'pos 1']	2	1	50.0
7	['pos 5', 'pos 6', 'pos 1']	4	2	50.0
8	['pos 3', 'pos 2', 'pos 1']	3	1	33.0
9	['pos 4', 'pos 5', 'pos 6']	3	1	33.0
10	['pos 3', 'pos 4', 'pos 1']	2	0	0.0
11	['pos 3', 'pos 5', 'pos 2']	1	0	0.0
12	['pos 1', 'pos 4', 'pos 6']	1	0	0.0
13	['pos 3', 'pos 1', 'pos 6']	1	0	0.0
14	['pos 2', 'pos 4', 'pos 5']	1	0	0.0
15	['pos 3', 'pos 5', 'pos 1']	2	0	0.0
16	['pos 2', 'pos 6', 'pos 4']	2	0	0.0

Figure A.8.: Positional data for each permutation Randersacker: quantity three

Index	position	appearance	successfull	successrate
1	['pos 4', 'pos 6']	2	2	100.0
2	['pos 1', 'pos 7']	2	2	100.0
3	['pos 1', 'pos 3']	1	1	100.0
4	['pos 4', 'pos 2']	1	1	100.0
5	['pos 5', 'pos 8']	1	1	100.0
6	['pos 1', 'pos 4']	1	1	100.0
7	['pos 6', 'pos 7']	1	1	100.0
8	['pos 7', 'pos 5']	4	3	75.0
9	['pos 6', 'pos 3']	4	3	75.0
10	['pos 3', 'pos 4']	3	2	67.0
11	['pos 1', 'pos 6']	3	2	67.0
12	['pos 7', 'pos 4']	3	2	67.0
13	['pos 2', 'pos 5']	3	2	67.0
14	['pos 8', 'pos 1']	2	1	50.0
15	['pos 8', 'pos 6']	2	1	50.0
16	['pos 8', 'pos 4']	2	1	50.0
17	['pos 2', 'pos 3']	3	1	33.0
18	['pos 5', 'pos 6']	4	1	25.0
19	['pos 2', 'pos 1']	2	0	0.0
20	['pos 8', 'pos 2']	1	0	0.0
21	['pos 5', 'pos 1']	1	0	0.0
22	['pos 8', 'pos 3']	2	0	0.0

Figure A.9.: Positional data for each permutation Bremen: quantity two

Index	position	appearance	successfull	successrate
1	['pos 5', 'pos 2', 'pos 1']	1	1	100.0
2	['pos 4', 'pos 3', 'pos 6']	1	1	100.0
3	['pos 5', 'pos 1', 'pos 6']	1	1	100.0
4	['pos 1', 'pos 6', 'pos 2']	1	1	100.0
5	['pos 4', 'pos 3', 'pos 2']	1	1	100.0
6	['pos 6', 'pos 2', 'pos 8']	1	1	100.0
7	['pos 1', 'pos 3', 'pos 4']	1	1	100.0
8	['pos 6', 'pos 3', 'pos 1']	1	1	100.0
9	['pos 4', 'pos 1', 'pos 2']	1	1	100.0
10	['pos 8', 'pos 1', 'pos 2']	1	1	100.0
11	['pos 3', 'pos 7', 'pos 6']	3	2	67.0
12	['pos 5', 'pos 4', 'pos 8']	3	2	67.0
13	['pos 7', 'pos 8', 'pos 4']	2	1	50.0
14	['pos 1', 'pos 6', 'pos 7']	2	1	50.0
15	['pos 8', 'pos 3', 'pos 2']	2	1	50.0
16	['pos 4', 'pos 7', 'pos 5']	1	0	0.0
17	['pos 6', 'pos 7', 'pos 2']	3	0	0.0
18	['pos 4', 'pos 8', 'pos 6']	1	0	0.0
19	['pos 5', 'pos 1', 'pos 4']	1	0	0.0
20	['pos 3', 'pos 8', 'pos 5']	1	0	0.0
21	['pos 3', 'pos 1', 'pos 5']	1	0	0.0
22	['pos 2', 'pos 3', 'pos 1']	1	0	0.0
23	['pos 4', 'pos 3', 'pos 8']	1	0	0.0
24	['pos 7', 'pos 4', 'pos 2']	1	0	0.0
25	['pos 5', 'pos 8', 'pos 6']	2	0	0.0
26	['pos 8', 'pos 4', 'pos 2']	1	0	0.0

Figure A.10.: Positional data for each permutation Bremen: quantity three

Bibliography

[1] *U.S. VR and AR users 2017-2023*. en. URL: https://www.statista.com/statistics/1017008/united-states-vr-ar-users/ (visited on 09/25/2024).

[2] *Topic: Virtual reality (VR)*. en. URL: https://www.statista.com/topics/2532/virtual-reality-vr/ (visited on 09/25/2024).

[3] Nestor Gilbert. *74 Virtual Reality Statistics You Must Know in 2024: Adoption, Usage & Market Share*. en. Aug. 2019. URL: https://financesonline.com/virtual-reality-statistics/ (visited on 09/25/2024).

[4] Andrii Sydoruk. *Insightful VR Statistics for Entrepreneurs and Tech Enthusiasts*. en. Aug. 2024. URL: https://smarttek.solutions/blog/virtual-reality-statistics/ (visited on 09/25/2024).

[5] *What is LiDAR? — IBM*. en-us. July 2023. URL: https://www.ibm.com/topics/lidar (visited on 07/06/2024).

[6] Leihui Li, Riwei Wang, and Xuping Zhang. "A Tutorial Review on Point Cloud Registrations: Principle, Classification, Comparison, and Technology Challenges". en. In: *Mathematical Problems in Engineering* 2021.1 (2021), p. 9953910. ISSN: 1563-5147. DOI: 10.1155/2021/9953910. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/9953910 (visited on 07/06/2024).

[7] Tim Golla and Reinhard Klein. "Real-time point cloud compression". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015, pp. 5087–5092. DOI: 10.1109/IROS.2015.7354093. URL: https://ieeexplore.ieee.org/abstract/document/7354093 (visited on 07/07/2024).

[8] *3DTK - The 3D Toolkit*. URL: https://slam6d.sourceforge.io/ (visited on 07/25/2024).

[9] *CloudCompare - Open Source project*. URL: https://www.cloudcompare.org/ (visited on 07/25/2024).

[10] Kjell Brunnström et al. *Qualinet White Paper on Definitions of Quality of Experience*. Mar. 2013.

[11] Anand Inasu Chittilappilly, Lei Chen, and Sihem Amer-Yahia. "A Survey of General-Purpose Crowdsourcing Techniques". In: *IEEE Transactions on Knowledge and Data Engineering* 28.9 (Sept. 2016). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 2246–2266. ISSN: 1558-2191. DOI: 10.1109/TKDE.2016.2555805. URL: https://ieeexplore.ieee.org/abstract/document/7456302 (visited on 06/17/2024).

[12] Microworkers. *Templates*. en. URL: https://www.microworkers.com/ (visited on 07/25/2024).

[13] Joshua R. de Leeuw. "jsPsych: A JavaScript library for creating behavioral experiments in a Web browser". en. In: *Behavior Research Methods* 47.1 (Mar. 2015), pp. 1–12. ISSN: 1554-3528. DOI: 10.3758/s13428-014-0458-y. URL: https://doi.org/10.3758/s13428-014-0458-y (visited on 07/15/2024).

[14] Michael Seufert et al. "Different Points of View: Impact of 3D Point Cloud Reduction on QoE of Rendered Images". en. In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. Athlone, Ireland: IEEE, May 2020, pp. 1–6. ISBN: 978-1-72815-965-2. DOI: 10.1109/QoMEX48832.2020.9123143. URL: https://ieeexplore.ieee.org/document/9123143/ (visited on 04/23/2024).

[15] "On subjective and objective quality evaluation of point cloud geometry". In: *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)* (2017). Ed. by Evangelos Alexiou and Touradj Ebrahimi. ISBN: 9781538640241 Meeting Name: Ninth International Conference on Quality of Multimedia Experience (QoMEX) Num Pages: 3. DOI: 10.1109/QoMEX.2017.7965681.

[16] Juan Zhang et al. "A subjective quality evaluation for 3D point cloud models". In: *2014 International Conference on Audio, Language and Image Processing*. July 2014, pp. 827–831. DOI: 10.1109/ICALIP.2014.7009910. URL: https://ieeexplore.ieee.org/abstract/document/7009910 (visited on 04/23/2024).

[17] Jeroen van der Hooft et al. "Objective and Subjective QoE Evaluation for Adaptive Point Cloud Streaming". In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. ISSN: 2472-7814. May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123081. URL: https://ieeexplore.ieee.org/abstract/document/9123081 (visited on 04/19/2024).

[18] Emil Dumic and Luis A. da Silva Cruz. "Point Cloud Coding Solutions, Subjective Assessment and Objective Measures: A Case Study". en. In: *Symmetry* 12.12 (Dec. 2020). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, p. 1955. ISSN: 2073-8994. DOI: 10.3390/sym12121955. URL: https://www.mdpi.com/2073-8994/12/12/1955 (visited on 04/26/2024).

[19] Lisha Wang et al. "QoE-Driven Adaptive Streaming for Point Clouds". In: *IEEE Transactions on Multimedia* 25 (2023). Conference Name: IEEE Transactions on Multimedia, pp. 2543–2558. ISSN: 1941-0077. DOI: 10.1109/TMM.2022.3148585. URL: https://ieeexplore.ieee.org/abstract/document/9706255 (visited on 04/25/2024).

[20] Bo Han, Yu Liu, and Feng Qian. "ViVo: visibility-aware mobile volumetric video streaming". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13. ISBN: 978-1-4503-7085-1. DOI: 10.

1145/3372224.3380888. URL: https://dl.acm.org/doi/10.1145/3372224.3380888 (visited on 08/02/2024).

[21] Rufael Mekuria, Kees Blom, and Pablo Cesar. “Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.4 (Apr. 2017). Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, pp. 828–842. ISSN: 1558-2205. DOI: 10.1109/TCSVT.2016.2543039. URL: https://ieeexplore.ieee.org/abstract/document/7434610 (visited on 08/02/2024).

[22] *Robotic 3D Scan Repository*. URL: https://robotik.informatik.uni-wuerzburg.de/telematics/3dscans/ (visited on 08/14/2024).