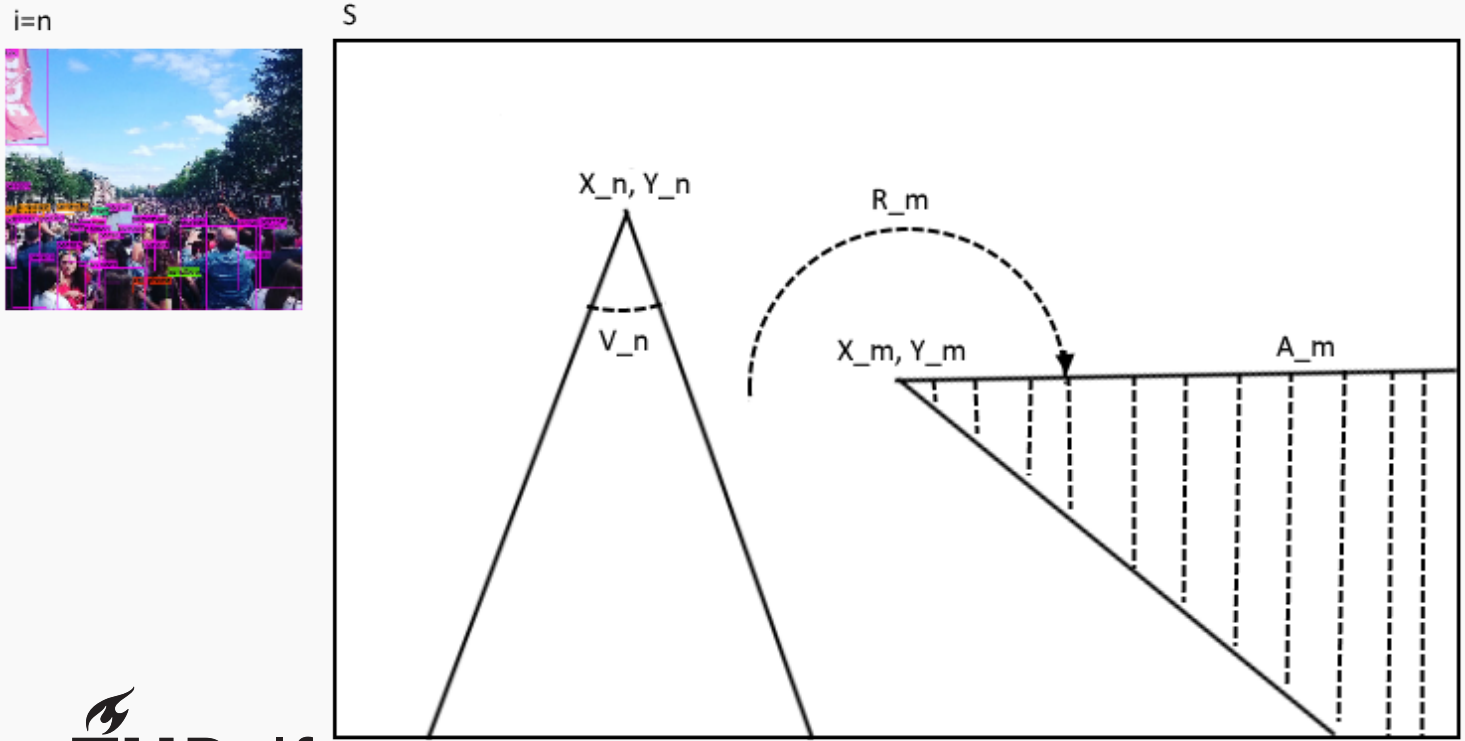
Master Thesis

Estimating crowd density and their emotions for city events using social media images.

Niels C. Bakker



TU Delft

Master Thesis

Estimating crowd density and their emotions for city events using social media images.

by

Niels C. Bakker

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on
Thursday December 17, 2020 at 14:00 PM.

Student number:	4161394	
Project duration:	Dec 11, 2019 – Dec 17, 2020	
Thesis committee:	Prof.dr.ir. Alessandro Bozzon,	TU Delft, supervisor
	Dr. Jie Yang,	TU Delft
	Dr.ir. Winnie Daamen,	TU Delft

An electronic version of this thesis is available at: http://repository.tudelft.nl/

TU Delft

Abstract

Event managers at large city events use crowd density as a metric in the process of maintaining safety at large-scale city events. Identifying the density of crowds at these events relies on expensive physical infrastructure to work well or have limited accuracy. We propose a method that addresses the gap of not relying on physical infrastructure while incorporating new data features that may help improve accuracy. In addition, Emotion Estimation may prove to be useful for an outlier detection system, such as stampedes. The gap addressed for Emotion Estimation is determining the distribution of emotions shown through facial expressions in social media images at city events.

To fill the gap, we propose a new density estimation method and analyze the distribution of emotions at events under normal circumstances. The proposed density estimation method does not rely on physical infrastructure. It estimates the density of crowds using social media images, the images' location, direction, the number of people in the image, and the image angle of view. We also propose a method to extract the images' location using Structure from Motion and Generalized Procrustes Analysis. We analyzed the location estimation through an experiment using manually gathered data. Density estimation was analyzed through an experiment using crowd simulation. Emotion estimation was done through social media images gathered at Kingsday and Sail in the Netherlands originally gathered by Gong et al. [15].

The results show that the location estimation method correctly determines 5% of images that are taken at the event area and 100% of images that are not taken at the event area. We found 5/7 results to be within 100 meters of their true location, according to our findings. For our density estimation technique, the proposed method outperformed other methods that do not rely on physical infrastructure for crowd densities of 0.1-0.3 (People/m^2) and social media activity of 0.01-0.03 (images/person). For our emotion estimation, we found that 25% of faces in social media images taken at the examined events were neutral, 70% were happy, and 5% were one of the following expressions: sad, surprised, and fear. No angry facial expressions were found.

Our research provides new approaches to calculate the location of social media images and estimate crowd density, which outperformed existing methods. Besides, we provide insights into emotions displayed in social media images taken at city events.

Preface

I would like to express my gratitude to my supervisors during this thesis. Alessandro Bozzon and Vincent Gong, without their feedback, tips, and advice I would not have been able to finish this thesis.

Furthermore, I would like to thank my parents, partner, brother, and sister for supporting me throughout my studies. Particularly I would like to call out to my partner, Susan de Boer, for being there for me during these trying times.

Finally, I would like to express my gratitude to André Veerman. Without his support in middle school, it would have been improbable that I would have entered university. Single-handedly demonstrating the impact a teacher can have by providing the right support at the right time.

Doing this thesis during the Covid-19 pandemic was not easy. Similarly, I can imagine all of our lives have been more complicated this past year. However, let us not forget that progress is still made and that these trying times too will come to pass. Until then, I wish everyone the strength to persevere.

Niels C. Bakker
Delft, the Netherlands,
December 2020

Contents

List of Figures

List of Tables

1 Introduction

This chapter introduces the background of this research, the research questions, contributions, and the outline of the structure for this thesis.

1.1. Background

City scale events used to happen and, despite the recent issues with the COVID-19 pandemic, are expected to happen again in the future on a regular basis. City-scale events are large events that take place in an urban setting. Examples of these in the Netherlands are King's Day and SAIL Amsterdam. Due to their scale, these events typically involve many stakeholders, such as the organizer, the municipality, national government, security, and the visitors themselves. All of these stakeholders have some interest in the proper functioning of the city-scale event. This can make crowd management a necessity[9, 38, 39]. Crowd management is especially necessary for large events[9]. Moreover, it usually involves monitoring and having plans to deal with crowd density, traffic generated by visitors, early arrivals, visibility of crowd management, and major incident planning, amongst other factors[9].

We will be focusing on *crowd density*, sometimes shortened to *density*, because we have identified a possible concept to estimate this differently from how this is usually done, with the benefit of not requiring physical infrastructure at the event, such as camera monitoring through *CCTV*, also known as video surveillance. The process of recording people and/or crowds through cameras. *CCTV* specifically is currently often used for this purpose. For example, it is recommended practice, for large events, by the Event Safety Alliance, a US-based group of amongst other event organisers[9].

Crowd density can be defined as the number of people present in a given area, divided by the size of that given area. It is given as (People/M^2) and is often used in pedestrian traffic flow theory[7, 8, 15]. It is one measurement that may need monitoring as it could be used to prevent stampedes, which are more likely to occur in high-density areas [22], or to indicate issues due to the corona pandemic[2]. Managing *crowd density* is considered good practice[9] and is, in some cases, obligatory to get a permit for being allowed to host city events [38, 39].

The stampedes mentioned above tend to be chaotic and dangerous events. Therefore, it would not be surprising to see emotions such as fear, surprise, and anger to be more common than during normal event times. It may be possible to detect this change in emotions if it exists, shown through social media images. However, it seems currently unknown what the distribution of the emotions shown by people in social media images is during normal event times.

For the necessary *crowd density* metrics, several commercial methods exist that allow *crowd density* estimation. Some of these are people counting systems located at all entrances and exits to count people, with the downside of requiring physical infrastructure present at the event such as counting machines[50].

WiFi measuring poles that measure WiFi signals entering and leaving its range[42]. This technique requires physical infrastructure present at the event. It may also be less accurate due to people not connecting to the WiFi and WiFi range being dependent on environmental variables likely to change over time. One final technique for estimating *crowd density* is detecting phones entering a phone network[3], which has the same problems as WiFi measurement techniques for density estimation. It also has the possible downside of mobile service providers not being allowed to share this data under the GDPR or similar laws. Methods also exist

that rely on the geodata connected to social media posts[15].

However, only a fraction of all social media posts have such geodata available. As is illustrated by 0.71% of English tweets having coordinate geodata as determined by Huang and Carley[23]. It would be of interest if a method could be devised to gather this geodata for a larger share of the social media posts. Having substantially more data available is likely to contribute to better results. Even if only because it would allow statistics to be gathered over more datapoints reducing uncertainties.

For estimating crowd emotions, surveys may be used, with the downside of only limited participation and comparatively high cost in addition to the possible introduction of selection bias. Other methods focus on the textual part of social media to estimate sentiment[13, 16, 26], which could be seen as related to emotions in that happiness is associated with positive sentiment and anger associated with negative sentiment. This research leaves social media images underutilized and a potential source for estimating emotions by extracting faces for social media images and classifying displayed emotions.

1.2. Research questions

These possible gaps lead us to propose the following main research question of the thesis.

MRQ: To what extent can social media images contribute to the estimation of the density and emotions of crowds during city events?

This research question is, however, quite large in scope. Therefore, it will be broken down into three sub-research questions to argue why answering this overall research question is useful and what contributions it would make. A visual overview of the idea behind these research questions can be found in Figure 1.1. While an overview of their relationships can be found in Figure 3.1

RQ1: How can the geographic location of a social media picture taken during a city-scale event be estimated?

One requirement in previous research, done by Gong et al.[15] was that the gathered social media data was geotagged. However, this reduces the data available substantially as only a small fraction of social media posts are geotagged, with only 0.71% of English tweets having coordinate data according to Huang and Carley[23]. Ideally, all social media images would have a location. It allows a much clearer image of the event's social media activity as this provides more data than is usually available, likely allowing a better estimation. This location estimation may be achievable using the social media image, as usually some part of the background is visible in these images. This work will employ Structure from Motion, a technique for combining 2d images into a 3d scene explained in more detail in subsection 3.2.1. This work will also employ a transformation function/matrix, which is a set of operations to convert between different coordinate systems, which is amongst other, used in computer graphics and is explained in more detail in subsection 3.2.2. This work will determine whether it is possible to construct a *Structure from Motion* 3d scene from these backgrounds. It uses images with known longitude and latitude to create a transformation function. This transformation function allows us to determine the social media images' location.

RQ2: How can images be used to estimate *crowd density* from social media without crowd measurement infrastructure?

For city events, it is interesting to know how many people there are in a given area. This amount helps determine how successful the event is. Furthermore, it might aid with the organization of the event itself by reducing potential risks such as stampedes[22]. It is also recommended practice by organizations such as the Event safety alliance[9] and may be necessary to receive a permit to organize the city event in the first place[38, 39]. Current methods rely on physical infrastructure, or lack accuracy as discussed in section 2.2. Thus, we will propose a methodology that considers specific features that may be extracted from social media images, such as the location, rotation, and the number of people counted in the image. We then analyze its accuracy using a crowd simulation.

RQ3: How can social media images be used to study the emotions expressed by participants of city-scale events?

While previous research has already been done on the sentiment of social media posts at city events[13, 16, 26], which could be seen as a generalization of emotions into positive emotions and negative emotions.

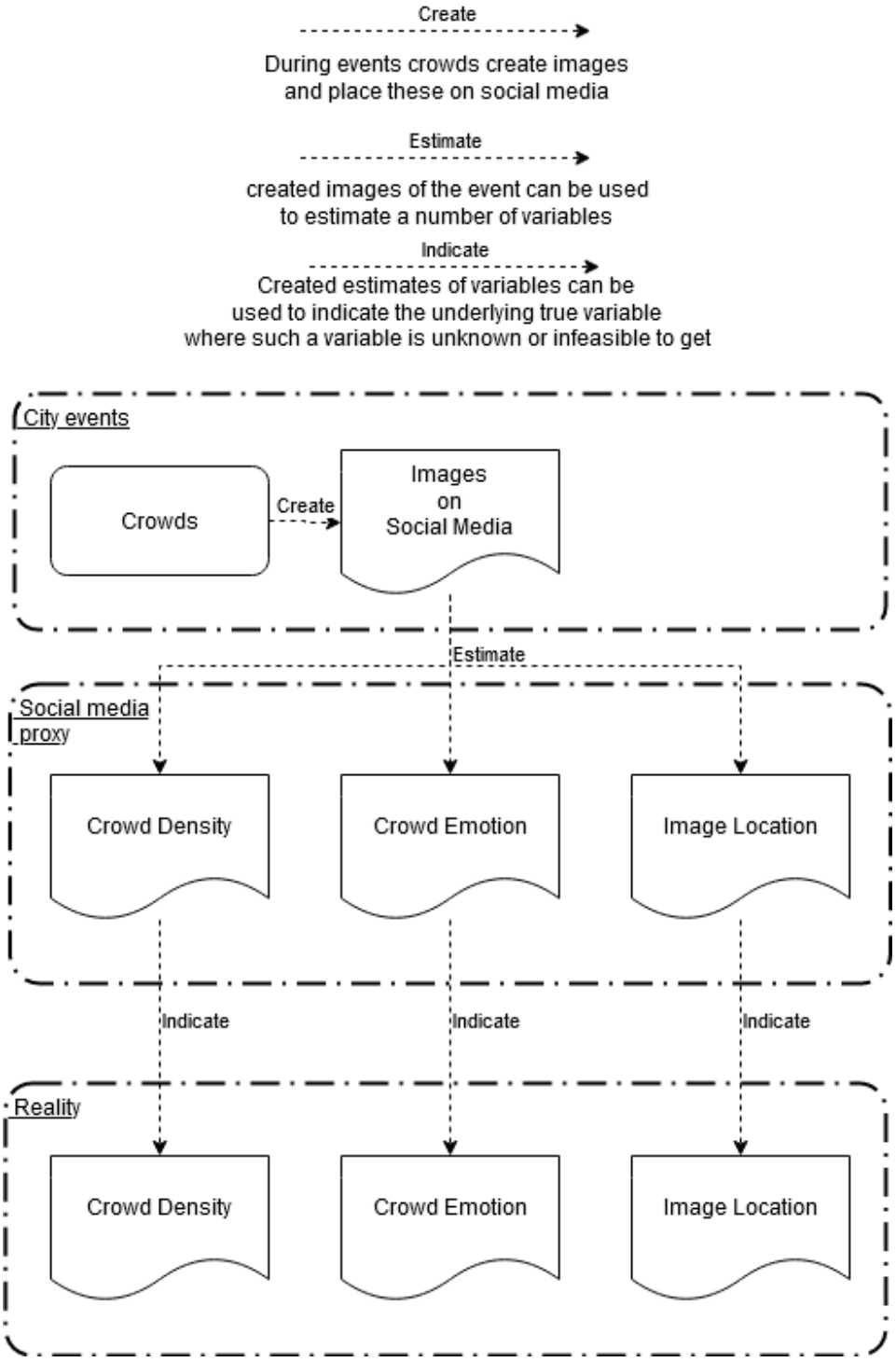


Figure 1.1: Visual overview of the relationship between the true variables of *crowd density*, crowd emotions and image location. And the estimates produced by social media images created by crowds in city events. Inspiration of visualization taken from [51]

Research has also been done on emotions in controlled settings[4]. It could be of additional insight to look into the underlying emotions of the people involved specifically for city events. As angry or sad emotions may be indicative of a problem requiring attention. Moreover, if not, it may still prove to be useful in evaluating the event. Our research will focus on the following facial expressions: neutral, angry, anxiety, fear, sadness, happiness, specifically, in the context of social media images taken at city events.

1.3. Original contributions

This paper will strive to make the following contributions.

- A method to automatically determine whether an image was taken within a predefined area or not. The method does not require data from a physical sensing infrastructure installed at the city event.
- A method to automatically determine the longitude and latitude of a location depicted in an image that is determined to be within a predefined area. The method does not rely on the physical infrastructure present at the city event. We shall refer to this method as the location estimation method.
- A method to automatically estimate *crowd density* for a city event that does not rely on physical infrastructure present at the city event. The method will rely on the following features: city event bounding box, image location, image direction, people counted in the image, and the image angle of view. Each feature is explained in more detail in section 3.1. We shall refer to this method as the density estimation method.
- An analysis of the effectiveness of our location estimation method in the context of Zuidplein, Sumatrakade, and De Dam. Which, amongst others, have been used to host the events of Kingsday, Sail, and a variety of protests, respectively. All three of these areas are located in Amsterdam.
- An analysis of the effectiveness of our density estimation method using a crowd simulation with respect to the features of true crowd density and the social media activity (the percentage of the crowd creating social media images).
- An analysis of emotions shown in facial expressions in social media images taken at city events. Specifically, the following emotions are considered: neutral, happy, sad, angry, surprised, and fear. For the social media images taken at city events, we will rely on data gathered by Gong et al.[16] for the events of Sail and Kingsday.

1.4. Outline

In this thesis, we will first examine how other research has approached the proposed research questions in chapter 2. In chapter 3, we will introduce the proposed methods by which the paper will address the research questions. In chapter 4 we will show how the experiments were conducted. chapter 5 will have the results of the conducted experiments. Furthermore, a discussion on the results can be found in chapter 6. Finally, a conclusion and proposals for future work can be found in chapter 7.

2

Related Work

In this chapter, related works are examined for each research question posed in section 1.2, i.e., location estimation using social media images, density estimation using social media images, and emotion estimation using social media images. We will briefly introduce the approach, results, and its relation to our research for each work. Finally, a summary shall be given, demonstrating the gaps our research will address.

2.1. Location estimation using social media images

The problem of location estimation from social media images can be formalized as: given a social media post containing an image, estimate the longitude and latitude of where that image has been taken. As far as can be determined, no research exists that focuses on this exact problem; however, research does exist that is similar to our research.

Chen et al.[5] in their work focus on estimating the location where a social media post was made using a social media post and the post history of the user involved. This location estimation is achieved by extracting the interest/theme, such as sports or entertainment, from the text part of the social media post. They also require a user to have at least one post with geodata and assume that a user does not stray more than a certain radius from the location given by the geodata, ever. They then use the interest taken from the post and then map this to *Points of Interest* with known geo-location and corresponding interest/theme, such as sports clubs or cinemas for sport and entertainment, respectively. They then assume that the post was made at the *Point of Interest* within the radius around the known geodata resulting in an estimated location where the post is made. For example, posting a social media post about sports (the interest) may be predicted to have made that post at the nearest football club (the point of interest). Their method has an error of 0.734, 0.8686, 2.397, 4.826 and 10.302 km for radii of 0.5, 1, 1.5, 2 and 3 km respectively. This error measurement has another drawback, however. As the post is matched to the location of a point of interest, only the locations of points of interest can be generated as estimates for social media post locations, which means that the error has a guaranteed minimum of the actual post location and the nearest point of interest. Finally, these points of interest are hard to be controlled for as they are features of the real world and, therefore, in limited supply. One can not increase the number of entertainment locations in the database arbitrarily by, for example, making more measurements but only by opening large amounts of, for example, cinemas or other points of interest. Meaning the resolution of the error can not simply be reduced by adding more data points in the form of Points of Interest. Their work differs from our research in several regards. First of all, their work only looks at the text part of social media, while ours will focus on the image part. Secondly, their work requires geotagged social media posts to work, which is sparse, while ours will not. Lastly, it is accurate to only half a kilometer.

Patwari et al. [42] propose a technique useable for detecting the location of phones. As phones are common with up to 81% of us citizens owning one[43]. This availability of smartphones makes it able to estimate the location of people by proxy as well. Their method uses geo-located WiFi base stations to locate mobile WiFi stations (such as phones) connected to the base stations. To do this, it relies on two features, the signal *Time of Arrival (TOA)* between devices and the *Received Signal Strength (RSS)*. Moreover, it assumes that pairwise measurements for all devices in the system are present. Based on this, they propose a 2d location estimation technique. They perform two experiments for 9x9m areas in a parking lot area and residential home with Root Mean Square (RMS) errors of 0.9-2.4 m and 1.0-2.7 meters, respectively. Their work differs

from our research because it requires physical infrastructure to be present in the form of WiFi stations, and their work requires users to connect to their self-hosted WiFi or have access to a WiFi infrastructure at the city event, while ours will not.

Hays and Efros [21] proposes a technique that estimates location for images. They achieve this by creating a geotagged image dataset with geographic keywords such as city names, country names, and popular tourist sites. After filtering out likely irrelevant images, they had 6,472,304 images. They calculated the following features for each of these images: Tiny images, Color histograms, Texton Histograms, Line features, Gist Descriptors, and the geometric context. Each of these features was then scaled to have a roughly equal standard deviation. The sum of these scaled features is then used for nearest-neighbor image comparison giving a location estimate. They also propose a clustering-based nearest-neighbor variant that takes the 120 nearest-neighbors and clusters them using mean-shift-clustering with a mean-shift bandwidth of 500 km. They then propose that the image is located at the largest cluster, giving a location estimate. Considering only the first choice is allowed, the single nearest-neighbor system performs better in achieving precise localization, with 15% being within 25km of the actual location. The mean-shift clustering technique performs better overall with an error of 1700km.

Li et al. [27] improves on the original research done by Hays and Efros[21]. It does this by improving computational time but, more importantly, when comparing to our work. They introduce the features of mean, standard deviation, and skewness of color moments for each color channel over five regions of the image, hierarchical wavelet packet descriptor, a method for representing the texture of an image as well as the Scale Invariant Feature Transform, more commonly known as SIFT, which allows for scale-invariant feature detection in images. They also propose a set of hierarchical structures to improve computation time and accuracy. They are using these additional features compared to Hays and Efros, resulting in city/geographic area classification accuracies of 97%, 91%, and 85% for the COREL5000, OxBuild5000, and GOLD datasets, respectively.

Our research differs from the examined related works regarding localization in several key ways. First of all, it does not need infrastructure as opposed to[42]. And it will focus on an area of accuracy measured in meters, not city or geographic area as opposed to [5, 21, 27].

2.2. Density estimation using social media images

Density estimation using social media images is formalized as: given a set of social media posts with images and an event area. Estimate the density of crowds at that event area during a given period. As far as can be determined, no research exists that focuses on our exact problem. However, research does exist that is similar to our research. In this context, research that focuses on counting people in images is explicitly included, as density can be derived by dividing the people counted by the area involved.

Gong et al. [15] focuses on density estimation using social media for city-scale events. They achieve this by introducing four different methods. *Geo-Based Density estimation*, a method that groups the sparse social media posts with geodata from the city event together into timeslots and calculates the density by taking the number of users that produced such a social media post and divide it by the area of the event. They also propose a modified version that considers a larger area than the event area to account for measurement errors in the location of the social media post. The second method proposed is *Speed-Based Density Estimation*, which counts users outside the event area with a probability function determined by pedestrian movement speed and the distance to the event area and posts made at the event. *Flow-Based Density Estimation* relies upon WiFi-sensors or counting systems at the event area boundary to determine the flow of users crossing it. Moreover, it can use this to determine the number of people present in the event area. For its results, it relies on ground data gathered for the event areas of Ruijterskade, Veemkade, Javakade, Sumatrakade en Zuidplein, all in Amsterdam. It finds a MAPE of 0.9879, 0.9588, 0.9705, 0.9344, and 0.9529 for these event areas, respectively, for *Speed-Based Density Estimation* the best-proposed method that does not rely on infrastructure. Moreover, it finds a MAPE of 0.8474, 0.8569, 0.5667, 0.7198, and 0.5235 for the event areas, respectively, for *Flow-Based Density Estimation* which does rely on WiFi-sensors or counting systems. Their work focuses on the post and ignores additional insights derivable from images commonly attached to social media posts. Their best performing method requires infrastructure to work, while the best performing method that does not require infrastructure performs worse than the method that does rely on infrastructure in every event area they examined. In our research, we aim to use the additional features that we can extract from the addition of social media images to improve on the results of the method that does not require infrastructure while still not requiring infrastructure such as WiFi-sensors or counting systems.

Rahmalan et al. [44] tries to estimate the density of crowds visible in images taken by a single camera with a fixed position. To achieve this, they introduce three different methods used for feature extraction. The first method is *Grey Level Dependency Matrix (GLDM)*, a method that looks at the joint probability of 2-pixel grey levels occurring in the image. The second method is *Minkowski Fractal Dimension (MFD)*, which is a measurement of the shape's roughness. Furthermore, thirdly, the *Invariant Orthonormal Chebyshev Moments (IOCM)*, which is a moment that does not change if a translation is applied. Each methods' features are then used to train a SOM classifier, a classifier that can do dimensionality reduction to allow high dimensional data to have easier visualizations. For the dataset, images from a camera with a fixed position above the crowd was used. The dataset's images were labeled Very low, Low, Moderate, High, or Very high in crowd density. They report their finding based on Morning, Afternoon, and Combined. For the combined result of both morning and afternoon, they find 80%, 40%, and 85% classification accuracy for the number of people present in the image for GLDM, MFD, and IOCM, respectively. Their work focuses on a single viewpoint as opposed to a city event terrain. Their images are from well-placed cameras (minimizing obfuscation). Thus infrastructure is required for this method.

Jiang et al. [24] are focused on counting people in images gained from fixed-position cameras. To do this, they propose a multi-layer convolutional neural network (MLCNN). This MLCNN is comprised of a single VGG16 body, a large neural network highly optimized for dealing with images. Three branches, one for each of the three final sub bodies, are added that break off from each of the three final sub bodies of the main VGG16 body to create a density map for each branch. The three generated density maps are then combined into a final density map used for counting people in the input image. This system results in an MAE of 9.94 on the WorlExpo'10 dataset. Their work focuses on fixed position cameras positioned to reduce the obfuscation of people as much as possible. In comparison, our work will be focused on social media images, which tend to be taken by people and, therefore, at the eye level. Furthermore, our work will be focused on crowd density over event areas with multiple viewpoints in the form of social media images as opposed to a single viewpoint in the form of a fixed camera. Finally, their work focuses on a single viewpoint, and therefore it can not have overlaps between viewpoints. In comparison, our work considers social media images as viewpoints. Therefore, our work will have to consider the overlap between these social media images to prevent the overcounting of people present in both overlapping images.

Liu et al. [28] are focussed on counting people in images gained from fixed-position cameras. They achieve this by combining three neural networks into one larger one. These three neural networks are RegNet, DetNet, and QualityNet. RegNet outputs a density map for each pixel in the input image. DetNet a head detection density map. QualityNet which allocates the qualities of RegNet and DetNet for each pixel. This QualityNet effectively results in a pixel specific weighting function on whether to use RegNet or DetNet. This method results in an MAE of 9.23 on the WorlExpo'10 dataset. Their work is different from our research because they only focus on counting people in an image, while our work will focus on density over an event area. This difference is important because it introduces overlapping viewpoints causing double counting of people, whereas their work only has one viewpoint and can not overlap. The images used in their works are taken from well-placed cameras, which minimizes obfuscation. That is why it is likely that physical infrastructure in the form of cameras may be necessary to achieve results similar to those in their work instead of being able to gather this through social media.

Our work will not require physical infrastructure present at the event to function as opposed to[24, 28, 44] and as opposed to the flow method proposed by Gong et al. [15]. Nor will it focus on a single viewpoint. Instead, the work will focus on an event area with multiple viewpoints as opposed to the research done by[24, 28, 44]. It will try to incorporate specific insights derivable from social media images; some introduced through methods proposed in this work, such as the number of people counted in the image, the location of the image, and the view direction of the image to improve the estimate as opposed to [15].

2.3. Emotion estimation using social media images

Emotion estimation using social media images is formalized as given a social media image, identify which emotions are shown by the people in that image. The emotions selected by a work can differ a bit from work to work but are generally some combination of neutral, anger, anxiety, fear, sadness, happiness, and surprise[4, 13, 26]. One closely related subject is sentiment estimation. Which only focuses on a combination of negative, positive, and neutral[16, 26] sentiments, which can be seen as individual emotions are usually strongly associated positively or negatively with happiness corresponding to positive sentiment and anger corresponding to negative sentiment, for example. These sentiments still provide insight into the state of

research. Therefore related works about sentiment are included as well.

Kumar et al. [26] propose 2 methods to estimate the sentiment and emotion respectively in social media text. For the used social media text dataset, 60195 tweets were gathered through the Twitter API for the hashtags #Google, #Microsoft, #Apple, and #Twitter. Their first method, used to classify sentiment, uses an unsupervised approach by assigning a score to each word present in the Lexicon based on the relation between the number of hits on google of "word in the lexicon" concatenated with "excellent" or "poor." it is used to classify sentiment and achieves an accuracy of 80.68%. The supervised method, which is used to classify emotions. Bag of Words, the set of words in the post, is used as features. A *Multinomial Naive Bayes (MNB)* classifier is used as the classifier. Moreover, they use a similar approach to gathering data as the first method. However, they use the emotions as hashtags and assume that people with these hashtags in their social media posts experience that emotion. Using this, they find an accuracy of 95.3% for classifying the emotions Anger, Fear, Joy, Love, Sad, Surprise, and Thankfulness. Their work is different from our research as it focuses on the text of a social media post while our focuses on the emotions of people shown in the image. This difference is important as the social media text is generated only by the poster, while the people visible in the image are likely to contain multiple people in the background, thereby providing much more insight into the emotions of the actual crowd. It also focuses on a highly specific subset of social media posts, those that self-report emotions shown, instead of specifically on social media posts originating from city events, which may have a very different distribution.

Gong et al. [16] measure the error of classifying sentiment for different methods on social media data for city events. To achieve this, they select multiple events Sail 2015, Kingsday 2016, europride 2017, and Feyenoord football riots 2017, which have different activities such as a boat parade, street parties, flea market, fireworks, and riots. These events were also in 2 different major cities in the Netherlands being Amsterdam and Rotterdam, respectively. The selected methods are mainly Lexicon, specifically SentiStrength and SentiWordNet, and Machine learning-based. Specifically, Naive Bayes, SGDClassifier, LinearSVC, NuSVC, and SVC are used. For the allowed sentiments, performance is measured for the scenario where neutral is allowed, and the scenario where neutral sentiment is not allowed, positive and negative being allowed in both scenarios. A common dataset is used, which is intended to reflect the true distribution of sentiment in social media posts. Furthermore, an Event-based dataset is used, which is intended to reflect the true distribution of sentiment in social media posts taken at city events. They find that all methods perform better if the neutral sentiment is excluded. Their lowest classification error for those that include neutral is 0.305 and uses LinearSVC and uses the event dataset for training. Their lowest classification error for those that do not include neutral is 0.177 and also uses LinearSVC and is trained on the event dataset. Their work is different from our research as it focuses on the text-based part of the social media post, not the image. They also focus on the sentiment, whereas we will be focusing on emotions expressed by people visible in the images.

Gajarla and Gupta [13] are interested in identifying the emotion associated with the image, not the emotions of subjects within the image. They achieve this by gathering data from Flickr queries for the emotions of Love, Happiness, Violence, Fear, and Sadness. They also analyze the sentiment by grouping the positive emotions Love, Happiness into positive sentiment and grouping the negative emotion Violence, Fear, Sadness into negative sentiment. They pass this data through a pre-trained VGG-ImageNet[47], a large neural network specialized in detecting features in images, and use the results of a specific layer as input to a Support Vector Machine (SVM), stripping neural network layers and inputting these into an SVM are also known as *word2vec* and used to encode linguistic or more generally context-aware properties. For the SVM, a One vs. All in the context of the selected emotions is used. They also perform several finetuning steps that have comparatively minor effects on the overall results. For the sentiment classification, they find an accuracy of 0.678, and for the emotion classification, they find an accuracy of 0.384. Their work is different from our research because it is not focused on the emotion or sentiment shown by people in the image. Instead, it focuses on the emotion evoked by the viewer of the image. It also does not focus on images taken at city events while ours will.

Chakraborty et al. [4] is focused on detecting the emotions displayed in faces. To achieve this, they create a dataset of front-facing faces in a lab setting. For each of the faces recorded, the person whose face was recorded was shown a clip chosen specifically to evoke one of the emotions chosen from Anxiety, Disgust, Fear, Happiness, and Sadness. The face image is segmented into three regions based on features specific to that part of the face. These face segments are the mouth, the eyes, and the eyebrows. These are then labeled into three fuzzy sets eye-opening, mouth opening, and eyebrow constriction, each with values of LOW, MODERATE, and HIGH. Using these features for the faces, a fuzzy relational model is constructed, which generates a probabilistic system that a specific emotion is shown. They find that their system predicts

correct emotions 88.2% of the time for males, 92.2% of the time for females, and 96.6% of the time for children.

Their work is different from our research because it was done in a highly controlled lab environment. In comparison, ours is focused on emotions shown in social media images at city events. We will also only be focusing on the distribution of emotions while they focus on classification.

Our work will be focused on emotions shown at city events as opposed to [4, 13, 26]. It will focus on the emotions displayed as opposed to [4]. Finally, our work will focus on emotions shown by subjects in images as opposed to the social media text generated by the social media user[16, 26].

2.4. Summary

This section will highlight the main differences from the discussed related works for each discussed research question.

2.4.1. Location estimation

Our research differs from the examined related works regarding localization in several key ways. First of all, it does not need physical infrastructure such as WiFi infrastructure[42], as opposed to[42]. And it will focus on an area of accuracy measured in meters, not city or geographic area as opposed to [5, 21, 27].

2.4.2. Density estimation

Our work will not require physical infrastructure present at the event to function as opposed to[24, 28, 44] and opposed to the flow method proposed by Gong et al. [15]. Nor will it focus on a single viewpoint. Instead, the work will focus on an event area with multiple viewpoints as opposed to the research done by[24, 28, 44]. Our work will incorporate specific insights derivable from social media images; some introduced through methods proposed in this work, such as the number of people counted in the image, the location of the image, and the view direction of the image to improve the estimate as opposed to [15].

2.4.3. Emotion estimation

Our work will be focused on emotions shown at city events as opposed to [4, 13, 26]. Our work will focus on the emotions displayed as opposed to [4]. Finally, our work will focus on emotions shown by subjects in images as opposed to the social media text generated by the social media user[16, 26].

3

Methodology

In this section, we first describe the methodology framework for investigating the main research question, i.e., using social media images to estimate crowds' density and sentiment in city events. Further, for each component in the framework – connected to a research sub-question, we introduce the related method. Namely: estimating social media images' location using *Structure from Motion (SfM)*, estimating the density and emotion of crowds using social media images.

3.1. Methodology framework

Our method for estimating the density of crowds shown by RQ2 in Figure 3.1 and proposed in section 3.3 is dependent on social media images being able to provide the following features. The timestamp when the image was taken, the number of people seen in the image, the angle of view of the image, the location the image was taken, and the direction in which the image was taken.

While some of the discussed related works demonstrate the ability to count people in a single image[24, 28, 44] or extract faces from images through tools such as YOLO[45] and can therefore be deemed extractable from social media images. Moreover, the timestamp is provided with the social media post. The angle of view can be estimated, assuming that most social media images are made with a smartphone, are usually limited to a small range of about 62-82 degrees, excluding specialty phones. Therefore the Angle of View can be approximated by simply taking the halfway point at 72 degrees.

The location is, in certain cases, also available for social media images in the form of geodata attached to the image. However, as shown in subsection 4.1.1, these locations may not represent the actual location the image was taken. Moreover, they are quite sparse compared to the overall amount of social media posts. To overcome these shortcomings of existing location data available to social media posts/images as well as increase the overall availability of location data, we will introduce a method to estimate the location demonstrated by RQ1 in Figure 3.1 and given in subsection 3.2.4.

The required image direction can be achieved through similar methods as subsection 3.2.4 by only taking the rotation element introduced in subsection 3.2.2 and applying it to the image direction extractable from subsection 3.2.1. The image direction was not closely examined in this work due to its direct relation to the location estimation as illustrated by Figure 3.2.

Finally, we will also be looking at the distribution of emotions shown in social media images at city events shown by RQ3 in Figure 3.1 the method of which is given in section 3.4. We will be focusing on the expected distribution of emotions in faces in social media images taken at city events.

3.2. Estimate location of social media image using *Structure from Motion*

In this work, we design and evaluate a method based on *Structure from Motion (SfM)* and *Generalized Procrustes Analysis* to determine the location of a social media image within a certain area or, if it is not from that area, reject it. To explain this, we first introduce the *Structure from Motion* and further describe the *Generalized Procrustes Analysis*, which uses the outputs of the *SfM* to estimate the location of where the social media images were taken. Both are shown as the green boxes in Figure 3.3.

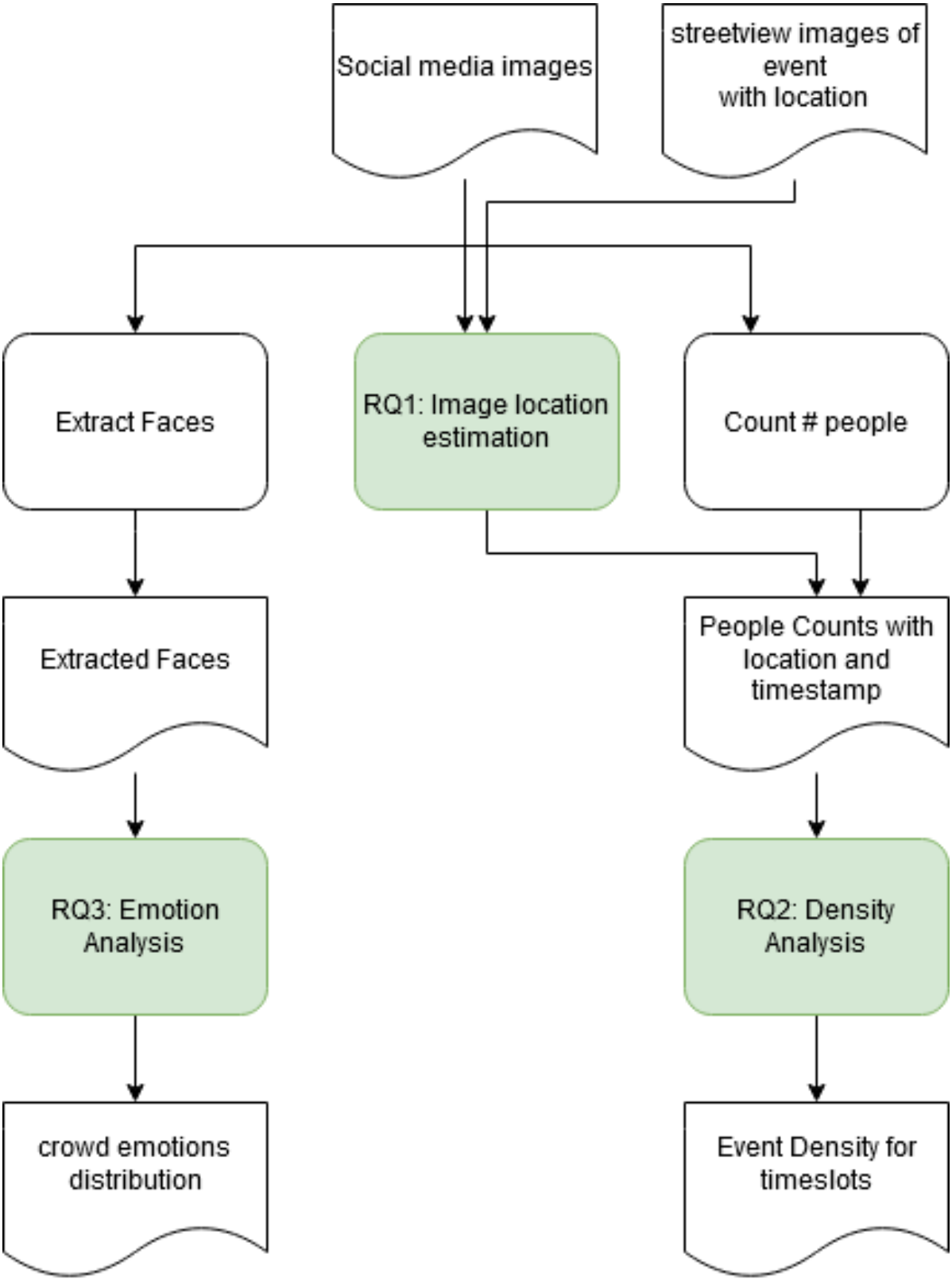


Figure 3.1: The methodology of estimating the density and emotions of crowds from social media images at city events. Green squares denote Research questions that are introduced. Rq 1 rq2 and Rq 3 have a methodology for investigating sub research questions in the following subsections.

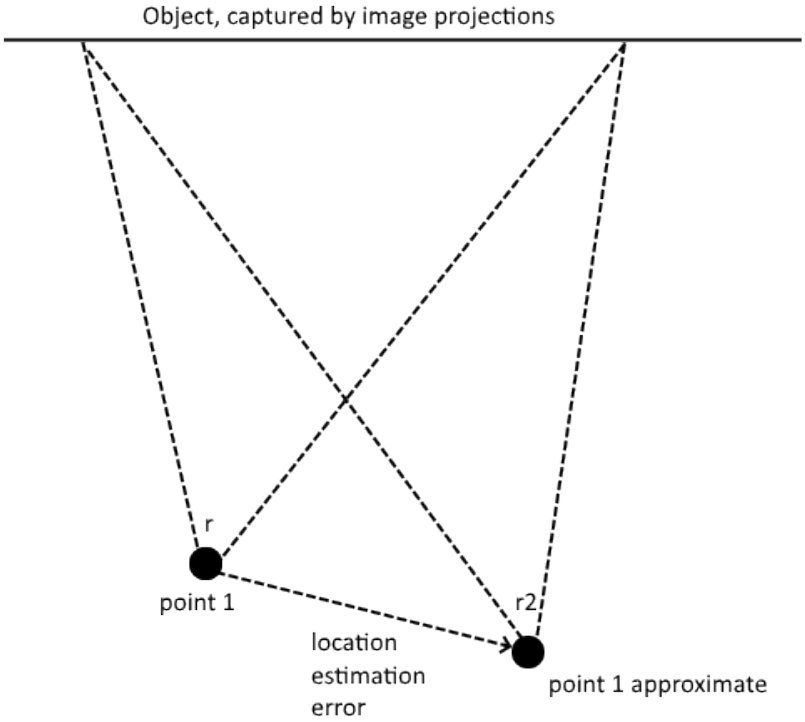


Figure 3.2: As can be seen introducing an location estimation error between *point 1* and *point 1 approximate* limits the rotation. Because the estimated location given by *point 1 approximate* still needs to see approximately the same scene as the original point 1. Therefor the rotation error is largely determined by the location estimation error. The reason why the scene has to stay more or less consistent is a property of how Structure from Motion systems function and explained in more detail in subsection 3.2.1.

3.2.1. *Structure from motion*

The first step in the proposed method of location estimation makes use of the *Structure from Motion (SfM)* approach.

Structure from Motion (SfM) is a process of creating a 3d scene reconstruction using 2d images by looking at overlapping features in the 2d images. It then combines these overlapping features and maps them to a 3d X, Y, Z coordinate space. This X, Y, Z coordinate space is the consensus relative distance space determined by the SfM. It is the coordinate space for which the lowest error could be found based on all the images used by the SfM. The Y dimension is height. The X, Z plane is in the same plane as longitude and latitude, assuming we ignore that the earth is curved.

This aligning of X, Z with Longitude and Latitude is only approximately true for smaller areas but should be sufficient for event areas, the area covered by the city event. This approximation is because the influence of the earth's curvature over a 1 km distance is about a 0.1 m deviation assuming the earth is round and has a radius of 6378 km, which is the radius given by NASA[34]. In practice, calculating distances on the planet involves many other complications, such as the planet not being a perfect sphere but ellipsoid and features such as mountains causing local variations. However, for city events where the event area is flat, these should not cause substantial deviations as they are either not relevant to city event scales. Or features of the earth not usually present at event areas within cities, such as mountains.

This X, Y, Z space is also called the 3d reconstruction of the Structure from Motion system of the 2d images or the reconstruction for short. This reconstruction also allows placing the 2d image in the X, Y, Z coordinate system, giving us the best estimate for where the image was taken within the X, Y, Z coordinate system.

An example of such a Structure from Motion reconstruction, using the opensfm tool, is shown in Figure 3.4. Together with an image that was, together with other images, used for its reconstruction is found in Figure 3.5. Please note how both have the palace visible in their bounding box.

Structure from Motion systems relies on several steps to derive the 3d reconstruction from a set of images. While different specific Structure from Motion systems may use slightly different steps to optimize for their specific problem, these steps should generally hold between different Structure from Motion systems.

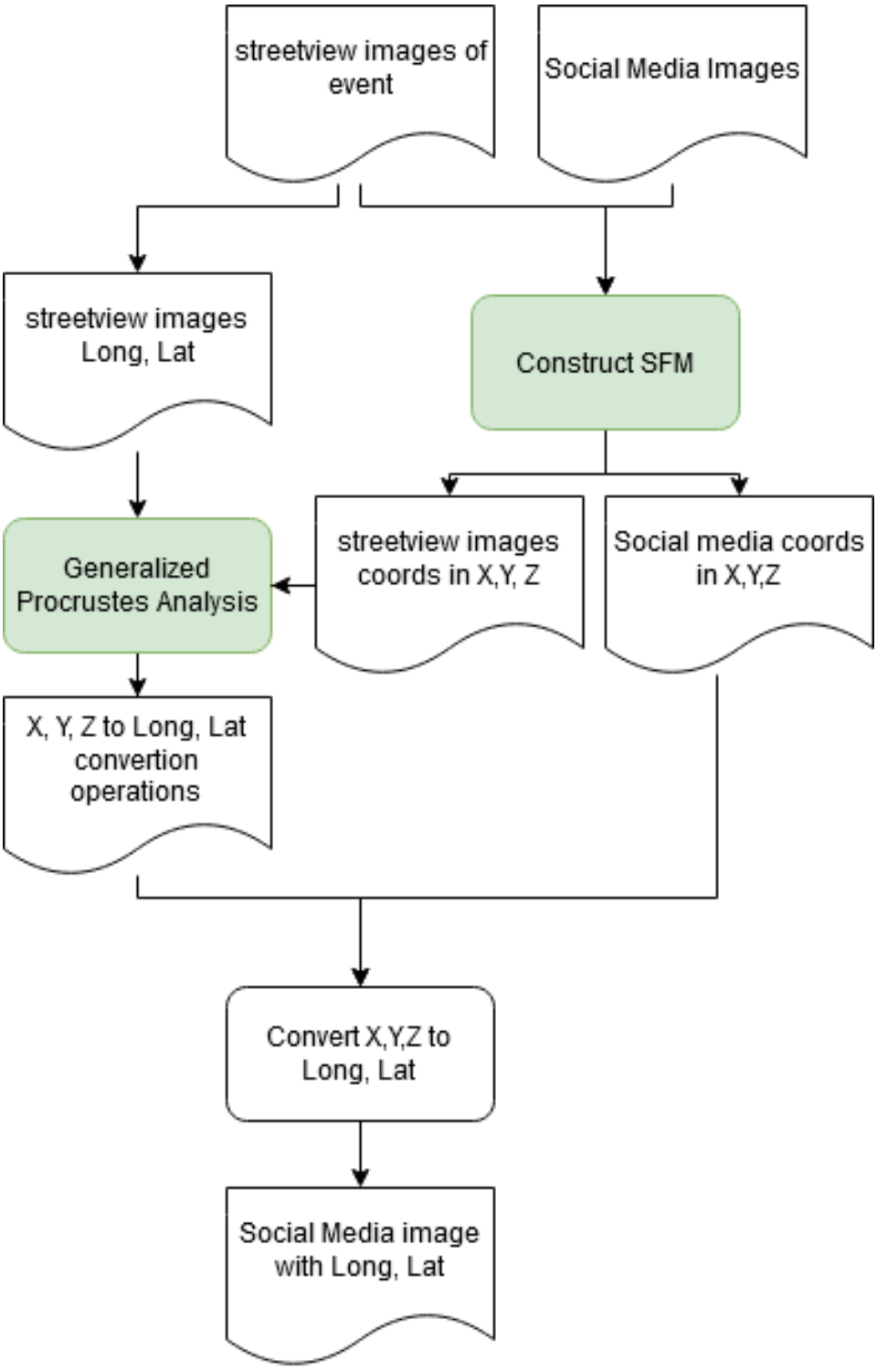


Figure 3.3: High level overview of the steps involved in achieving a longitude and latitude location estimation for a social media image suspected to be taken at the city event.

The first step in this process is to extract feature points from each image. These feature points are generally located at a specific point in the image but can aggregate information over the entire image in itself. These feature points are specifically selected so that if the same object is shown in different images, the feature points are the same. As different images usually vary in factors such as rotation, scaling, angle of view, illumination, and other factors. The methods that extract these feature points usually provide ways to handle (a specific subset of) these factors. Examples of methods that provide these feature points are *Scale Invariant Feature Transform (SIFT)*[29], *Hessian Affine Transformation with Histogram of Oriented Gradients* (HAHOG)[30] and *Global Image Features (GIST)*[40]. SIFT provides translation, scaling, and rotation insensitive feature points and can handle some changes in lighting. In comparison, GIST's feature points are invariant to scale and rotation and can deal well with affine distortion, change in viewpoint, noise, and illumination.

The second step is to match the features found in images in the first step between different pairs of images. To determine if an overlap, as illustrated by Figure 3.9, between these images exists and, if so, what that overlap is and how likely it is that this overlap is genuine. Due to the many features and noise present in the features between images, approximate matching is often applied. An example technique used for this is *Fast Library for Approximate Nearest Neighbour (FLANN)*[32]. In general, the feature matching step assumes that features are more or less in the same real world position in all images. This makes Structure from Motion well suited to, for example, urban environments and less suited to environments where shapes continuously change, such as forests.

The third step is to determine the feature points present for multiple images. For example, if multiple images capture the same door. Usually, a certain minimum amount of images should agree on the same feature point to limit bad matches. These sequences of images with agreed-upon feature points are called tracks. It should then be expected that the features generated for that door are present in each of these images. Therefore, a track should be present for each of these feature points with the images that capture the door.

The fourth step is to do the actual 3d reconstruction. This reconstruction is achieved by trying to estimate the current disagreement of the model. There is a certain amount of disagreement for any specific combination of values of feature point positions and image positions. One feature point may indicate that the image was taken at one location, while another feature point may indicate that that same image was taken somewhere else. If one were to sum the overall disagreement within a specific combination of values for feature point and image positions in the 3d X, Y, Z space, one would have an error metric. The lower the disagreement in positions, the more accurate the model becomes. An error of 0 implies that the model is a good reconstruction based on what is seen in the images. An error of 0 is unlikely to occur in real-world applications due to measurement noise, amongst others. An example of a technique used for optimizing the total model disagreement is *Bundle Adjustment (BA)*, which, based upon a set of initial estimates, can lower the overall model disagreement.

As mentioned in the description of steps for a Structure from Motion system, one important step in the process is detecting overlaps between images if no such overlaps exist. Alternatively, if only part of the images overlaps, then no reconstruction can be made, or the reconstruction will only involve the image set that do have overlap. For example, given two images. One taken in New York and 1 taken in London. Deriving a 3d structure from the combined images is impossible as they share no information and therefore allow no triangulation between feature points in the images. To avoid this issue of being unable to reconstruct or creating a wrong reconstruction, we seed the *Structure from Motion* system with source images with known longitude and latitude. These images should be of sufficient number so that the entire event area is covered, and enough overlap exists between them to ensure a successful Structure from Motion reconstruction.

For each image whose location is to be determined, we will do a reconstruction with the seed images and the image whose location is to be determined. The image whose location is to be determined may or may not be taken at the event area. It is up to the Structure from Motion method to determine whether this is the case. In this research, as proposed in subsection 4.2.1, we will be focusing on social media images from different city events. Though if random, non-adversarial, images are provided because it relies on these feature points for its initial modeling. These random images are expected to be mostly not to be included in the reconstruction. Moreover, even if random images are included by coincidence or accident. These random images are still likely to be thrown out as they are unlikely to form long enough tracks to be kept for the initial modeling, further reducing the likelihood of including images in the reconstruction that are not taken at the actual event area.

The *Structure from Motion* system then produces an X, Y, Z coordinate value for all images in the reconstruction. If the image was not in the reconstruction, the image does not have coordinate values.

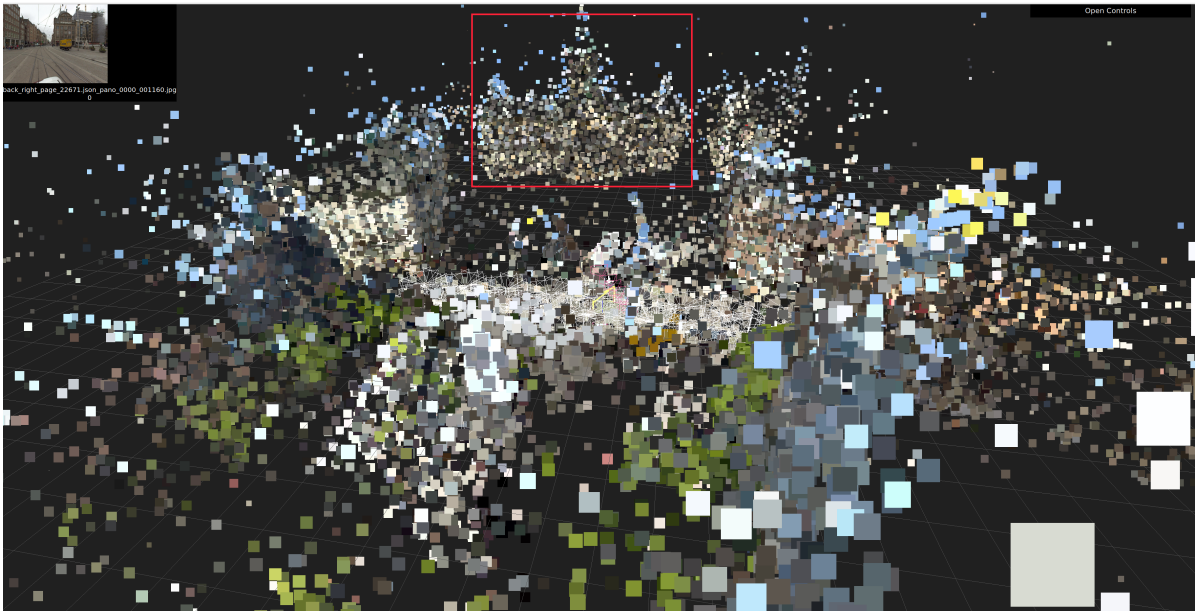


Figure 3.4: Structure from Motion reconstruction using Opensfm[30] with default settings of "De Dam" in Amsterdam using panorama photos. The image in the upper left corner is the currently selected image location. The palace on the dam is contained by the red bounding box.

It should, however, be mentioned that Structure from Motion reconstruction is highly computationally expensive. The system scales quadratically for a single reconstruction in the number of input images, as explained in subsection 7.2.1. Moreover, the constant time of each input is quite high (in the order of up to several seconds for a single input image) Even for comparatively small event areas with 100 images or 1000 images, this can mean 10000 or 1000000 seconds of computing time assuming a constant compute time of 1 second per input in the worst case. This computing expense is also the reason much research is focused on reducing the computation time of Structure from Motion systems, as demonstrated by Agarwal et al.[1] and Frahm et al. where Agarwal et al. is focused on achieving (part of) city-scale Structure from Motion. While Agarwal et al. is focused on the same problem while reducing the resources to fit within a single computer.

3.2.2. *Generalized Procrustes Analysis*

Generalized Procrustes is a process aimed at aligning coordinate systems using rotation, scaling, and translating. As shown in Figure 3.6.

As shown in Figure 3.3, one step in the process requires matching the created X, Y, Z coordinate system to longitude latitude. In this context, we will ignore the Y coordinate as it is almost constant (being at roughly eye height) compared to X, Z. We also assume that the longitude latitudes involved are city scale. Meaning the curvature of the earth can be ignored as discussed in subsection 3.2.1. So that longitude and latitude can be considered a linear coordinate system. This assumption simplifies the problem to matching points in an X, Z coordinate system to points in a longitude-latitude coordinate system. With the social media images included in the reconstruction, which only have coordinates in the X, Z coordinate system need to have their longitude and latitude be predicted.

Generalized Procrustes Analysis[19] is capable of converting between Cartesian coordinate systems as we require and guarantees the minimal achievable *Root Mean Squared Error* in the scope of the following operations: translation, scaling, and rotation for this conversion. *Generalized Procrustes Analysis* produces a set of translation, scaling, and rotation operations that transform a coordinate in X, Z to longitude and latitude. These types of transformation operations are examples of affine transformations.

3.2.3. Outlier Corrected Generalized Procrustes

One potential risk in using Generalized Procrustes Analysis is that it tries to optimize its selected affine transformations using Root Mean Squares Error. This error measurement has the unfortunate downside that if a single coordinate has a large error in the measurement, Procrustes' real result is disproportionately influenced by that measurement. For example, if we have the 3 points given in Figure 3.6 and introduce a single

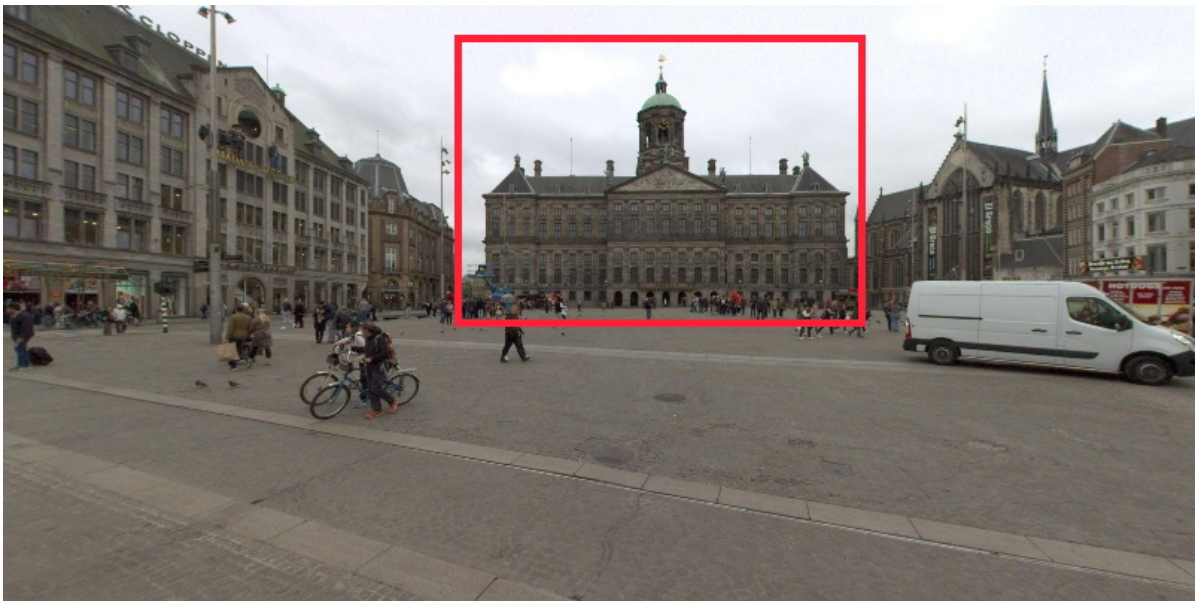


Figure 3.5: Example of image extracted from panoramas provided by the Amsterdam open dataset for the Dam in Amsterdam. The palace on the dam is visible in the center of the image, and also surrounded by a red bounding box.

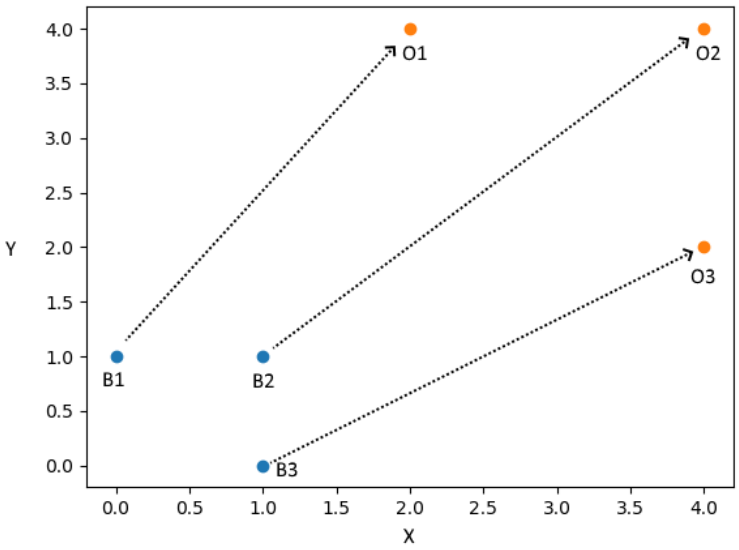


Figure 3.6: Example of a scaling of 2 and translation of 2 for both X and Y to go from the blue coordinate system given by B1, B2, and B3 to the orange coordinate system given by O1, O2, and O3. As this example features no error in the points being measured, Procrustes would be expected to report an error of 0 and the exact scaling and translation used.

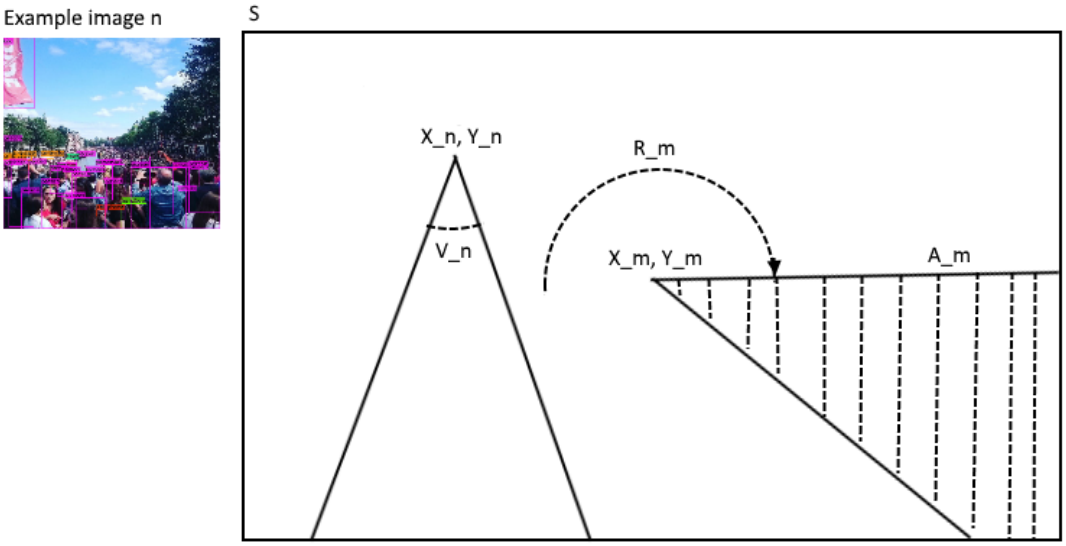


Figure 3.7: An overview of the available data. With an example of a social media image with people counted annotated by the purple boxes. The count of which gives P_n. X_n,Y_n gives the location at which image n is located. while R_m gives the direction the camera is pointed. V_n is the angle of view for image n. A_m as represented by the area given by the dotted lines is the area spanned by the projection of m into the boundary formed by the event shape S

additional point with a measured error of 10 in the x-axis. Now the transformation function will drastically change, even though the underlying affine transformations have not. For our problem, we would prefer an affine transformation that relies on less, but likely more accurate points. To achieve this, we propose first running generalized Procrustes as introduced in subsection 3.2.2 and then using its affine transformation function to see which coordinates have the smallest distance from their actual true location. We then take the first n points with the smallest distance and run Procrustes on these points. We will use the affine transformation produced by this second Procrustes as the affine transformation function for Outlier Corrected Generalized Procrustes. The n introduced should be selected in such a way that the law of large numbers still holds. A maximum of a percentage of the overall points and a constant cutoff amount of points would guarantee this in all cases where sufficient points are available to meet the constant cutoff amount of points.

3.2.4. Estimate location from images using *Structure from Motion*

Initially, the combined streetview and social media image, whose location is to be determined, is used by the *Structure from Motion* system to produce X, Y, Z coordinates for those social media images that could be reconstructed. Social media images that are not part of the reconstruction are determined not to be in the event area. The streetview images' X, Z coordinates, and longitude and latitude are used by *Generalized Procrustes Analysis* to produce a translation, scaling, and rotation operations to map any X, Z coordinate to a corresponding longitude, latitude coordinate. These translation, scaling, and rotation operations are then applied to the X, Z coordinates of social media images, produced by the *Structure from Motion* system, to produce the longitude, latitude location estimation for the social media images.

3.3. Estimate density of crowds at city events using social media images

We propose a methodology for estimating the density of crowds at city events using social media images. For our initial set of social media images I, for every social media image $i \in I$ taken during the events, we assume to have the number of people in the image P_i. We have the longitude X_i, latitude Y_i, and rotation projection R_i of the image. We also have the timestamp T_i. We know the shape, or geometry, S of the event, which forms an area within which all images must be located, also referred to as the event area.

The variable A_i is the area covered by the image projection, or A for the entire area covered by S. See also Figure 3.7 for a visual representation of the introduced variables. If we were to assume that X_i, Y_i, and R_i give the exact location and rotation and that the angle of view is known and assume that P_i gives the exact accurate amount of people in the image, even including (fully) obfuscated people. Then the density of part of the terrain can be determined perfectly through Equation 3.1, which corresponds to the density of everything

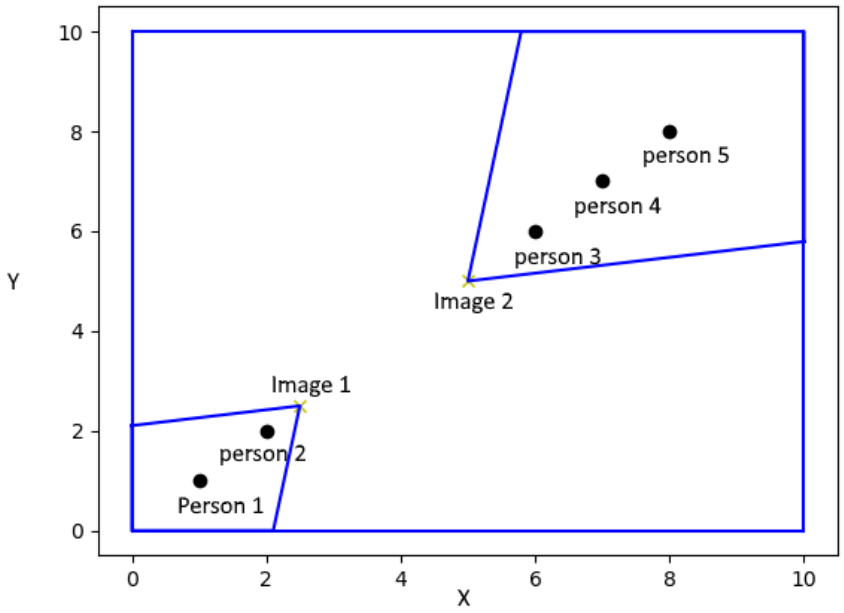


Figure 3.8: Example of the situation where all people are present in exactly one image. There are 2 camera points given by image 1 at [2.5,2.5] and image 2 at [5,5]. the angle of view is 72 degrees and there are 5 people identified by the image by the dots named person 1 through 5 specifically at [1,1], [2,2], [6,6], [7,7], [8,8]. The event area is from 0-10 for both x and y and recorded in meters giving an area of $100m^2$. The true density in this example is 5/100=0.05. the density method proposed in Equation 3.2 is also 0.05 for this example.

seen in the image.

$$D_i = \frac{P_i}{A_i} \tag{3.1}$$

If no image projections were to overlap, and each person is present in an image, then every person is counted exactly once. This ensures $\sum_{i \in I} P_i$ equals the number of people present and the density can be calculated through Equation 3.2 an example of this situation can be found in Figure 3.8. A real-world example of overlap can be seen in Figure 3.9

$$D = \frac{\sum_{i \in I} P_i}{A} \tag{3.2}$$

This method would, provided all assumptions hold, provide a perfect density estimation. However, some of these assumptions are unlikely to hold for any real-world implementation. We propose several modifications to these core assumptions to allow for a density implementation. By loosening these core assumptions, we introduce specific sources of error to the estimate. However, it allows the model to be used in real-life settings.

To address the assumption that no image projections can overlap, we first must look at what loosening this restriction entails. Set theory tells us that allowing overlap between pairs of image projections produces an overcount of $P1 \cap 2$ as per Equation 3.3

$$P_1 + P_2 = P_{1-2} + P_{2-1} + 2 * P_{1 \cap 2} \tag{3.3}$$

If we can determine $P_{1 \cap 2}$, we can correct this and allow for overlaps of pairs of image projections. However, it is unknown which part of the set people belong to as their exact location will be unknown. To make getting a result still possible, we propose assuming that people are uniformly distributed over the image projection area. This assumption means that where overlap exists, people present are the average of the overlapping image projection areas. This results in Equation 3.4

$$P_{1 \cap 2} = \frac{D_1 A_{1 \cap 2} + D_2 A_{1 \cap 2}}{2} \tag{3.4}$$

(a) Example of an hypothetical overlap given by the red bounding box. One person is present in the image overlap.

(b) Example of an hypothetical overlap given by the red bounding box. One person is present in the image overlap.

(c) Example of an hypothetical overlap given by the red bounding box. One person is present outside the image overlap.

(d) Example of an hypothetical overlap given by the red bounding box. One person is present outside the image overlap.

Figure 3.9: 4 images with illustrative image projection overlaps given by the red bounding box. Given these 4 images we would count 4 people if only the aggregate of people counted in images is used. However for Figure 3.9a and Figure 3.9b there is a person present within the overlap. This means that the person in the overlap gets counted twice and should only be counted once. therefor an overcount of 1 person has occured by naively counting people in the images. The persons in Figure 3.9c and Figure 3.9d do not occur in any overlaps and therefor should be counted normally as is.

We can then update our densities by distributing the overcount of people in $P_{1\cap 2}$ between the people counted in image 1 and 2 in proportion to the amount of people seen in the images as shown in Equation 3.5 and Equation 3.6 and an example can be found in Figure 3.10. The updated density can be calculated by dividing the updated amount of people by the area as shown in Equation 3.1 a visual example of this overcount correction is available in Figure 3.10. This process is then repeated for all overlaps created by the overlapping areas between pairs of image projections.

$$P_1 = P_1 - P_{1\cap 2} * \frac{P_1}{P_1 + P_2} \tag{3.5}$$

$$P_2 = P_2 - P_{1\cap 2} * \frac{P_2}{P_1 + P_2} \tag{3.6}$$

To address our assumption that each person is present in an image, we assume that the density of the area covered by images is equal to the density of the area not covered by images. This assumption is, in fact, the assumption that sampling a true distribution can result in measuring the true distribution. It is a general assumption made for all measurements and should hold as long as the error measured is randomly distributed and has no bias. Social media data, as proposed for our usage, is generally considered a biased data source[6, 17, 20]. While it is certainly the case that certain high-interest areas may be over-represented by social media images, we already correct for this with the overcount correction. So unless social media users go out of their way to exclusively make social media images of high or low-density areas, this bias should be limited. We are also currently assuming that we count the people in an image perfectly. This assumption

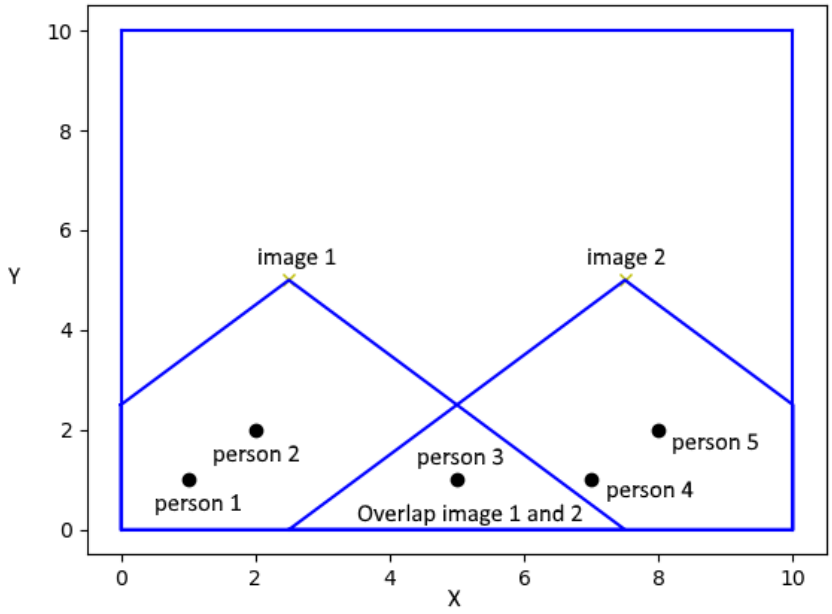


Figure 3.10: Example of overcounting correction. The projection of image 1 and image 2 are double-counting person 3 at [5,1]. Therefore the true overcount is 1. Resulting in an estimated crowd size of 6 purely based on people counted in images compared to a true crowd size of 5. The areas of both image projections have a density of 0.137, and the overlapping area is 6.25. Our proposed overcount method results in an $P_{1\cap 2}$ of 1.7125 as given by Equation 3.4. This variable, in turn, leads to updated P_1 and P_2 of 2.14375 for both. The total amount of people estimated to be present after overcounting correction is 4.2875, whereas it was six without overcount correction. The overcount is not perfectly addressed because, in this example, the persons are not uniformly distributed over the image projection.

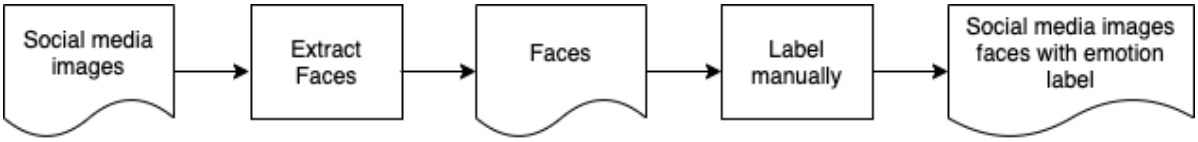


Figure 3.11: Methodology for emotion analysis for crowds at city events using social media images.

is kept as it isolates our proposed method's error from the error of any one specific "counting people in an image" method. As there are multiple methods[24, 28, 51] that focus on this specific sub-problem, it seems reasonable to isolate the problem of counting people in images from an overall density estimate method. Currently, there is also the assumption that multiple social media images are taken at the same time. We instead propose to assume that density and the distribution of people are consistent over smaller time-frames. This assumption allows social media images in the same time window instead of the same timestamp to contribute to the density estimation. This approach of using time windows to bin social media posts together for a single density estimate has been shown to work in real-world settings in previous research by Gong et al.[15]

This addressing of assumptions leaves our assumption that the overcount correction adequately addresses the existing overcount. For this, we propose an experiment in subsection 4.2.2

3.4. Analyse emotions of crowds using social media images

Our research introduces a method to analyze the emotions of crowds at city events using social media images. It will focus on the following facial expressions and their related emotions: neutral, angry, anxiety, fear, sadness, happiness. To describe the method, we will first introduce how to extract faces from images and then explain how we will detect emotions in faces. A visual overview of the steps involved can be found in Figure 3.11.

3.4.1. Extract faces from images

Emotions are usually displayed individually, although some correlation may exist between people close by each other in the same image. Therefore we start by identifying people individually. Extracting faces is a logical choice, as humans show many emotions with their faces. This face extraction can be achieved using a framework such as YOLO[45], which takes an image as input and outputs a bounding box for detected objects. One such object is the face. Afterward, this bounding box can be used to construct an image of the face itself.

3.4.2. Detect emotions in facial expressions

Detecting emotions in images is a field where research is already done[4, 13, 25]. However, no information seems known about the distribution of these emotions in social media images at city events. We propose manually labeling faces to produce an emotion from Neutral, Happy, Sad, Angry, Surprised, or Fear label for that face. We opted to manually label these images as while automated systems exist to classify emotions in faces. They tend to be quite limited in their accuracy compared to their ground truth data, with an accuracy of 0.71161 achieved in a public contest for achieving high accuracy in facial emotion classification[25]. The process by which this is done specifically is given in subsection 4.2.3

4

Experimental Setup

In this section, we introduce the experiment setting for answering the research question. First, we present the four datasets used for the experiments. As the main research question consists of 3 research sub-questions, we conduct three experiments, one for each research sub-question. We introduce each experiment's setting in terms of variables, process, dataset, and evaluation metrics.

4.1. Datasets

In our research, four datasets were used. We will give an overview of the data involved and justify its inclusion in this research.

1. Social media dataset collected by Gong et al.[15]
2. Amsterdam open dataset[36]: panorama images
3. Manually produced social media image data
4. Crowd simulation data: social media activity and density

4.1.1. Social media dataset collected by Gong et al.[15]

This dataset contains Instagram social media posts taken at six city events located in Amsterdam and Rotterdam. Each data row represents a social media post. Each data row includes the post URL, which references the original web page of the post and allows retrieval of the original post image and the corresponding data gathered alongside it. It was originally used in their works [15, 16, 51]. Each data item consists of the following fields.

1. Post link: This is the URL to the original social media post's web page, which also contains the social media image. 374 posts out of 2027 were no longer available and thus excluded from this research.
2. Timestamp: the timestamp in The Netherlands for the moment the post was created.
3. Event: the event during which the post was sent.
4. Longitude, latitude: used for locating where the image was supposedly taken. However, out of 512 images for all Kingsday and Sail events, only 88 locations are distinct. Due to the unlikelihood of recording multiple images at the exact longitude and latitude due to noise in the measurement. We had to exclude this parameter from our research due to unreliability in defining the location an image was originally taken.

It was included in this research as a source of social media images taken at city events. This dataset was used for the experiment proposed in subsection 4.2.3.

Figure 4.1: Example of an equirectangular_full panorama image taken at "De Dam" in Amsterdam taken from the Amsterdam Open Dataset.

4.1.2. Amsterdam Open Dataset

The Amsterdam Open Dataset[36] contains a set of data related to the city of Amsterdam. For this research, we only used their panorama data[37] and the corresponding longitude and latitude. More specifically, we used the image provided by the URL located at "equirectangular_full" in the Amsterdam Open Dataset panorama data structure. This image is the full-size panorama image. An example panorama image is given by Figure 4.1. Furthermore, the Amsterdam Open Dataset was used for the coordinates in the form of longitude and latitude values corresponding to where the panorama image was taken. It was included in this research as a source of images with location data for seed images in being able to determine how an area looks using *Structure from Motion* as described in subsection 4.2.2

4.1.3. Image Data collected manually in this research

We decided to take images using a smartphone for images with corresponding longitude and latitude, gathered similarly to how social media images would be made. Primarily because of the integrated GPS allowing simultaneous capture of image and location. This gathering of images was done in several areas over Amsterdam, for which the Amsterdam open dataset has panorama images. In addition, all event areas correspond to previous work done by Gong et al. allowing better comparability. Except for the De Dam event area, which was included as it is frequently used for events and one of the few areas in the Netherlands that featured comparatively large quantities of people during the covid-19 pandemic. The images were gathered using a Nokia 8 Sirocco, and GPS was turned on. The Google camera app automatically saves the location into the images' Exif data. Images were recorded using the main 12 Megapixel camera, stored in .jpg, and no additional compression was applied. The images were gathered randomly by walking a few meters and then taking one or more images in random directions. All event areas are in an urban environment in Amsterdam. An example image of the dataset can be found in Figure 4.2.

It was included in this work as a source of stand-in images for social media images at city events with known longitude and latitude, with an accuracy of about 5 meters as per the U.S. Air Force on the accuracy of smartphone GPS signal[11]. It was selected over social media images gathered at city events as due to the Covid-19 pandemic; no actual events took place. Moreover, no dataset could be identified of social media images taken at city events with reliable longitude and latitude for where the image itself was taken. One such dataset examined was provided by Gong et al.. However, it was unsuited for our needs due to issues with the location data as discussed in subsection 4.1.1. It was used for the experiment introduced in subsection 4.2.1

They are considered equivalent to social media images taken at city events in the following respects:

- They are taken through a smartphone, resulting in more ”noisy” images compared to images taken with professional camera equipment.
- They are taken at areas used for events, which means that the efficacy of the provided method can be related to event areas.

They may deviate from social media images taken at events in the following ways:

Figure 4.2: Image recorded at latitude and longitude of 52.373159 and 4.892017 correspondingly at De Dam in Amsterdam.

- No selfies or group photos, which while a common occurrence in social media images taken at city events. Based on the dataset discussed in subsection 4.1.1, these images still leave large parts of the background architecture visible in the background in most images. So this should have minimal impact on the overall accuracy for the purpose of location estimation.

The selected event areas along with the number of gathered images for that area can be found in Table 4.1.

Table 4.1: Event areas and their corresponding amount of manually gathered images.

Event Area	Gathered Images
Javakade	86
Ruijterskade	62
Sumatrakade	57
Veemkade	69
De Dam	53
Zuidplein	33

Event area: Event area images where gathered.
Gathered Images: The number of images gathered for the corresponding event area.

4.1.4. Crowd simulation data: social media activity and density

For our crowd simulation of city events, we are grouping together multiple timestamps of social media posts into a single timeframe as used successfully for the purpose of estimating density by Gong et al.[15]. This grouping implies that the amount of people that make a social media image in that timeframe is controlled

Figure 4.3: example of 2 simulated groups moving through an urban map, using Cromosim[10] for crowd simulation. The image was taken after 6.90s of simulation. The population starts randomly distributed at the bottom and heads for the exit at the top, given in red. People disappear once they hit the exit, causing a rapid reduction of the population after several seconds.

by the length of the timeframe, with a longer timeframe likely to capture more social media posts. Therefore, it makes sense to focus on the percentage of people who make social media posts, as, for a real-world application, the timeframe can be matched to fit that percentage.

The other important factor for our crowd simulation is the true density. This factor allows comparing with the prediction given by the method determining the accuracy of the method.

There are many different approaches to doing crowd simulation. Twenty-five of which are examined by Richards[46]. However, many of these approaches are either not freely available such as uCrowds [49]. Do not allow for successful installation due to reasons such as deprecation, not providing required software (or the specific required versions of software), or do not provide installation instructions prohibiting installation. Finally, of those crowd simulations that do work and are freely available, simulations are not suited for crowds at city events because they rely on static populations as shown in Figure 4.3. In contrast, city events feature people entering and leaving the event terrain continuously. The freely available alternatives focus on timescales of seconds and amounts of people up to roughly 100. While the density of crowds at city events is measured in minutes or hours, as exemplified by Gong et al.[15] and can easily feature tens of thousands of people. These crowd sizes make computation of city event scale crowd simulations implausible using these methods.

Formal crowd simulation techniques focus on the movements and interactions between people. While it could be of interest to see how our method functions at different times. It is not strictly necessary for our needs. These crowd simulation methods usually focus on different interactions between people, such as keeping a minimum distance from others or staying closer to the group they move in. However, taken at a large scale, these still result in a roughly uniform distribution in the areas where people are located in the simulation. In addition, their initial location is usually given by a uniform distribution. Therefore, taking a uniform distribution of people's locations should be somewhat realistic for crowd simulation purposes. This approach is, for example, taken by [10] which is based on the works by Maury and Faure[31].

This approach is not to say that no uniform distributions can exist. For events such as Kingsday, or a protest, the more-or-less uniform distribution of crowds makes sense as people do not have anyone particular focus at the event. For events like Sail this assumption does not necessarily make sense as on the waterside of Sumatrakade, boats will be sailing, forming the points of interest. Therefore, it can be expected that people bunch together closer to the waterside and thin out towards the back to see more of the point of interest.

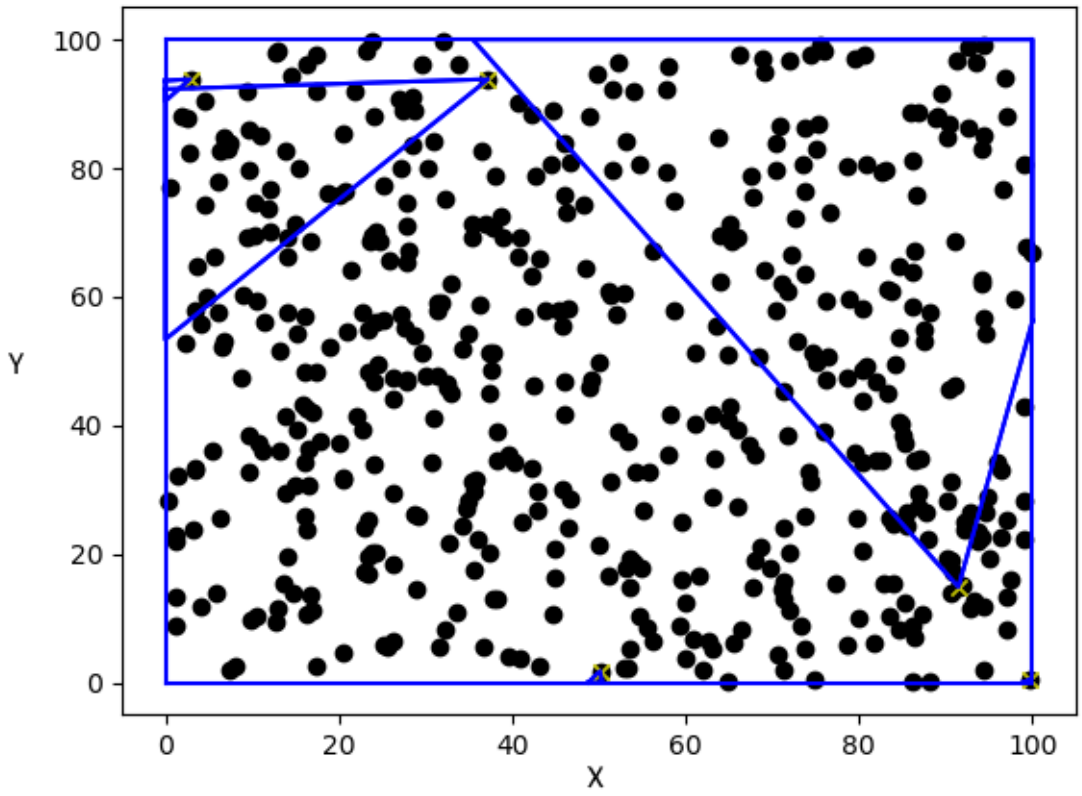


Figure 4.4: Crowd simulation of a 100 by 100 event area with 500 people (i.e., a *crowd density* of 0.05) and five people are producing social media images (i.e., a social media activity of 0.01). Two of these are easily identified by the blue projection lines. The other three are located in the top left, center bottom, and bottom right. The top left, and the top center viewpoints are the only ones that feature an overlap. In the diagram, the X and Y-axis denote meters and are intended to demonstrate relative positions between people who are given by the black dots.

Based on this, we will propose our crowd simulation method. Our crowd simulation has three variables. The control variable event area A. The free variable true *crowd density* at the event: D. The free variable percentage of people who produce social media images is also referred to as the social media activity: P and the control variable angle of view for each image projection: AoV. The simulation is then constructed as follows:

1. Calculate the number of people N at the event by multiplying the event area size by the true *crowd density*.
2. Create N uniformly random coordinates within the event area for each of the people at the event.
3. Randomly select $N * P$ people. These people are designated as those who make a social media image. Assume each image is taken in a random direction R_i and see everyone in a triangle starting at the image's location with an angle of view AoV.
4. For each social media image, count the number of people that are in the projection. This count gives C_i the number of people seen in image i.
5. Record the coordinates X_i, Y_i, direction R_i, angle of view AoV and people counts C_i of each image i. These are the features available for density estimation. For comparison purposes, the true density and percentage of people who produce social media images are recorded.

A visual representation of a crowd simulation can be found in Figure 4.4

Crowd simulation data was used for the density experiment proposed in subsection 4.2.2

4.2. Experimental setting

In the following section, we introduce three experiments. Each experiment is introduced for investigating each research sub-question.

4.2.1. Experiment 1: Estimate location from images using *Structure from Motion (SfM)*

To answer the first research question, "How can the geographic location of a social media picture taken during a city-scale event be estimated?" we propose an experiment to validate the location estimation method proposed in section 3.2. The experiment should validate the accuracy of predicting whether an image is taken at an event area and determine the accuracy of the predicted location if it is in the event area.

The independent variables of this experiment are the selected event areas. These influence the dependent variables of the location estimate.

For predicting whether an image is in the event area or not, we will report the confusion matrix for images taken in the event area, referred to also as in event area, or not in event area if an image was not originally taken in the event area involved, as well as images not taken in the event area. The confusion matrix is a metric used for classification problems[48] and provides insight into how well a classification system performs.

For predicting where exactly an image was taken at the event, we use the distance between the true location and the predicted location as the error in location estimation. We will be reporting these errors in the form of radii of confidence, which is commonly used in location-based metrics[18] and provides insight into how likely a particular deviation is. The radius of confidence is the likelihood that the true location and estimated location deviate less than n. For our experiment, n is chosen to be set at [5, 10, 20, 50, 100, 100+] and is measured in meters. These were chosen because the accuracy of smartphones is about five meters[11], giving us a significant lower bound. Furthermore, the upper bound error of 100+ meters corresponds to being outside of the event area in most cases.

To perform this experiment, we need to select event areas. We selected De Dam, Sumatrakade and Zuidplein in Amsterdam shown in Figure 4.9, Figure 4.8 and Figure 4.7 respectively. Sumatrakade and Zuidplein have been previously used by Gong et al.[15]. These event areas were chosen because they are locations used for city events in the past, such as Kingsday, Sail, and protests. Sumatrakade and Zuidplein were also chosen because they would allow comparing results with previous research done by Gong et al.[15].

For these event areas, we need social media images with reliable locations and seed images of the event terrain with their location as described in subsection 3.2.1. The amount of these images should be large enough that the entire area is covered in a reconstruction. While being small enough to still be able to compute the necessary amount of reconstructions within a reasonable timeframe as described in subsection 3.2.1. If the improvements proposed in subsection 7.2.1 are applied, this requirement would be lessened.

For the necessary social media images, there is a dataset by Gong et al.[15]. However, due to the problems described in subsection 4.1.1, i.e., the data likely not representing the actual location the image was taken. This dataset was not used, and instead, data was manually gathered as described in subsection 4.1.3. Specifically, the data items gathered consist of the following: an image taken at the event area and the longitude and latitude where that image was taken. For the images of the event terrain, also referred to as seed images as discussed in subsection 3.2.1, we selected the Amsterdam open dataset[36], which contains panorama images and their location for most of Amsterdam. This dataset was chosen because of the high density of image locations and accurate measurements (errors of up to 0.8m Root Mean Square[35])

For our *Structure from Motion (SfM)* method, we used *OpenSfM*[30] this implementation was selected because it features a more-or-less default implementation of the *SfM* approach as introduced in subsection 3.2.1, its primary deviation is that it uses HAHOG, a combination of Hessian Affine Region Detector and Histogram of oriented gradients as its default feature extractor. It was constructed to perform similarly to SIFT, both being rotation and scaling invariant. However, unlike the more commonly used SIFT technique, which is patented and therefore not necessarily publicly available, it is open source. It was originally developed by Mapillary to reconstruct urban streetviews. Therefore it should provide a decent baseline of the performance of our method because it is a more or less default implementation of a Structure from Motion system and has been used in urban settings with success before.

Our experiment is conducted through the following steps:

1. For each of the selected event areas, retrieve the seed images and their longitude and latitude from Amsterdam open data.

2. For each of the seed images convert it from panorama view to cubic images in the top, back, front, left, right, front-left, front-right, back-left, and back-right view using the equirectangular toolbox[33].

Figure 4.5: Example of an left cubic projection, extracted using the equirectangular toolbox. This cubic left view was generated from the panorama in Figure 4.6

The equirectangular toolbox allows taking an equirectangular projection and creating cubic projections from it. The bottom view is not created as it only included the vehicle taking the images, not the actual event area. An example of a panorama image can be found in Figure 4.6, an example of an extracted view from that panorama image can be found in Figure 4.5

3. To limit computational complexity as discussed in subsection 3.2.1, if there are more than 1000 seeds images, select a random sample of 1000 seed images. This sampling was used for De Dam and Sumatrakade, which had 3323 and 6246 seed images, respectively, for their area size.

4. For the selected event terrain, use the images of that event area in the manual dataset introduced in subsection 4.1.3 as the images that were taken at that event area. For the images not taken at the event area, we will select a random sample of 100 taken from all event areas that were not selected also introduced in subsection 4.1.3

5. Construct a *Structure from Motion (SfM)* reconstruction for the selected event area for each of the selected social media images. The *SfM* uses all seed images of that terrain and the selected social media image, as proposed in subsection 3.2.1. These combinations of data inputted into the *SfM* produce the reconstructions. For the SfM, we used OpenSfM[30] as discussed earlier in this section.

6. For each social media image. Construct a reconstruction as proposed in subsection 3.2.4. Classify based on this reconstruction whether the social media image is at the event terrain or not as per the method proposed in subsection 3.2.4.

7. For each social media image classified as at the event terrain. Compute the longitude and latitude using the Procrustes method proposed in subsection 3.2.2 for the location estimation method proposed in subsection 3.2.4. This gives the predicted location of the social media image, using Procrustes.

8. For each social media image classified as at the event terrain. Compute the longitude and latitude using the *Outlier Corrected Procrustes* method proposed in subsection 3.2.3 for the location estimation method proposed in subsection 3.2.4. This gives the predicted location of the social media image, using outlier corrected Procrustes.

9. Calculate the distance between the true and predicted location for both Procrustes and Outlier Corrected Procrustes using the distance function given by Geopy[14]

10. Use the distance between the true and predicted location as the error for the radii of confidence introduced earlier in this section.

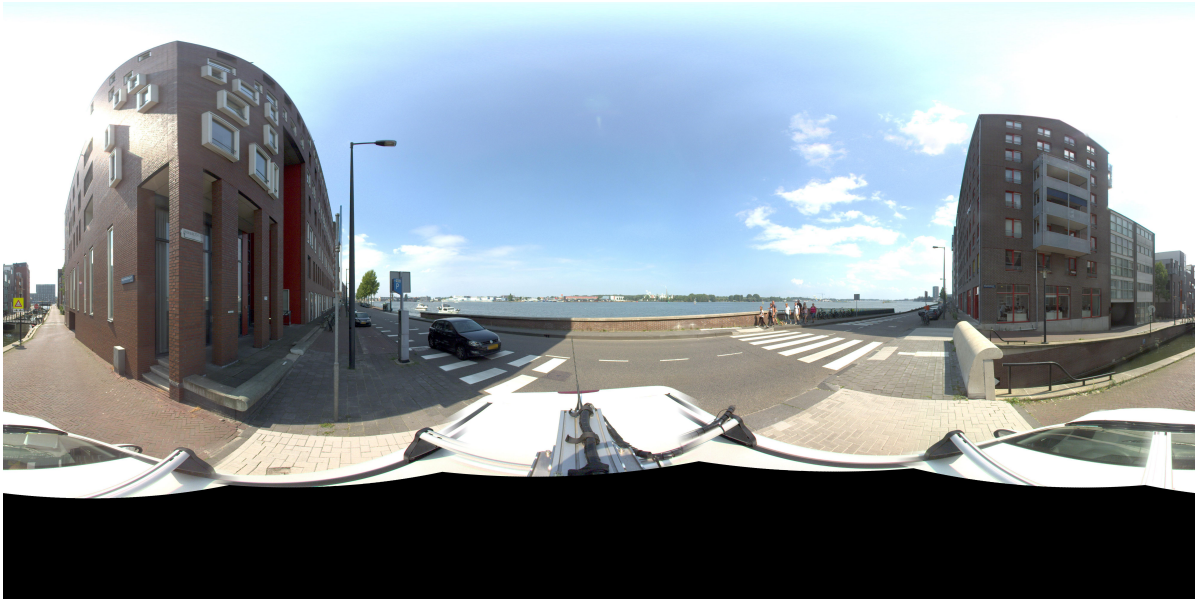


Figure 4.6: Panorama view example for the Sumatrakade, taken from the Amsterdam Open Dataset[36]

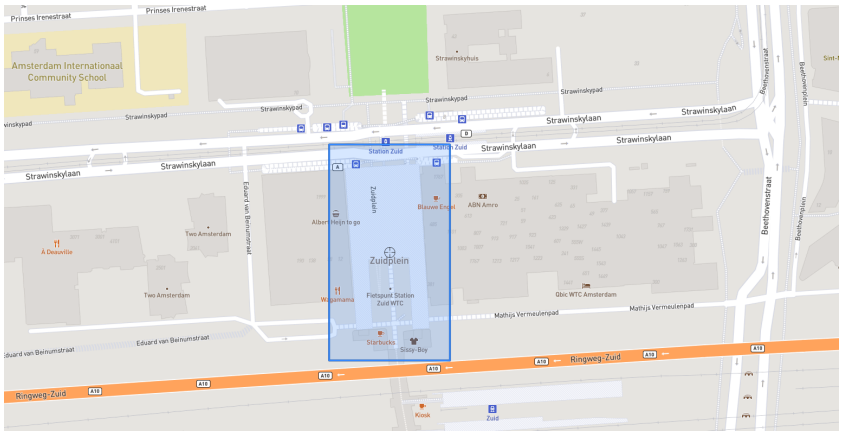


Figure 4.7: The blue area denotes the area used for the Zuidplein event area.

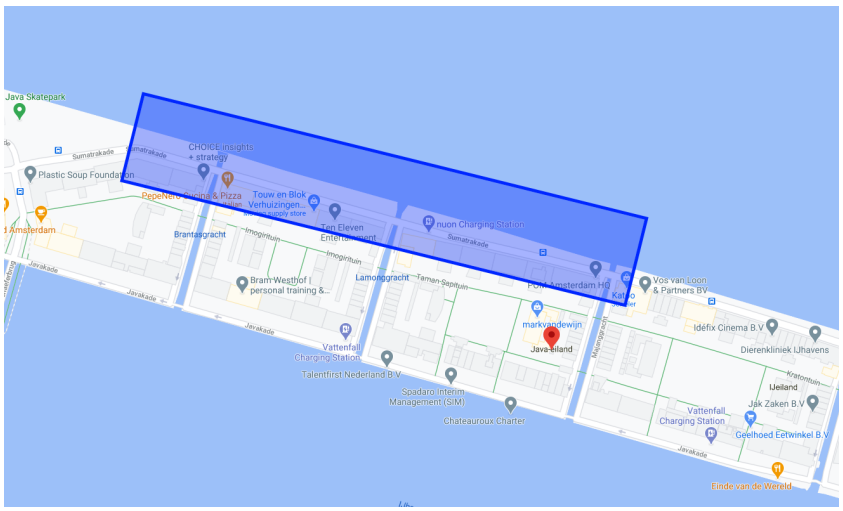


Figure 4.8: The blue area denotes the area used for the Sumatrakade event area.

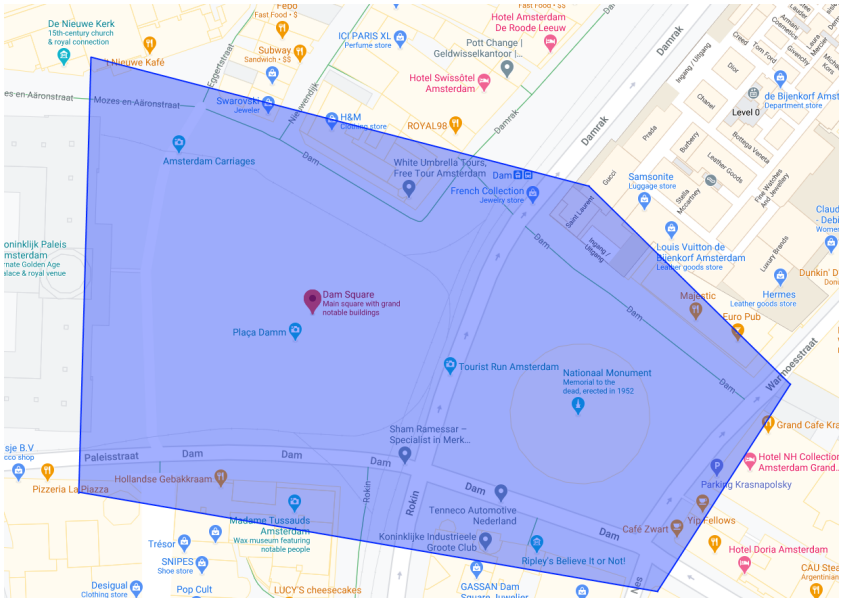


Figure 4.9: The blue area denotes the area used for De Dam event area.

4.2.2. Experiment 2: Estimate density of crowds using social media images

To answer our second research question, "How can images be used to estimate *crowd density* from social media without crowd measurement infrastructure?" we propose an experiment to validate the methodology proposed in section 3.3 the experiment should validate the accuracy of the estimated density.

In this experiment, we will be relying on a crowd simulation for producing location, direction, and amount of people in social media images. This was done because of our need for a ground truth of density for an event as well as ground truth data of the direction and location of where the social media images were taken. The proposed method for extracting these features in a real-world scenario can be found in section 3.1.

Initially, a dataset providing ground truth density data matching the event areas in the social media dataset introduced in subsection 4.1.1 was identified. However, as the locations in the social media dataset were found to not match the image location as discussed in subsection 4.1.1 and manually labeling the locations of these images was deemed to be too unreliable, this option was not chosen.

Gathering a new dataset was also infeasible as the research occurred during the Covid-19 pandemic. Therefore all large-scale city events were canceled and are expected to remain canceled for the foreseeable future. These reasons lead us to select crowd simulation as a fallback option to still be able to analyze our proposed method. The specifics of the used crowd simulation are discussed in subsection 4.1.4

For this experiment, we will be focusing on the independent variables of *social media activity* and true *crowd density*. Both influence the density estimate, the dependent variable. For the true density value, we propose a range of 0.1 to 1 in steps of 0.1. These are based on the crowd densities found by counting infrastructure in the work of Gong et al.[15]. For the social media activity, we propose a range from 0.01 to 0.1 in steps of 0.01 We will be repeating the experiment 10 times for each combination of independent variables.

For this experiment, we will be reporting the following comparing metrics. The Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) to allow comparing results with those in Gong et al.[15]. The Mean will also be recorded to provide insight into the actual values found. Furthermore, the Mean Square Error (MSE) as this metric punishes larger errors quadratically as opposed to linearly, which fits better from a crowd management perspective where a uniform error of 10% in the density estimate may still be useable. However, a single 50% or 100% difference may not.

To demonstrate the problems caused by overcounting, and thus the necessity of overcount correction. We will be recording these metrics for both the overcount corrected and non overcount corrected density estimates.

Our experiment will be conducted through the following steps:

1. For each combination of true density value and social media activity, construct ten crowd simulations using an angle of view of 72 degrees.

2. For each crowd simulation record the density estimates given by the method proposed in section 3.3 without the overcount correction applied.

3. For each crowd simulation run the density estimation methodology described in section 3.3 resulting in a density estimate

4. Record each density estimate and the variables under which it was produced.

5. From the recorded values, calculate the comparing metrics for both the overcount corrected and non overcount corrected method.

4.2.3. Experiment 3: Estimate emotions of crowds using social media images

To answer our third research question, "How can social media images be used to study the emotions expressed by participants of city-scale events?". We propose an experiment to validate the methodology proposed in section 3.4. The experiment should show the distribution of emotions in social media images at city events.

To perform this experiment, we need data about emotions shown in social media images. For this, we selected the dataset gathered by Gong et al.[15] because it allows us to gain additional insight for the same event areas used by the other experiments Kingsday and Sail. The social media dataset by Gong et al. contains 2027 images about six events in the Netherlands. It contains the following data items: the URL referring to social media posts with images and the event during which the social media post was made. More details about the dataset are found in subsection 4.1.1

We will be using the counts of emotions detected in social media images at city events for the comparing metrics.

Our experiment will be conducted through the following steps:

- download the images for connected to the social media posts in the dataset gathered by Gong et al.[15] and discussed in subsection 4.1.1

- extract faces from the selected social media images using YOLO[45]

- label each of the extracted faces with their true label chosen from neutral, angry, anxiety, fear, sadness, and happiness. This labeling gives the true label.

- record the distribution of emotions shown at social media events.

5

Findings

In this section we will present the findings for each of the executed experiments proposed in chapter 4.

5.1. location estimation

In the experiment for location estimation proposed in subsection 4.2.1, we proposed using the confusion matrix as the metric to be recorded. This metric gives for each class, each combination of the predicted class and the actual class. For our experiment, these classes are the class that an image was in the event area, and the class that an image was not in the event area.

These classes can be seen in the confusion matrix for the De Dam event area given in Table 5.1. Were of the 53 images from the event area, four were correctly predicted to be from there. Simultaneously, 49 images from the event area were classified, in error, as not being from the event area. For images not from the event area 0 were classified, in error, as taken at the event area. With 100 images not from the event area being, correctly, classified as not being from the event area. For the dam, based on our measurements, this means we achieved a 7.5% accuracy at correctly classifying images from that area. At the same time, it achieved a 100% accuracy at correctly classifying images, not from the event area.

Table 5.1: De Dam results for predicting whether an image is in an event area or not.

Actual	**Predicted**	
	In event area	Not in event area
In event area	4	49
Not in event area	0	100

Predicted: Prediction given by our system
Actual: Ground truth from data gathered in subsection 4.1.3
(Not) In event area: Whether the image up for consideration is predicted or actually at the event area or not. These are the classes considered.
Result cell value: The amount of images for which the given actual class matches the given predicted class.

Similarly to the De Dam event area, for the Sumatrakade we find in Table 5.2 3 out of 57 images taken at the event area are correctly classified, with 54 incorrectly classified. Furthermore, all 100 images not from the event area are correctly classified as not being from the event area. These results mean that, based on our measurements. For the Sumatrakade, our proposed method achieves a 5% accuracy in classifying whether an image is from the event area and a 100% accuracy for images not taken at the event area.

Table 5.2: Sumatrakade results for predicting whether an image is in an event area or not.

Actual	**Predicted**	
	In event area	Not in event area
In event area	3	54
Not in event area	0	100

Predicted: Prediction given by our system
Actual: Ground truth from data gathered in subsection 4.1.3
(Not) In event area: Whether the image up for consideration is predicted or actually at the event area or not. These are the classes considered.
Result cell value: The number of images for which the given actual class matches the given predicted class.

For Zuidplein, as shown in Table 5.3, we find no successful classifications for images taken at the event area. This absence of results is possibly due to problems with the seed images for that area. However, it may also be attributable to the low probability of an image being correctly classified, as shown by De Dam and Sumatrakade classification probability. This problem is discussed in detail in section 6.1

Table 5.3: Zuidplein results for predicting whether an image is in an event area or not.

True Class	**Predicted Class**	
	In event area	Not in event area
In event area	0	33
Not in event area	0	100

Predicted: Prediction given by our system
Actual: Ground truth from data gathered in subsection 4.1.3
(Not) In event area: Whether the image up for consideration is predicted or actually at the event area or not. These are the classes considered.
Result cell value: The number of images for which the given actual class matches the given predicted class.

Overall as shown in Table 5.4, these results lead to 7 out of 143 images from the event area being correctly classified as being from the event area. These results mean a 5% classification accuracy for images from the event area. All 300 tested images, not from the event area are classified as not being from the event area resulting in a 100% classification accuracy for images not from the event area.

Table 5.4: Overall results for predicting whether an image is in an event area or not.

Actual	**Predicted**	
	In event area	Not in event area
In event area	7	140
Not in event area	0	300

Predicted: Prediction given by our system
Actual: Ground truth from data gathered in subsection 4.1.3
(Not) In event area: Whether the image up for consideration is predicted or actually at the event area or not. These are the classes considered.
Result cell value: The amount of images for which the given actual class matches the given predicted class.

The seven images correctly classified as being part of the event area earlier in this section. We were able to extract the radii of confidence as proposed in subsection 4.2.1.

The results of the radii of confidence found through the generalized Procrustes approach introduced in subsection 3.2.2 are found in Table 5.5. We find that all four measurements are between 50 and 100 meters accurate for the De Dam event area. While for the Sumatrakade, all three measurements deviate from their true value by more than 100 meters. As Zuidplein did not have any images correctly classified as part of the event area, no area of radii of confidence could be generated for these images.

Table 5.5: Circle of accuracy achieved by comparing location estimation based on Procrustes method introduced in subsection 3.2.2 method with ground truth. with each row representing the event area compared. And each column the circle of accuracy radius.

Event	**Circle of accuracy**					
Area	5m	10m	20m	50m	100m	100+m
de Dam	0	0	0	0	4	0
Sumatrakade	0	0	0	0	0	3
Zuidplein	0	0	0	0	0	0
Overall	0	0	0	0	4	7

Event Area: the selected event areas as introduced in subsection 4.2.1.
Circle of accuracy: The circle centered on the true location with a radius of the column value within which images are accurately located.
Result cell value: The number of images for the event terrain for which the circle of accuracy, with a radius given by the column value, encompasses the estimated location.

We also proposed *Outlier Corrected Procrustes* in subsection 3.2.3 the results for applying this method are found in Table 5.6. Here we find that the De Dam event area has one measurement with an accuracy between 20 and 50 meters and three measurements with an accuracy between 50 and 100 meters. For Sumatrakade, we have one measurement with an accuracy between 5 and 10 meters and two measurements with an accuracy of fewer than 100 meters. As Zuidplein did not have any images correctly classified as part of the event area, no area of confidence could be generated for these images.

As can be seen from the measurements in Table 5.5 and Table 5.6. Accuracies are given by *Outlier Corrected Procrustes* improve on the accuracies given by *Generalized Procrustes* for all event areas.

We discuss implications of the found results in section 6.1

Table 5.6: circle of accuracy achieved by comparing location estimation method based on *Outlier Corrected Procrustes* introduced in subsection 3.2.3 with ground truth. with each row representing the event area compared. And each column the circle of confidence's radius.

Event	**Circle of accuracy**					
Area	5m	10m	20m	50m	100m	100+m
de Dam	0	0	0	1	4	4
Sumatrakade	0	1	1	1	1	3
Zuidplein	0	0	0	0	0	0
Overall	0	1	1	2	5	7

Event Area: the selected event areas as introduced in subsection 4.2.1.
Circle of accuracy: The circle centered on the true location with a radius of the column value within which images are accurately located.
cell value: The number of images for the event terrain for which the circle of accuracy, with a radius given by the column value, encompasses the estimated location.

5.2. Density estimation of crowds using social media images

The Mean results found for experiment 2 can be found in Table 5.11 for the overcount corrected results. In Table 5.7 the Mean results without overcount correction can be found. The Mean Absolute Errors (MAE) found for experiment 2 can be found in Table 5.12 for the overcount corrected results. In Table 5.8 the MAE results without overcount correction can be found. The Mean Absolute Percentage Errors (MAPE) found for experiment 2 can be found in Table 5.13 for the overcount corrected results. In Table 5.9 the MAPE results without overcount correction can be found.

Finally the Mean Squared Errors (MSE) for experiment 2 can be found in Table 5.14 for the overcount corrected results. In Table 5.10 the MSE results without overcount correction are found.

For our overcount corrected result, we find that the top-left cell highlighted in grey for a true density of 0.1 and social media activity of 0.01 has a Mean Absolute Error of 0.00423 in Table 5.12. This error results in a 4.23% deviation from the true density of 0.1 as given by the Mean Absolute percentage Error in the top left highlighted cell, under the same true density and social media activity, in Table 5.13. We also have a Root Mean Square Error of $2.25 * 10^{-5}$ for the top-left cell highlighted in grey for the same combination of true density and social media activity in Table 5.14

For the central cell highlighted in grey for a true density of 0.5 and social media activity of 0.05, we find a Mean Absolute Error of 0.497 in Table 5.12. This error results in a 99.4% deviation from the true density of 0.5 as given by the Mean Absolute percentage Error in the central highlighted cell, under the same true density and social media activity, in Table 5.13. We also have a Root Mean Square Error of 0.247 for the central cell highlighted in grey for the same combination of true density and social media activity in Table 5.14.

The produced errors are quite uniformly distributed amongst the measurements. This is demonstrated by our MSE measurements in Table 5.14, where if a single measurement were to deviate strongly, it would show up as a much larger deviation than the MSE cells around it, while a similar effect would not necessarily be seen in our MAE measurements in Table 5.12. However, this is not the case, and the MSE errors feature a steady increase in error from scenario to scenario, more or less in tandem with the MAE errors. This demonstrates the stability of measurements produced by the proposed method.

In fact, for all Mean Absolute errors reported in Table 5.12 as either social media activity or true density or both increases, the measured error also increases. The same holds true for MAPE in Table 5.13 and for MSE in Table 5.14. If the error is less uniformly distributed among the measurements, MSE should be expected to deviate much more strongly than the MAE. This deviation does not happen. It means that the distance error is more or less evenly distributed among the measurements for a specific combination of true density and social media activity.

Beyond values for crowd density larger than 0.5 or social media activities larger than 0.05, we find that the estimated density has effectively converged to 0 as shown in Table 5.11. This demonstrates a likely problem in that our overcount correction is too aggressive. The consequences and implications of which will be discussed in section 6.2.

We also measured all proposed measurements for the method proposed without overcount correction applied. Here we find that for the mean given by the highlighted top-left cell in Table 5.7 a similar result is found as with the equivalent overcount corrected result found in Table 5.11. This is likely due to the low probability of overlaps at this amount of crowd density and social media activity. However, as the crowd density and social media activity increase, the not overcount corrected error grows much faster in accuracy than the overcount corrected error as demonstrated by the two having an error of 82% and 12% respectively for crowd densities of 0.2 and social media activity of 0.02 as shown by Table 5.9 and Table 5.13. This demonstrates the added value of the overcount correction step over the method without overcount correction.

We discuss the implications of these results in section 6.2

5.3. Sentiment estimation of crowds using social media images

In Table 5.15, the aggregate counts for all hand-labeled images can be found. It is of note that 95% percent are neutral or happy. It is outnumbering the other emotions of Sad, Anger, Surprise, and Fear. We discuss the implications of these results in section 6.3

Table 5.7: Mean of 10 density estimates without overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.109	0.267	0.387	0.467	0.611	0.7	0.908	1.01	1.11	1.21
0.2	0.492	1.02	1.46	2.01	2.41	2.89	3.55	4.04	4.4	5.09
0.3	1.12	2.31	3.42	4.46	5.68	6.6	7.91	8.79	10.2	11.3
0.4	1.99	3.76	5.8	7.97	9.81	11.3	13.7	15.7	18.5	19.7
0.5	3.17	5.93	8.96	12.3	16.1	18.7	22	25.7	28.3	31.6
0.6	4.44	8.97	13.8	18.1	22.5	27.5	31.6	36.2	39.9	45.4
0.7	6.05	12.2	18.7	23.6	30.9	36.5	41.3	49.5	56	60.9
0.8	8.01	16.3	24.3	32	40	47.2	55.6	64.6	74.2	82.7
0.9	10.2	20.2	30	40.3	51.3	62.3	69.6	81.8	90.8	101
1	13	24.7	36.7	50.1	61.8	74.6	87.5	101	111	125

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values (People/M^2): Mean of the density estimate without overcount correction.

Table 5.8: Mean Absolute Error (MAE) of 10 density estimates without overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.0243	0.167	0.287	0.367	0.511	0.6	0.808	0.915	1.01	1.11
0.2	0.292	0.821	1.26	1.81	2.21	2.69	3.35	3.84	4.2	4.89
0.3	0.824	2.01	3.12	4.16	5.38	6.3	7.61	8.49	9.85	11
0.4	1.59	3.36	5.4	7.57	9.41	10.9	13.3	15.3	18.1	19.3
0.5	2.67	5.43	8.46	11.8	15.6	18.2	21.5	25.2	27.8	31.1
0.6	3.84	8.37	13.2	17.5	21.9	26.9	31	35.6	39.3	44.8
0.7	5.35	11.5	18	22.9	30.2	35.8	40.6	48.8	55.3	60.2
0.8	7.21	15.5	23.5	31.2	39.2	46.4	54.8	63.8	73.4	81.9
0.9	9.34	19.3	29.1	39.4	50.4	61.4	68.7	80.9	89.9	100
1	12	23.7	35.7	49.1	60.8	73.6	86.5	99.9	110	124

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values: Mean Absolute Error of the density (People/M^2) estimate without overcount correction.

Table 5.9: Mean Absolute Percentage Error (MAPE) of 10 density estimates without overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.243	1.67	2.87	3.67	5.11	6	8.08	9.15	10.1	11.1
0.2	1.46	4.1	6.31	9.05	11	13.4	16.8	19.2	21	24.4
0.3	2.75	6.68	10.4	13.9	17.9	21	25.4	28.3	32.8	36.6
0.4	3.98	8.41	13.5	18.9	23.5	27.3	33.3	38.2	45.2	48.3
0.5	5.35	10.9	16.9	23.6	31.2	36.4	43.1	50.4	55.7	62.3
0.6	6.4	14	22	29.1	36.5	44.8	51.7	59.4	65.5	74.7
0.7	7.65	16.4	25.8	32.8	43.2	51.1	58	69.7	79	86
0.8	9.01	19.4	29.4	39	49	58	68.5	79.8	91.7	102
0.9	10.4	21.4	32.3	43.8	56	68.2	76.3	89.9	99.9	111
1	12	23.7	35.7	49.1	60.8	73.6	86.5	99.9	110	124

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values: Mean Absolute Percentage Error of the density (People/M^2) estimate without overcount correction.

Table 5.10: Mean Squared Error (MSE) of 10 density estimates without overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.000888	0.0317	0.0874	0.139	0.266	0.368	0.662	0.851	1.04	1.25
0.2	0.0923	0.705	1.63	3.33	4.9	7.37	11.3	14.8	17.7	24
0.3	0.698	4.1	9.91	17.6	29.1	39.7	58.1	72.4	97.4	121
0.4	2.63	11.7	29.3	57.4	88.7	120	178	235	328	374
0.5	7.32	29.5	71.8	140	244	332	465	636	776	972
0.6	15	70.4	174	305	480	723	964	1.27e+03	1.55e+03	2.01e+03
0.7	29.5	134	327	527	917	1.29e+03	1.65e+03	2.38e+03	3.06e+03	3.63e+03
0.8	52.6	243	555	975	1.54e+03	2.15e+03	3.01e+03	4.08e+03	5.39e+03	6.72e+03
0.9	88.5	373	848	1.56e+03	2.54e+03	3.77e+03	4.72e+03	6.54e+03	8.09e+03	1e+04
1	146	566	1.28e+03	2.42e+03	3.71e+03	5.42e+03	7.49e+03	9.99e+03	1.22e+04	1.54e+04

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values : Mean Squared Error of the density (People/M^2) estimate without overcount correction.

Table 5.11: Mean of 10 density estimates with overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.0976	0.0776	0.0579	0.0483	0.0307	0.0222	0.0129	0.00936	0.00832	0.006
0.2	0.163	0.0797	0.0412	0.0201	0.0121	0.00762	0.00438	0.00348	0.00326	0.00212
0.3	0.176	0.0549	0.0209	0.0108	0.00626	0.00457	0.00292	0.00221	0.00122	0.00109
0.4	0.175	0.0415	0.0142	0.00667	0.00418	0.00236	0.00173	0.00187	0.00119	0.00094
0.5	0.136	0.0282	0.0108	0.00505	0.00313	0.00235	0.00164	0.000881	0.000987	0.000593
0.6	0.119	0.0196	0.00845	0.00472	0.00284	0.00156	0.00123	0.000922	0.000767	0.000601
0.7	0.101	0.0181	0.00595	0.00402	0.00157	0.00122	0.000986	0.00098	0.00056	0.000492
0.8	0.081	0.0125	0.0061	0.00317	0.00214	0.00105	0.000794	0.000733	0.000494	0.000442
0.9	0.0736	0.0118	0.00437	0.00239	0.00154	0.000975	0.0011	0.000829	0.000551	0.000447
1	0.0451	0.00864	0.00419	0.00254	0.00142	0.00083	0.000699	0.000639	0.000438	0.000568

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values (People/M^2): Mean of the density estimate with overcount correction.

Table 5.12: Mean Absolute Error (MAE) of 10 density estimates with overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.00423	0.0224	0.0421	0.0517	0.0693	0.0778	0.0871	0.0906	0.0917	0.094
0.2	0.0372	0.12	0.159	0.18	0.188	0.192	0.196	0.197	0.197	0.198
0.3	0.124	0.245	0.279	0.289	0.294	0.295	0.297	0.298	0.299	0.299
0.4	0.225	0.358	0.386	0.393	0.396	0.398	0.398	0.398	0.399	0.399
0.5	0.364	0.472	0.489	0.495	0.497	0.498	0.498	0.499	0.499	0.499
0.6	0.481	0.58	0.592	0.595	0.597	0.598	0.599	0.599	0.599	0.599
0.7	0.599	0.682	0.694	0.696	0.698	0.699	0.699	0.699	0.699	0.7
0.8	0.719	0.787	0.794	0.797	0.798	0.799	0.799	0.799	0.8	0.8
0.9	0.826	0.888	0.896	0.898	0.898	0.899	0.899	0.899	0.899	0.9
1	0.955	0.991	0.996	0.997	0.999	0.999	0.999	0.999	1	0.999

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values: Mean Absolute Error of the density (People/M^2) estimate with overcount correction.

Table 5.13: Mean Absolute Percentage Error (MAPE) of 10 density estimates with overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	0.0423	0.224	0.421	0.517	0.693	0.778	0.871	0.906	0.917	0.94
0.2	0.186	0.601	0.794	0.9	0.939	0.962	0.978	0.983	0.984	0.989
0.3	0.414	0.817	0.93	0.964	0.979	0.985	0.99	0.993	0.996	0.996
0.4	0.563	0.896	0.964	0.983	0.99	0.994	0.996	0.995	0.997	0.998
0.5	0.728	0.944	0.978	0.99	0.994	0.995	0.997	0.998	0.998	0.999
0.6	0.801	0.967	0.986	0.992	0.995	0.997	0.998	0.998	0.999	0.999
0.7	0.855	0.974	0.991	0.994	0.998	0.998	0.999	0.999	0.999	0.999
0.8	0.899	0.984	0.992	0.996	0.997	0.999	0.999	0.999	0.999	0.999
0.9	0.918	0.987	0.995	0.997	0.998	0.999	0.999	0.999	0.999	1
1	0.955	0.991	0.996	0.997	0.999	0.999	0.999	0.999	1	0.999

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values : Mean Absolute Percentage Error of the density (People/M^2) estimate with overcount correction.

Table 5.14: Mean Squared Error (MSE) of 10 density estimates with overcount correction applied. 10 crowd simulations where generated per combination of social media activity in the column and true density in the row, one for each density estimate.

True	**Social Media Activity**									
Density	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.1	2.25e-05	0.000622	0.00187	0.00278	0.00485	0.0061	0.00761	0.00823	0.00842	0.00884
0.2	0.00159	0.0149	0.0253	0.0324	0.0353	0.037	0.0383	0.0386	0.0387	0.0392
0.3	0.0163	0.0602	0.078	0.0837	0.0863	0.0873	0.0883	0.0887	0.0893	0.0893
0.4	0.0526	0.129	0.149	0.155	0.157	0.158	0.159	0.159	0.159	0.159
0.5	0.134	0.223	0.239	0.245	0.247	0.248	0.248	0.249	0.249	0.249
0.6	0.232	0.337	0.35	0.354	0.357	0.358	0.359	0.359	0.359	0.359
0.7	0.36	0.465	0.482	0.484	0.488	0.488	0.489	0.489	0.489	0.489
0.8	0.518	0.62	0.63	0.635	0.637	0.638	0.639	0.639	0.639	0.639
0.9	0.683	0.789	0.802	0.806	0.807	0.808	0.808	0.809	0.809	0.809
1	0.912	0.983	0.992	0.995	0.997	0.998	0.999	0.999	0.999	0.999

True Density (People/M^2): True crowd density used in generating the simulation as described in subsection 4.1.4.
Social Media Activity (Images/person): Chance that a person takes an social media images as described in subsection 4.1.4.
Result Cell Values : Mean Squared Error of the density (People/M^2) estimate with overcount correction.

Table 5.15: Counts of labels given to emotions shown in faces in social media images as described in subsection 4.2.3

Facial Expression	Amount
Neutral	58
Happy	166
Sad	7
Anger	0
Surprise	4
Fear	1

Facial expression: Facial expression of faces extracted from social media images taken at city events.
Amount: Amount of times the corresponding facial expression was found in the faces extracted from the dataset.

6

Discussion

In this chapter we reflect on the results given in chapter 5 of the performed experiments proposed in chapter 4. The discussion is broken up by experiment and includes possible limitations and threats to the validity of the performed experiments.

6.1. Discussion on location estimation

For our first experiment, the intended results for Zuidplein were not found as the *Structure from Motion system (SfM)* never did reconstruction with images that should be part of the reconstruction. This absence of reconstructions may be due to the low chance of 5% an image being correctly classified as taken at the event area.

This low chance implies that for the 33 images taken at the Zuidplein event area, the expected amount of images to be classified as part of the event area is $33 * 0.05 = 1.65$. It is not entirely inconceivable that due to the low probability that an image is correctly classified as part of the event area, this ends up being 0 in our experiment. For example, it may be possible that Zuidplein is less likely to include images in the reconstruction due to a large amount of highly reflective glass and trees in the architectural choice compared to Sumatrakade and De Dam.

However, it was noticed upon close examination that the Zuidplein panoramas having been converted to cubic consistently displayed a graphical error, as demonstrated by the module on the car in Figure 6.6. This problem was also present in the panoramas themselves gathered by the Amsterdam open dataset, as shown in Figure 6.7. So this may indicate problems in how this part of the dataset was constructed. This graphical error seems to be part of a specific vehicle. While this vehicle does cover other parts of Amsterdam, usually those parts are covered by other vehicles as well, and the faulty seed images are rejected by the Structure from Motion system. Unfortunately for Zuidplein, as it is on the edge of Amsterdam was only covered by one vehicle.

Whether this causes the Structure from Motion system to fail depends on whether the features extracted from the image are influenced, as this is the only feature taken from the seed image involved in our location estimation method. The Amsterdam open dataset likely uses multiple cameras, a common approach to taking panoramic images, and stitches them into a single coherent panorama view. Close inspection of the faulty panoramas Figure 6.7 shows that the graphical error occurs most evidently around vertical lines, strengthening the case for a bad stitch. If this bad stitch is caused by the bad alignment of the cameras or slight overlap in the camera view, this will cause a slight compression or expansion over the entire image. This bad stitching may therefore cause the features extracted for the Structure from Motion system to be different from those extracted from the manually gathered images introduced in subsection 4.1.3. Which would mean the manually gathered image would never be classified as part of the event area.

In the time between running the Zuidplein experiment and writing this research, the municipality of Amsterdam has released the 2020 panorama image dataset, which includes new Zuidplein panorama images. These were, however, not considered anymore for this work. A manual inspection does show that the same graphical error persists in this new dataset.

Overall, we have found a classification accuracy of 5% for images from the event area while finding a 100% classification accuracy for images not from the event area. As social media posts with geodata are sparse

compared to overall post counts, with only 0.71% of English tweets having coordinate geodata as determined by Huang and Carley[23]. Our approach could increase the amount of data available approximately sevenfold based on the 0.71% percent availability of coordinate geodata in English tweets given by Huang and Carley [23]. If the event takes place at a point of interest known to Twitter, this will reduce to only a two and a half times increase in the availability of data. At the same time, the proposed method has only a small chance of a false positive being included as exemplified by the 0% false positive rate overall in Table 5.4. Examples of this are the methods proposed by Gong et al. for density estimation[15] which rely on such sparse geo data for determining if a social media post was taken at an event area.

For the classification accuracy of images taken at the event area, it may be recommended in future research to perform a quantitative performance analysis to see under which specific combination of settings and features for the SfM phase. To see if the classification accuracy can be increased further using this approach. However, this quantitative performance analysis would not be recommended without the computation improvement introduced in subsection 7.2.1. Due to the nature of the quantitative analysis, this would multiply the required computational needs substantially. As for this research, two systems running 20 reconstructions between them in parallel took several months for the calculations to finish. Adding even more complexity, which would multiply computation time, would not be recommended.

We also determined the radii of confidence for the seven images that were correctly matched to the event area. These results do demonstrate that it is possible to determine the exact location, albeit with limited accuracy. The limited amount of measurements, however, makes it hard to generalize. What can be concluded, unless these measurements were particularly bad, is that based on these measurements, the method is unlikely to achieve an average error between 5 and 10 meters for either the Procrustes or outlier corrected Procrustes approaches. Particularly in the case of the Sumatrakade, it appears that the SfM method may have had trouble determining the rectangular shape of the event area. This may indicate issues in the used configuration of the SfM method.

Therefore, it is recommended for future work to do a quantitative performance analysis with respect to the used features and settings with a specific focus on the accuracy of the generated 3d structure. This analysis would, however, not be recommended without the computation improvement introduced in subsection 7.2.1 for the same reason as given earlier in the section.

6.1.1. Threats to validity

Here we will provide limitations and possible threats to the validity of the found results

- While different event areas were considered for the research, all of these were located in Amsterdam. It may be that Amsterdam features in the form of, for example, its choice of architecture some unique features different from other cities.
- For the seed images, we relied on the Amsterdam Open Dataset, which is unique to Amsterdam. Differences in similar datasets for other cities such as google streetview panorama data may provide other results.
- Reliance on self-gathered images instead of social media images that were taken at the event. While we did try to emulate the circumstances under which social media images are taken as addressed in subsection 4.1.3. It may still be that some factors of social media images have been overseen.
- All tested images were taken during good weather conditions during the day. Additional research may be needed to determine accuracy during the night time and during adverse weather.

6.2. Discussion on density estimation

For our second experiment, a true density of 0.1 people/m^2 and social media activity of 0.01 resulted in an MSE, MAPE, and MSE of 0.00423, 0.0423, and $2.25 * 10^{-5}$, respectively. Higher densities and higher social media activity quickly deteriorate, culminating in an error roughly equal to the estimate itself at a true density of 1 and social media activity of 0.1. The mean absolute error consistently grows as either the true density or social media activity increases or both.

This correlation of true density and social media activity with the estimate could indicate that the overcount correction applied by the methodology proposed in section 3.3 does not correctly address the overcount. Because if there is no overlap between views, we presume the count must be right, as argued in section 3.3. In fact, the actual densities converge to 0 for high amounts of overlap, which are present in high

Figure 6.1: Example of a social media image that was included in the "De Dam" reconstruction

Figure 6.2: Example of a seed image that was included in the "De dam" reconstruction

Figure 6.3: Example of a social media image that was not included in the "Zuidplein" reconstructions but should have based on location

Figure 6.4: Example of a seed image that was included in the "Zuidplein" reconstructions

Figure 6.5: Example of multiple camera setup being used by google to gather streetview panorama images. Multiple cameras are located in the blue ball on top of the car. The sick modules are for lidar, a technique used for depth mapping. Courtesy of [41]

Figure 6.6: Example of the graphic error for the Zuidplein event area as can be seen by the module on the car

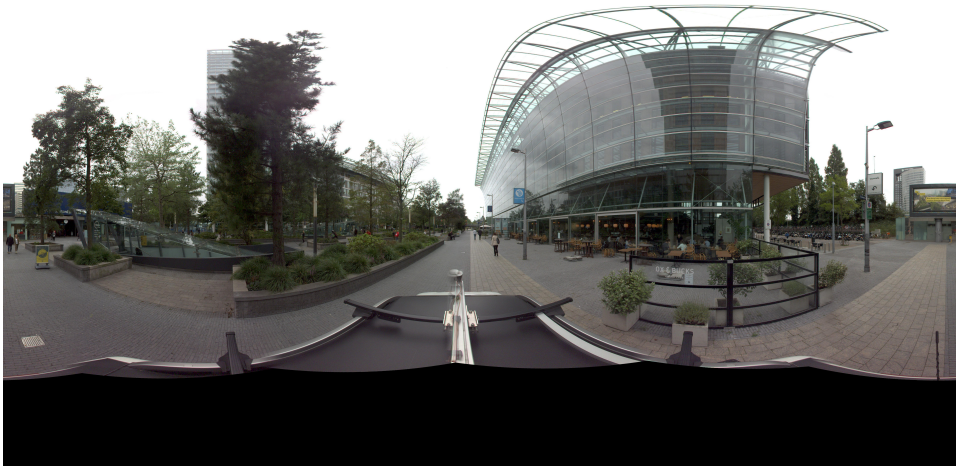


Figure 6.7: Example of the graphic error in the original Amsterdam open dataset panorama image as can be seen by the module on the car

density/high social media activity scenarios, as shown in Table 5.11. As shown in section 5.2, the results of the overcount correction do improve over those without overcount correction. But are currently over-correcting the overcount. More research would be necessary to see if weighting the overcount correction with the number of overlaps would improve overall estimation results.

Our system may be outperforming the non-infrastructure based methods proposed by Gong et al.[15] on lower densities if the amount of social media images put into the system is limited. With an MAE of 0.00423 - 0.279 for all values of true density 0.1-0.3 and social media activity of 0.01-0.03

This study's key limitation is that we used a simulation. Due to covid-19, no city events were held, and no historic dataset was identified that contained both location and rotation data. The usage of a simulation may mean that our method applied to reality may find different results. Another key limitation is our exclusion of errors from counting people in the image but instead assuming we know the exact amount of people in the image. While this is a reasonable assumption from the perspective of testing the error of our method in isolation. For applying our method to reality, such an error would be unavoidable. Therefore, it should be assumed that our results would become worse in a real-world experiment due to that additional error source. As discussed in subsection 4.1.4 for events where a more or less uniform distribution could be expected, such as Kingsday or protests. Our approach may not necessarily be suited to events where this is not the case, such as Sail. Due to the non-uniform distribution caused by the concentrated points of interest of the crowd in the form of the boats being observed. Here further research would be necessary to determine how these types of events would impact the estimation.

Future research that should come from this research is identifying if the over-correction of the overcount can be addressed, possibly improving results. Moreover, it is recommended to conduct a real-life experiment to see if the simulated data holds up compared to the real data.

6.2.1. Threats to validity

- Data provided by the crowd simulation may not match those of real-life events. In particular, events where a more or less uniform distribution is not to be expected, can be expected not to provide the same results as found in this research.

- Social media activity may only match to real life time frames as proposed in subsection 4.1.4 for too long time frames for density to stay more or less constant.

6.3. Discussion on emotion estimation

Our third experiment identified the emotions shown by faces in social media images taken at city events. Out of 236 faces, 25% were neutral, and 70% were happy. This result means that only 5% is either sad, surprised or showing fear (no faces were determined to have shown anger). This result could mean that detecting a large amount of sad, surprised, or fearful images could be a useful indicator of something being out of the normal. However, this assumption would need more research as currently, it is unknown what causes the detected emotions.

Gong et al.[16] found event-based social media posts are 62% positive, 13% neutral and 25% negative based on the text. Assuming the happy emotion matches the positive sentiment, neutral emotion matches the neutral sentiment, and the remaining emotions match the negative sentiment. Comparing these results, the happiness emotion and positive sentiment are comparable with 70% and 60%, respectively, a 15% difference. Neutral is 13% for our method, and 25% in the work done by Gong et al., a 50% difference. The largest difference is between the negative emotions being 5% and 25% for sentiment, an 80% difference. Overall, the text part of social media posts is more negative compared to the faces seen in images. This would make facial expressions a lot more suited to a potential outlier detector than the text part of a social media post.

Future research should focus on what causes these outlier emotions, such as fear, surprise, and sadness in social media images at city events. Future research could also focus on whether at outlier events that have had, for example, stampedes, these outlier emotions are indeed shown and can function as features in an outlier detection system. Finally, it is currently unknown what the actual emotions of the crowd are at city events. This experiment only focused on the faces shown in social media images at city events. Therefore it is unknown how these emotions compare to the actual emotions of crowds at city events.

6.3.1. Threats to validity

Here we will provide limitations and possible threats to the validity of the found results

- All labels were generated by the author, who is also a single person. While this can be acceptable for a first impression as emotions derived from facial expressions are generally agreed upon amongst humans. As demonstrated by the fact that humans use this in their non-verbal communications. A bias in the results can not be excluded. Crowdsourcing through a service such as Amazon Mechanical Turk with an appropriate consensus threshold would diminish this bias, but it is focused on too large a scale for the scope of this research.
- For our emotion results, all selected events are "positive" events such as Kingsday and Sail. These results may not hold up at more negative events such as protests, where the general mood of the event may be a lot more negative.

7

Conclusion and future work

7.1. Conclusion

This section will repeat the three sub-research questions and the answer we have derived for them in this research. We also repeat the main research question and the answer we have arrived at through our three sub research questions.

RQ1: How can the geographic location of a social media picture taken during a city-scale event be estimated?

We have identified a location estimation method given in subsection 3.2.4. This method allows determining whether an image was taken at an event area with a 5% chance, with an approximately 0% chance of generating false positives, i.e., an image being falsely classified as taken at the event terrain, based on results found in section 5.1.

We have also identified a method that allows determining the longitude and latitude of where an image was taken. The accuracy of this method could not be accurately determined due to the small number of measurements. However, this method likely requires further work to become adequately suitable for purposes such as being applied to the density method proposed in section 3.3.

RQ2: How can images be used to estimate *crowd density* from social media without crowd measurement infrastructure?

Images can be used to provide the number of people visible in a certain picture using the works by Rahmalan et al.[44], Jiang et al.[24] and Liu et al.[28] or YOLO[45] which are proposed in section 3.1. This can be integrated into the method introduced in section 3.3 to allow density estimation that for densities of 0.1-0.3 persons/m^2 and social media activity of 0.01-0.03 images/person has a Mean Absolute Error between 0.00423 and 0.279 while gradually worsening for higher values.

Using the location estimation method proposed in subsection 3.2.4 it is possible to generate additional data for the density estimation methods proposed by Gong et al.[15], as discussed in section 6.1. This additional data would likely improve those density estimation methods because the methods proposed that do not rely on physical infrastructure fail in multiple time windows due to lack of data.

RQ3: How can social media images be used to study the emotions expressed by participants of city-scale events?

Social media images can be used to study people's emotions visible in social media images taken during city events. This can be achieved by automatically extracting these city events' faces from the social media images and classifying the emotion shown. It was found that of those emotions shown in social media images taken at city events, 25% are neutral, and 70% are happy. These results may make emotions shown in facial expressions better than sentiment extracted from text in social media posts. However, it is still to be determined that the remaining 5% of sad, surprised, and fear showing faces could form part of an outlier detection system if it is demonstrated that these are indeed strongly correlated to outlier events.

MRQ: To what extent can social media images contribute to the estimation of the density and emotions of crowds during city events?

Social media images can contribute to the estimation of density and emotions by allowing the counting of people in the image and identifying their emotions through facial expressions. We have shown that social media images can improve the data available for the existing density estimation method proposed by Gong et al.[15]. If additional research improves the accuracy of the longitude, latitude location estimation method, our proposed density estimation method given in section 3.3 could still be shown to improve on existing density estimation methods that do not require infrastructure to function.

7.2. Future Work

This section gives a plausible computation optimization for the specific problem of doing multiple Structure from Motion (SfM) reconstructions with a common set of images shared between them. In addition to one or more images unique to each specific reconstruction.

7.2.1. Single image optimization

One of the main drawbacks of our first research was the extensive time cost of running the experiment with a full *Structure from Motion (SfM)* reconstruction being required for each image given a location. However, for every reconstruction n images are involved, of which n-1 are shared between all reconstructions. This overlap may lead to an order speed up from $O(m * n^2)$ to $O(m * n + n^2)$ of the image localization system, with n being the number of images used in the reconstruction and m the number of reconstructions. This optimization is also known as memoization and does not apply to Structure from Motion problems in general. This optimization only applies to the location estimation of an image using Structure from Motion.

The optimization may be achieved through the following concept. Opensfm specifically has five steps involved in a reconstruction.

- extract the metadata from the image. This operation only needs to be done once for all n-1 images shared between all reconstructions. By preprocessing these images, this step is only O(1), i.e., the need to extract the metadata from the reconstruction's unique image.

- detect features in the image. Similarly to extracting the metadata from the image by preprocessing the n-1 non-unique images, this step can be reduced to O(1).

- Match features of images. In this step, a pairwise comparison is made for each pair of images. This comparison is equal for all combinations of the lower triangle without diagonal with the n images on the row and column entry, as shown in Table 7.1. However, most of the pairwise combinations are shared between all reconstructions and can therefore be preprocessed. Only those cells with the new image as part of the combination pair, as demonstrated in Table 7.2 need to be computed. There are at most n-1 of these cells so that this step can be reduced from $O(n^2)$ to O(n).

- Create tracks of images. Similarly to matching the features of an image requires comparing the lower triangle without a diagonal. Can be optimized in the same way as matching features to be reduced to O(n).

- Reconstruct the 3d scene. Similarly to matching the features of an image requires comparing the lower triangle without a diagonal. Can be optimized in the same way as matching features to be reduced to O(n).

Table 7.1: example of precompute: Lower triangle operations to be done for precomputation. p means that the operations is precomputed while - means no operations is necessary for this example. For example 3,1 needs precomputation while 1,3 does not need to be computed as it is identical to 3,1.

Image #	**Image #**		
	1	2	3
1	-	-	-
2	p	-	-
3	p	p	-

Image #: The specific image being considered.
-: no computation is necessary for this combination of Image #.
p: for this step, the combination of Image # can be pre-computed.

Table 7.2: Example of precompute optimization: Introduction of image 4 necessary for determining the location of image 4 in the scene constructed based on image 1,2 and 3 requires n-1=3 extra computations given by c. The other computations have already been precomputed and are given by p.

Image #	**Image #**			
	1	2	3	4
1	-	-	-	-
2	p	-	-	-
3	p	p	-	-
4	c	c	c	-

Image #: The specific image being considered.
-: no computation is necessary for this combination of Image #.
p: for this step, the combination of Image # can be pre-computed.
c: for this step, the combination of Image # will need to be computed.

A single reconstruction is now O(n), as there are m reconstructions, this results in O(n*m). Finally, the precompute phase is $O(n^2)$. This combined should reduce the system to $O(m*n+n^2)$ from $O(m*n^2)$ with m the number of reconstructions and n the number of seed images. For real-time processing, if similar optimization can be done for graphical processor-based *SfM* methods. Such as those proposed by Frahm et al.[12]. Then close to real-time localization of images may be achievable for event area sized scenes.

Bibliography

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Netherlands Public Broadcasting. *Volle dam bij racismeprotest, geen anderhalvemeterboetes uitgedeeld | NOS*, 2020 (Accessed September 09, 2020). URL https://nos.nl/collectie/13842/artikel/2335843-volle-dam-bij-racismeprotest-geen-anderhalvemeterboetes-uitgedeeld.

[3] Donald M Cardina and Anastasios L Kefalas. System and method for imei detection and alerting, January 6 2009. US Patent 7,474,894.

[4] Aruna Chakraborty, Amit Konar, Uday Kumar Chakraborty, and Amita Chatterjee. Emotion recognition from facial expressions and its control using fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(4):726–743, 2009.

[5] Yan Chen, Jichang Zhao, Xia Hu, X. Zhang, Zhoujun Li, and Tat-Seng Chua. From interest to function: Location estimation in social media. In *AAAI*, 2013.

[6] Aron Culotta. Reducing sampling bias in social media data for county health inference. In *Joint Statistical Meetings Proceedings*, pages 1–12, 2014.

[7] Winnie Daamen, Serge P Hoogendoorn, and Piet HL Bovy. First-order pedestrian traffic flow theory. *Transportation research record*, 1934(1):43–52, 2005.

[8] Carlos F. Daganzo. Emerald, Inc., 2008. ISBN 978-0-08-042785-0. URL https://app.knovel.com/hotlink/toc/id:kpFTTO0003/fundamentals-transportation/fundamentals-transportation.

[9] Donald C. Cooper Event safety alliance. *The event safety guide*, 2020. URL https://www.eventsafetyalliance.org/the-event-safety-guide.

[10] Sylvain Faure. Cromosim, 2020. URL https://github.com/sylvain-faure/cromosim.

[11] U.S. Air Force. Gps.gov: Gps accuracy, 2020. URL https://www.gps.gov/systems/gps/performance/accuracy/.

[12] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building rome on a cloudless day. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 368–381, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15561-1.

[13] Vasavi Gajarla and Aditi Gupta. Emotion detection and sentiment analysis of images. *Georgia Institute of Technology*, 2015.

[14] geopy. geopy, 2020. URL https://github.com/geopy/geopy.

[15] Vincent X Gong, Jie Yang, Winnie Daamen, Alessandro Bozzon, Serge Hoogendoorn, and Geert-Jan Houben. Using social media for attendees density estimation in city-scale events. *IEEE Access*, 6:36325–36340, 2018.

[16] Vincent X. Gong, Winnie Daamen, Alessandro Bozzon, and Serge P. Hoogendoorn. Estimate sentiment of crowds from social media during city events. *Transportation Research Record*, 2673(11):836–850, 2019. doi: 10.1177/0361198119846461. URL https://doi.org/10.1177/0361198119846461.

[17] Sandra González-Bailón, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. Assessing the bias in samples of large online networks. *Social Networks*, 38:16–27, 2014.

[18] Google. Location | android developers, 2020. URL https://developer.android.com/reference/android/location/Location.

[19] J. C. Gower. Generalized procustes analysis. *Psychometrica*, 40(1):33–51, 1975.

[20] Eszter Hargittai. Potential biases in big data: Omitted voices on social media. *Social Science Computer Review*, 38(1):10–24, 2020.

[21] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[22] Dirk Helbing, Lubos Buzna, Anders Johansson, and Torsten Werner. *Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions*, volume 39. INFORMS, 2005.

[23] Binxuan Huang and Kathleen M Carley. A large-scale empirical study of geotagging behavior on twitter. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 365–373, 2019.

[24] X. Jiang, L. Zhang, P. Lv, Y. Guo, R. Zhu, Y. Li, Y. Pang, X. Li, B. Zhou, and M. Xu. Learning multi-level density maps for crowd counting. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8): 2705–2715, 2020.

[25] Kaggle. Challenges in representation learning: Facial expression recognition challenge, 2019. URL https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-data.

[26] Rout Jitendra Kumar, Choo Kim-Kwang Raymond, Dash Amiya Kumar, Bakshi Sambit, Jena Sanjay Kumar, and Williams Karen L. A model for sentiment and emotion analysis of unstructured social media text. *Electronic Commerce Research*, 18(1):1572–9362, 2018.

[27] J. Li, X. Qian, Y. Y. Tang, L. Yang, and T. Mei. Gps estimation for places of interest from social users' uploaded photos. *IEEE Transactions on Multimedia*, 15(8):2058–2071, 2013.

[28] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2018.

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[30] mapillary. Github - mapillary/opensfm: open source structure-from-motion pipeline, 2020. URL https://github.com/mapillary/OpenSfM.

[31] Bertrand Maury and Sylvain Faure. *Crowds in Equations: An Introduction to the Microscopic Modeling of Crowds*. World Scientific, 2018.

[32] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[33] Nitish Mutha. Equirectangular-toolbox, 2020. URL https://github.com/NitishMutha/equirectangular-toolbox.

[34] NASA. Earth fact sheet, 2020. URL https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html.

[35] Municipality of Amsterdam. Normkwaliteit panoramabeelden binnen het stelsel - stelselpedia, 2020. URL https://www.amsterdam.nl/stelselpedia/panorama-index/normkwaliteitpano/.

[36] Municipality of Amsterdam. *Home - Data en Informatie - Amsterdam*, 2020 (Accessed September 09, 2020). URL https://data.amsterdam.nl/.

[37] Municipality of Amsterdam. *Datapunt: Atlas API*, 2020 (Accessed September 09, 2020). URL https://api.data.amsterdam.nl/panorama/panoramas/.

[38] Municipality of Delft. *Aanvraagformulier evenementenvergunning*, 2020 (accessed Oktober 14, 2020). URL https://www.delft.nl/sites/default/files/2018-04/Aanvraagformulier%20evenementenvergunning_1.pdf.

[39] Municipality of Delft. *Voorbeeld veiligheidsplan in delft.pdf*, 2020 (Accessed September 09, 2020). URL https://www.delft.nl/sites/default/files/2018-02/Voorbeeld%20veiligheidsplan%20evenement%20in%20Delft.pdf.

[40] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[41] Padaguan. File: Google street view camera car.jpg - wikimedia commons, 2020. URL https://commons.wikimedia.org/wiki/File:Google_Street_View_camera_car.jpg.

[42] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137–2148, 2003.

[43] pew research. Mobile fact sheet, 2020. URL https://www.pewresearch.org/internet/fact-sheet/mobile/.

[44] Hidayah Rahmalan, Mark S Nixon, and John N Carter. On crowd density estimation for surveillance. 2006.

[45] Joseph Redmon. *YOLO: Real-Time Object Detection*, 2020 (Accessed September 09, 2020). URL https://pjreddie.com/darknet/yolo/.

[46] Thomas Richards. A review of software for crowd simulation, 2020. URL https://urban-analytics.github.io/dust/docs/ped_sim_review.pdf.

[47] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Visual Recognition*, 2020 (accessed Oktober 12, 2020). URL http://www.robots.ox.ac.uk/~vgg/research/very_deep/.

[48] Sergios Theodoridis and Konstantinos Koutroumbas. Pattern recognition. 2003. 2009.

[49] Ucrowds. Ucrowds - your crowd simulation solution, 2020. URL https://www.ucrowds.com/.

[50] V-count. *People counting | People counting systems | People counter device*, 2020 (Accessed Oktober 14, 2020). URL https://v-count.com/solutions/people-counting/.

[51] Alessandro Bozzon Vincent Gong, Winnie Daamen and Serge P. Hoogendoorn. Counting people in the crowd using social media images for crowd management in city events. *under review, WIS at TU Delft*, N.A.(N.A.):N.A., 2020.