Vision-based geolocation methods to assist outdoor navigation of persons with visual impairment

Renato Busatto Figueiredo

A thesis submitted for the
Degree of Doctor of Philosophy

University of East Anglia
School of Computing Sciences

March 2024

Abstract

Persons with visual impairment are affected daily by the lack of accessibility. In this thesis, we address the research question: Can we use computer vision techniques to improve the accuracy of geolocation estimation to potentially assist persons with visual impairment navigating outdoors? We analyse the requisites to create an outdoor visual navigation system and highlight the main problems involved. The main challenge identified is estimating an accurate user position. To tackle this problem, we detail the construction of the HLoc+SV, a vision-based geolocation method inspired by a version of the hierarchical localisation framework that exploits information from a set of geotagged street-view images. In a dataset of 58 pictures and 80 geotagged street-view reference images, HLoc+SV had a mean absolute geolocation error of 0.77 m (SD 0.41), while a smartphone GNSS receptor had a 12.09 m (SD 8.67) error. Nonetheless, the HLoc+SV is a potential solution suitable only for the scenario when the GNSS service is available and relatively accurate. When the GNSS service is unreliable or unavailable, we analyse a framework to geolocate an image using a GMCP-based image retrieval method combined with the Self Quotient Image (SQI) illuminance normalisation. We found out there is a degradation of 4 % on results compared to the original method when a geometric transformation is estimated by combining images with and without SQI. We also propose a method to isolate and measure the impact of changes in illuminants using a graph-morphological algorithm known as Sieve. We investigate the impact of using SQI on local features by segmenting images by levels of detail. We found that flat areas produced by the sieve have a positive effect on the detection of MSER blobs. MSER combined with SQI and sieve proved to be robust in matching street view images, increasing the matching score by 90 % in specific scenarios compared to SIFT features extracted from original images.

Contents

List of figures

List of tables

List of algorithms

Acknowledgements

First and foremost, I thank my supervisor Prof. Richard Harvey for his support and guidance over the last years. This doctoral research was an incredibly intense journey, and I am grateful he has given me confidence in my own abilities. I also thank my Viva examiner, Dr. Christos Efstratiou, for his helpful suggestions and comments on this thesis.

Thanks to my Zen Buddhist teacher, Reverend Master Leoma Hague, to whose compassion and wisdom I can only aspire. I am and will be forever grateful for her invaluable teachings.

A special thank you to my family and friends who unconditionally supported me during my research and thesis writing.

Finally, I fully appreciate the generous funding by the Brazilian National Council for Scientific and Technological Development (CNPq) that made this research possible.

To my mum,

for her love and dedication.

Chapter 1

Introduction

Disability has different meanings in different communities and its definition has been changing over the centuries. Currently, the British Equality Act 2010 defines disability as ‘a physical or mental impairment that has a substantial and long-term negative effect on the ability to do normal daily activities’ [145]. The World Health Organisation broadens this understanding, defining disabilities as ‘an umbrella term for impairments, activity limitations, and participation restrictions, denoting the negative aspects of the interaction between an individual (with a health condition) and that individual’s contextual factors (environmental and personal factors)’ [163]. These modern definitions emphasise that disability is not solely a medical condition but is also shaped by social, economic, and cultural factors.

Approximately 2.2 billion persons in the world who are blind or partially sighted have difficulties engaging in activities that involve social relations, which affects their process of socialisation [137, 164]. Transportation is one of the greatest barriers and a major challenge for persons with visual impairment.

1.1 Autonomous navigation of persons with visual impairment

The Royal National Institute of Blind People UK (RNIB) shows that transportation is a major challenge [116]. Those who became blind in adulthood, especially those who used to drive, find life too limiting without the flexibility that a car provides. Many use taxis, but their cost is a limiting factor. Only a minority of the persons with visual impairment report using public transport. This option is most commonly used by persons who are blind from birth and have become more confident in using it over the years. Even so, there are challenges in the various stages of using public transport – from getting updated information, often not having anyone to ask; identifying the correct transport arriving; knowing the exact place of getting out; until the arrival itself, and finally, accessing a public venue.

Autonomous mobility is affected not only by transport availability, but also by difficulties in walking outdoors, including crossing streets and locating the final destination. There are, for example, risks of accidents and difficulty in identifying obstacles on their way. People commonly stumble across road signs and architectural barriers [116].

Although persons with visual impairment already have access to some technology to assist them with professional and daily tasks, huge barriers in transport and locomotion often prevent them from obtaining formal education [116]. Difficulties to arrive at the workplace, school or university creates a vicious circle: persons with visual impairment do not get a university degree because they cannot get a job and they cannot get a job because they do not have a degree. They then end up being excluded from society because pursuing a technical or undergraduate degree to get a job is extremely challenging.

Historically, canes have been the easiest and most widely used tool to detect obstacles outdoors, mainly due to their affordability and portability [116]. Using a cane, it is possible to detect obstacles at ground level by scanning the area in front of them. The tool helps to discover barriers such as holes, stairs, steps, and walls, but it fails to detect obstacles in movement, over the knee level, or beyond a 30 cm to 60 cm range. These kinds of obstacles sometimes can be detected only when they are dangerously close to the user.

Dog guides are a common alternative to canes [116]. They are competent in guiding and providing a good detection range, but they cannot avoid risky barriers at the head level. Furthermore, guide dogs typically work for an average of six years and it requires regular expenses and relevant changes in the user's lifestyle [82].

Navigation Assistance for the Visually Impaired (NAVI) refers to systems that assist or guide persons with visual impairment through audible instructions [6]. Most of the systems found in the literature focus on the detection of obstacles. Few solutions have been proposed to improve outdoor mobility and safety for persons with visual impairment. Analysing the NAVI systems present in the literature, it became evident that for most people they are not viable [55, 123]. In general, the users need to carry complex hardware systems and, in some cases, the environment also has to be somehow prepared beforehand. Thus, part of this study involves analysing the requirements of a solution that integrates all outdoor journey stages, using new technologies to potentially support and improve the well-being of persons living with such conditions.

A key aspect of a safe NAVI system is that it must estimate the user's location with high accuracy in real-time. At first glance, this problem might seem solved with the use of GNSS. However, the GNSS accuracy is approximately 5 m to 10 m in urban areas [148], which does not even allow identifying on which side of the street the user is. It also lacks directional information, usually provided by additional sensors.

A navigation system for persons with visual impairment must also consider the case when the GNSS receptor information is not reliable or completely nonexistent. For the application studied, a GNSS outage or signal obstruction represents a life threat to the user.

Therefore, the use of GNSS alone is not suitable for safe and autonomous pedestrian navigation of persons with visual impairment. The geopositioning accuracy must be greatly improved before making such a system available to the public. There are plenty of methods aiming to improve GNSS receptor accuracy, but they usually involve bulky, expensive and slow equipment.

1.2 Vision-based geolocation

With advances in computer vision algorithms, a camera is a potential ally to accurately geolocate someone in real-time at a low cost. The vision-based geolocation problem is a key topic in the field of computer vision and a major challenge for researchers all over the world. Many practical applications require automatic, accurate, and fast visual recognition outdoors, such as autonomous vehicle driving, pedestrian navigation, and robot path planning.

The idea of using a camera to navigate is not new. Simultaneous Localisation and Mapping (SLAM) is a technique used in computer vision to enable an autonomous agent to build a map of an environment while simultaneously tracking its own direction and location in local coordinates. The agent usually uses sensors such as Light Detection and Ranging (LiDAR) and Inertial Measurement Units (IMU) to collect data about the environment. More recently, camera-based SLAM systems, called Visual SLAM (V-SLAM), have gained more attention due to their low cost and rapid development of powerful computer vision algorithms [14, 77]. SLAM and V-SLAM are particularly useful when there is no environment map available in advance, e.g. in search and rescue operations or exploration missions. Although a few georeferenced SLAM systems have been proposed [21], usually SLAM and V-SLAM do not attempt to locate the agent in world coordinates.

Vision-Based Localisation (VBL) is a problem closely related to V-SLAM that focuses on estimating the geolocation and orientation of an agent using visual information only (e.g. pictures or video frames). Unlike SLAM and V-SLAM, which estimate a relative pose, VBL techniques retrieve the absolute pose of a camera using a reference world coordinate system. The process typically involves matching visual features extracted from the camera (query images) with features found in a dataset (reference images). Differences found in photographs of the same scene taken a long time apart may be significant.

This problem may also puzzle humans to some extent. Visiting a city after many years, for example, can be challenging: weather changes, buildings are refurbished over time, facades are covered during maintenance works, old stores give place to new ones, and pavements are extended. The appearance of buildings changes in all sorts of ways over the course of years.

The light also plays an important role in computational vision-based scene understanding. Shadows can decrease the performance not only of image geolocation, but also scene analysis, image segmentation, object recognition and shape reconstruction [41]. Therefore, using local image descriptors invariant to illuminance seems a good starting point for solving this problem. The accuracy of vision-based localisation systems can be improved with a better understanding and manipulation of shadows, although automatically detecting and removing shadows from images is known to be an extremely challenging task.

Other factors such as low image resolution, scenes photographed from different points-of-view and occlusion are also crucial for matching images. All of these changes affect both the detection of consistent features and the ability to match features when their appearance is substantially distinct.

There are studies on the impact of specific changes on images [99], but when we compare street-view images taken years apart, all these aspects occur simultaneously. The change of point of view in the same scene is handled by SLAM algorithms [7, 46]. Partial occlusion of a scene, changes in the appearance of buildings, or even minor changes in illumination are well handled by local feature detectors and descriptors. Neural networks using bottleneck layers can process small-scale information on high-resolution images. A much greater challenge arises when some or all of those aspects are combined.

1.3 Motivation and aim

This study aims to tackle the problem of outdoor geolocation estimation employing vision-based methods. While GNSS technology has revolutionised outdoor geolocation, it has limitations in dense urban areas where the satellite signal is subject to interference or can be completely blocked.

Vision-based geolocation methods offer an alternative approach that can be used in areas where GNSS is not available or inaccurate. These methods rely on cameras and computer vision algorithms to determine the location of a person or object by analysing the visual features of the environment, handling variations in illumination, weather conditions, and environmental factors.

In this study, we consider two scenarios: when the GNSS service is available and when it is not. For urban areas with GNSS coverage, we refine the user's GNSS geoposition information using a hierarchical localisation framework with geotagged street-view images and a Structure from Motion (SfM) model built on-the-fly. For the more challenging scenario where the GNSS service is not available or unreliable, we analyse the use of a GMCP-based geolocation method combined with Self-Quotient Image (SQI) illuminance normalisation. The results obtained by this last experiment led us to conduct a careful investigation on the impact of illuminance normalisation with SQI on the accuracy and repeatability of SIFT, SURF, and MSER detectors and descriptors. We use a graph-morphological algorithm known as sieve to analyse differences in local features detection between original and SQI images by progressively eliminating image details.

1.4 Research question

In this thesis, we ask: 'Can we use computer vision techniques to improve the accuracy of geolocation estimation to potentially assist persons with visual impairment navigating outdoors?'

There are several relevant points for why we find this topic of interest:

1. The GNSS geolocation service is mistakenly seen as more accurate and reliable than it often is.

2. A solution to guide persons with visual impairment places much higher geolocation accuracy requirements than other applications.

3. The rapid advances in computer vision ought to feed into better vision-based geolocation systems.

1.5 Thesis structure

This thesis outline is as follows. Chapter 2 presents a review of state-of-the-art techniques of NAVI systems, focusing primarily on the equipment used, their features, and limitations. We also present a brief description of image processing techniques relevant to image geolocation estimation and deep-feature extraction. In Chapter 3, we analyse the requirements of a vision-based NAVI system. In Chapter 4, we propose and evaluate a method for improving the GNSS geolocation using a single camera and a set of street-view reference images. In Chapter 5, we study a method to estimate the user geolocation when there is no GNSS service available using a GMCP-based image retrieval geolocation estimation and SQI illuminance normalisation. In Chapter 6, we investigate the impact of normalising illuminance in images with the Self-Quotient Image (SQI) filter on the accuracy and repeatability of local feature detectors and descriptors using a graph-morphological algorithm known as sieve. Finally, conclusions and perspectives of future work are drawn in Chapter 7.

The Appendix A presents an introduction to image formation and light normalisation techniques. Appendix B brings an overview of the graph-morphological sieve algorithm. Finally, Appendix C shows examples of SIFT, SURF and MSER features detected in images at progressive scales of detail using sieve, with and without SQI illuminance normalisation.

1.6 Statement of originality

Unless otherwise noted or referenced in the text, the work described in this thesis is that of the author. The following aspects of this work are considered novel:

1. Requirement analysis of vision-based outdoor navigation assistants for persons with visual impairment (Chapter 3)

2. Development of a GNSS geolocation refinement method using a single camera and a set of geotagged street-view reference images (Chapter 4)

3. Use of the SQI illuminant normalisation associated to a GMCP–based image matching method as an attempt to geolocate street-view images (Chapter 5)

4. Creation of the structured Anne Frank House image dataset (Chapter 6)

5. Use of sieve as a proxy to evaluate the performance of local feature detection and description at different levels of detail, as well as the impact of using SQI on this process (Chapter 6)

1.7 Contributing publications

The following publication has been produced by the work in this thesis:

- R. Busatto and R. Harvey, ‘Outdoor Navigation Assistants for Visually Impaired Persons: Problems and Challenges,’ *Journal on Technology and Persons with Disabilities*, vol. 10, pp. 184–205, 2022.

Chapter 2

Literature review

Part of the content of this chapter features in the following published article:

- R. Busatto and R. Harvey, ‘Outdoor Navigation Assistants for Visually Impaired Persons: Problems and Challenges,’ *Journal on Technology and Persons with Disabilities*, vol. 10, pp. 184–205, 2022.

2.1 Background

The perspective on disability has evolved over time. Throughout the history of civilisation, limited understanding and acceptance of human differences led to segregation, exclusion, and even extermination of minority groups.

In ancient Greece, it was common to kill newborn babies and children for various reasons. In Figure 2.1, for example, the Greek semi-goddess Medea is depicted killing one of her sons. Newborns with deformations or disabilities were murdered. Their view of disabilities was less about the individual’s capacity to lead a regular life, but more aesthetic [141]. The same word πεπηρωμένον (pepēromēnon) was used to describe both a severe wound and a newborn baby with a clear physical impairment. In Sparta, they used to examine their infants for any indication of fragility or abnormality; if they did show any of these signs they were thrown from an abyss [141]. In Athens, there was a ceremony called Ἀμφιδρόμια (Amphidromia), translated as ‘running around the circle’, in which a newborn baby was carried around by family members to be inspected. In Plato’s Theaetetus [121, s. 160e], he uses this ceremony as an analogy to eliminate bad arguments. Most ancient texts addressing disability follow the concept of exposure, which is now recognised as infanticide [141].

Nowadays, some societies still sacrifice persons who are born with disabilities. Members of the Brazilian tribe Yanomami, for example, sacrifice newborns in whom serious health problems are identified [117]. There are also reports of murders of persons with disabilities by relatives who considered them to be heavy burdens [168]. Such murders are rooted in superstition and ignorance. Although these cruel and unjustifiable acts go violently against basic human

Figure 2.1: Greek semi-goddess Medea killing one of her sons. Side A from a Campanian (Capouan) red-figure neck-amphora, ca. 330 BC. From Cumae, Italy. Photographer: Bibi Saint-Pol. Wikimedia Commons (public domain).

rights, they are often done in the name of tradition or religious beliefs, considered part of the culture, or merciful.

According to Judith Butler, people who deviate from the normative standards of society are often not conceived as humans. In Frames of War, Butler raises questions about this precariousness of life to which we are all exposed:

> There are "subjects" who are not quite recognizable as subjects, and there are "lives" that are not quite—or, indeed, are never—recognized as lives. In what sense does life, then, always exceed the normative conditions of its recognizability? To claim that it does so is not to say that "life" has as its essence a resistance to normativity, but only that each and every construction of life requires time to do its job, and that no job it does can overcome time itself. In other words, the job is never done "once and for all." [25, p. 4]

Butler makes clear that some people (i.e. subjects) are not recognised as people, therefore their lives are perceived as less worthy. This is the root cause of human rights violations: we discriminate between lives worth living and lives that can be ended, eliminated, or neglected to death. Persons with some kind of impairment claim their humanity every time their rights are flouted, often on a daily basis.

Even when the infanticide of newborns with disabilities has been overcome, adults with disabilities continue to be marginalised, undervalued, and hidden from social life. According to the World Health Organisation [163], persons with disabilities have more limited access to healthcare, lower formal education, lower income and higher poverty rates compared to the general population. Persons who are disabled often experience huge barriers to accessing services that most people take for granted, such as health services, education, employment, transportation, and information. The urban environment is often designed for persons with no disability, or at best, only the most common disabilities are considered, and when it does not imply a high cost of implementation.

2.2 Navigation assistance for persons with visual impairment

Within this complex landscape, the importance of wayfinding for individuals with disabilities becomes evident. The British RNIB Wayfinding Project [165] breaks down this task into key journey stages – walking, catching a transport (bus, train, tube, ferry, plane), and navigating within a building. Walking is the most important stage that connects the other stages of the journey, yet it is the one with the least amount of information or assistance. These stages are further refined into activities and actions following four principles of wayfinding: getting information and using it, orientating within the environment, navigating within the environment, and identification of entrances and exits.

Many studies have been conducted to develop equipment and technology to assist autonomous navigation of persons with visual impairment, known as Navigation Assistance for Visually Impaired (NAVI) systems. In this chapter, we review the main NAVI systems available, primarily focusing on the equipment used, their features, and limitations. Table 2.1 outlines the systems reviewed along with the equipment used and navigation abilities.

The Table 2.1 covers a variety of systems and technologies designed to assist persons with visual impairments in navigating indoors and outdoors. These systems can be broadly categorised into three groups based on their functionalities: remote human assistance, sensor-based navigation, and smartphone-based navigation. In this section, we analyse the differences between the various technologies proposed, equipment used, and sensors embedded in these systems.

Table 2.1: Overview of NAVI systems including main features, sensors used and references. (continue)

System, year [ref.]	Equipment	Obstacle detection	Object identif.	Indoor path	Outdoor path
Aira Explorer, 2015 [74]	Remote human agent, smartglasses, smartphone, camera, GNSS	✓	✓	✓	✓
Be My Eyes, 2015 [161]	Remote human agent, smartphone, camera	✓	✓	✓	✓
Drishti, 2004 [122]	Computer, GNSS, wifi, sonar	✓	-	✓	✓
Wang et al., 2014 [158]	RGBD camera	✓	✓	-	-
Wang et al., 2017 [155]	RGBD camera, computer, haptic	✓	✓	-	-
ISANA, 2016 [85]	Tablet, RGBD camera	✓	-	✓	-
NAVIG, 2010 [73, 76]	Computer, GNSS, stereoscopic camera, motion tracker	✓	-	-	✓
ODILIA, 2008 [96]	Computer, mobile phone, GNSS, infrared, dead reckoning device	✓	-	-	✓
WaveOut, 2021 [37]	Smartphone, GNSS, camera	✓	-	-	✓
Koley and Mishra, 2012 [80]	GNSS, sonar	✓	-	-	✓[a]
BlindSquare, 2012 [20]	Smartphone, GNSS, compass, bluetooth	-	-	✓	✓
Wayfindr, 2017 [159]	Smartphone, bluetooth	-	-	✓[b]	✓[b]

Table 2.1: Overview of NAVI systems including main features, sensors used and references. (continued)

System, year [ref.]	Equipment	Obstacle detection	Object identif.	Indoor path	Outdoor path
Agrawal et al., 2017 [4]	Sonar, GNSS, GSM	✓	-	-	-
Aladrén et al., 2014 [6]	RGBD camera, infrared, RFID	✓	-	-	-
Ifukube et al., 1991 [69]	Sonar	✓	-	-	-
Kanwal et al., 2015 [75]	RGBD camera, infrared	✓	-	-	-
RG, 2005 [81]	Computer, RFID, sonar	✓	-	-	-
Mahmud et al., 2014 [92]	Sonar	✓	-	-	-
G4B, 2017 [93]	Sonar, infrared	✓	-	-	-
Nandhini et al., 2014 [104]	GNSS, RFID, sonar	✓	-	-	-
Tapu et al., 2014 [142]	Camera	✓	-	-	-
Smart Cane, 2011 [152]	Sonar, water detector	✓	-	-	-
Cydalion, 2016 [45]	Smartphone, camera	✓	-	-	-
EYECane, 2009 [71]	Computer, camera	✓	-	-	-
Orcam, 2013 [103, 110]	Wearable camera	-	✓	-	-
NavCog, 2016 [5]	Smartphone, wifi, bluetooth	-	-	✓	-

Table 2.1: Overview of NAVI systems including main features, sensors used and references. (continued)

System, year [ref.]	Equipment	Obstacle detection	Object identif.	Indoor path	Outdoor path
Chaccour and Badr, 2015 [28]	Smartphone, bluetooth, wifi, surveillance cameras, head marker	-	-	✓	-
Fusco and Coughlan, 2020 [47]	Smartphone, camera, gyrocompass	-	-	✓	-
Jain, 2014 [70]	Smartphone, RFID	-	-	✓	-
VI-Navi, 2011 [97]	RFID, bluetooth, compass	-	-	✓	-
Nassih et al., 2012 [105]	RFID	-	-	✓	-
Öktem et al., 2008 [108]	RFID, compass	-	-	✓	-
RightHear, 2015 [125]	Smartphone, bluetooth	-	-	✓	-
Nearby Explorer, 2013 [9]	Smartphone, GNSS	-	-	-	✓
Brusnighan et al., 1989 [22]	GNSS	-	-	-	✓
Victor Reader, 2017 [68]	GNSS	-	-	-	✓
Voice Maps, 2010 [72]	Computer, GNSS, gyrocompass, keyboard	-	-	-	✓
Voice Helper, 2015 [86]	Smartphone, GNSS	-	-	-	✓
Seeing Eye, 2013 [135]	Smartphone, GNSS	-	-	-	✓

Table 2.1: Overview of NAVI systems including main features, sensors used and references. (continued)

System, year [ref.]	**Equipment**	**Obstacle detection**	**Object identif.**	**Indoor path**	**Outdoor path**
BrailleNote GPS, 2002 [134]	GNSS	-	-	-	✓
PGS, 2005 [87]	Computer, GNSS, compass	-	-	-	✓
MOBIC, 1996 [118]	Handheld computer, mobile phone, GNSS, compass	-	-	-	✓
Ariadne GPS, 2011 [31]	Smartphone, GNSS	-	-	-	✓
iMove, 2013 [40]	Smartphone, GNSS	-	-	-	✓
MyWay Classic, 2012 [140]	Smartphone, GNSS	-	-	-	✓
Seeing Assistant, 2013 [147]	Smartphone, GNSS	-	-	-	✓
ViaOpta Nav, 2014 [107]	Smartphone, GNSS	-	-	-	✓
Loadstone GPS, 2004 [78]	Smartphone, GNSS	-	-	-	✓
Corsair GPS, 2016 [138]	Smartphone, GNSS, compass	-	-	-	✓
Lazarillo, 2016 [38]	Smartphone, GNSS, compass	-	-	-	✓
PocketNavigator, 2010 [120]	Smartphone, GNSS, compass	-	-	-	✓
OsmAnd, 2010 [111]	Smartphone, GNSS, compass	-	-	-	✓
NAVIGON, 2011 [48]	Smartphone, GNSS, compass	-	-	-	✓

Table 2.1: Overview of NAVI systems including main features, sensors used and references. (continued)

System, year [ref.]	Equipment	Obstacle detection	Object identif.	Indoor path	Outdoor path
ARGUS, 2014 [27, 112]	Smartphone, GNSS, dead reckoning device	-	-	-	✓[a]
Lakehal et al., 2020 [83]	Smartglasses, smartphone, GNSS	-	-	-	✓[a]
Soundscape, 2018 [143]	Smartphone, camera	-	-	-	✓[a]
Dharani et al., 2012 [34]	RFID	-	-	-	-
Gulati, 2011 [59]	GNSS	-	-	-	-
TANIA, 2009 [67]	Tablet, GNSS, RFID, inertial sensor	-	-	-	-

[a] Path has to be recorded beforehand [b] Just on underground stations and selected locations

2.2.1 Remote human assistance

Systems based on connection with a remote human agent [74, 161] connect a user with visual impairment to a sighted remote agent. When the user requests assistance, the system makes a video call so that the agent can talk to the user and help them. Both mentioned systems share the image captured by a camera and audio in real-time. Be My Eyes [161] is a free smartphone app that requires the user to point their device to the scene or object, which can be challenging for persons with an impairment. The Aira Explorer [74] provides smart glasses with a camera, allowing agents to see from the user's point of view, as well as access the user's geolocation and compass direction.

Using remote assistance, users can request help and interact using natural spoken language. The scenarios covered are varied – users could ask, for example, for assistance to shop, read books, cook meals, or navigate to unknown locations. This type of app is used on a daily basis by some users [35].

Although remote human assistance can be versatile, a reliable internet connection must be available during the video call. The interruption or delay of calls poses a great risk, such as when there is traffic nearby or when the user is crossing a street. No call history is available to remote agents – information provided by users on previous calls is not stored. Furthermore, disclosing what users are seeing or doing to remote agents can be undesirable and embarrassing. Users report that they do not feel safe to disclose where they are going to strangers, neither in person nor in a video call [13, 162]. It further raises privacy and legal concerns, since agents and users may be in different jurisdictions with different mores. For these reasons, we do not consider using remote human assistants as viable for truly autonomous navigation.

2.2.2 Global Navigation Satellite Systems

Global Navigation Satellite System (GNSS) refers to any positioning and navigation services provided by a constellation of satellites on a global or regional basis. The most well-known GNSS service is GPS, owned and managed by the United States. Nevertheless, other nations have also developed their own independent systems. The main ones are Galileo (European Union), QZSS (Japan), GLONASS (Russia), BeiDou (China) and NavIC (India) [149].

Back in 1989, when the GPS service was made available to civilians by the United States government, Brusnighan et al. [22] proposed an outdoors NAVI system that employed such technology. Since then, most proposed NAVI systems have made use of GNSS services [9, 20, 22, 27, 30, 31, 37, 38, 40, 48, 68, 72–74, 76, 78, 80, 83, 86, 87, 96, 107, 111, 118, 120, 122, 134, 135, 138, 140, 147]. In general, the user sets a destination, and a route is calculated using the user's geolocation, points of interest and a map with allowed pedestrian paths. They are usually operated through smartphones or dedicated mobile devices with an internet connection.

Although GNSS systems are a global solution for geolocation, there are some challenges to using them on NAVI systems. Their horizontal accuracy of approximately 5 m to 10 m [148] makes it impossible to safely guide a pedestrian with acceptable precision. In places with many obstructions, such as metropolitan areas or inside buildings, obstructions can block satellites' signals to the extent that the receiver is not able to calculate its position.

2.2.3 Camera and computer vision algorithms

Digital cameras are used in this context to connect helpers by video call [74, 161], detect obstacles [6, 45, 71, 73, 75, 76, 85, 142, 155, 158] or recognise objects [103, 110, 155, 158]. Most research projects use common cameras, including the ones built in smartphones [45, 71, 74, 103, 110, 142, 143, 161]. A few projects make use of more complex cameras with depth sensors [6, 73, 75, 76, 85, 155, 158].

Cameras are cheap components, compact and easy to maintain, and are widely available on smartphones and laptops. When associated with computer vision algorithms, it becomes possible to perform tasks such as reading signs, labels, and texts, identifying colour information, objects, people or cash. Although distinguishing between close and far objects with a single camera is possible, this task is not trivial. Usually, extra sensors are used to accomplish this task, e.g. stereo or RGBD cameras; ultrasonic, Bluetooth, or infrared devices.

Computer vision algorithms have been advancing since the last decade. Complex algorithms now can run in real-time on smartphones and wearable devices due to the miniaturisation of hardware components and an increase in processing power, storage capacity, and faster mobile internet connection. Computer vision algorithms are easier to reproduce and can interpret real scenes locally without human interaction, allowing greater privacy for the user.

Few solutions exploit the potential of computer vision algorithms. OrCam [110], for example, aims to recognise labels, products, text and other objects close to the user. It is not clear whether they assist in outdoor navigation. Google has announced a visual navigation system, available at selected locations and integrated into their map app [124]. Although it is not specifically designed for persons with visual impairment, they use augmented reality and computer vision algorithms to match images taken in real-time from the user's smartphone with a dataset to improve their geolocation estimation. The path is then shown on the screen using augmented reality.

Computer vision algorithms rely on visual appearance to detect obstacles and objects. They are therefore sensitive to factors that change the visual appearance of scenes, such as illumination, point of view, and occlusion. Internal factors such as processing power and trained models also affect accuracy. Some classes of objects are well studied and present a high detection accuracy, while others need more study and larger datasets for training purposes.

2.2.4 Ultrasonic range sensor

Ultrasonic sensors use sound propagation to measure the distance to objects in a short range. Widely available Raspberry Pi sensors, for example, work at about 2 cm to 400 cm [1]. They are cheap and do not need preparation of the environment beforehand.

Although ultrasound pulses propagate in three dimensions, the distance information is unidimensional. It is possible to combine sensors pointing in different directions, but this approach can be problematic when performing more complex tasks, such as measuring long-range distances or the shape of objects. On NAVI systems, ultrasonic sensors may help to avoid obstacles, but they cannot help the user reach a point of interest following a proposed path. They also need to be attached to the main processing device, as it is not usually embedded on laptops or smartphones.

2.2.5 Bluetooth and infrared beacons

Bluetooth and infrared beacons are cheap devices used as tags, installed on the environment beforehand. A receptor detects nearby beacons and reads the information transmitted by them. These devices receive contextual information in indoor environments and can infer their self-location if the exact position of the beacons is known. Their maximum range is limited to 100 cm [130] and infrared beacons usually need to be aligned with the receptor.

Bluetooth technology has the advantage of being embedded on mainstream devices like laptops and smartphones, and can also be used to connect compatible wireless devices. However, Bluetooth beacons are not suitable for navigation in unprepared outdoor areas that are unknown to the user.

2.2.6 Inertial instrument

Inertial instruments use motion and rotation sensors to trace the route of a moving object. Their main advantage is that there is no need for external references. Although precise inertial instruments are complex and expensive, most smartphones have basic motion and rotation sensors. Precise inertial instruments are particularly useful on NAVI systems when there is no access to the GNSS signal, temporarily or permanently.

The errors of the calculated position using inertial instruments are cumulative and increase over time. Even the best accelerometers would accumulate about 50 m errors within 17 min [60]. Therefore, the position must be periodically corrected by other sensors or by the navigation system. Guo et al. [60], for example, created a device embedded on a boot that corrects the user's position at every step, achieving an accumulative error of 4 m after an 85 min walk.

2.2.7 Compass

Electronic compasses rely on the magnetic field of the Earth to identify the orientation of the user. They are useful when combined with GNSS or other positioning systems. Their measurements must be carefully considered, as they can be affected by any magnetic field other than Earth's. Most smartphones have a built-in electronic compass.

2.2.8 Internet connection

An internet connection allows systems to make use of the processing power of a server, especially when more complex algorithms are needed. For a real-time experience, processing must be done locally as much as possible.

Ideally, outdoor NAVI systems should function without an internet connection, as it is not reliable when outdoors. Nonetheless, it can be used to fetch updated information about maps, roads, pavements and available routes.

2.3 Computing devices

Smartglasses

Smartglasses are wearable devices that have limited computing power and usually connect wirelessly to the main processing device, e.g. a smartphone. Common components are the camera, visual interface projected on the glasses' lens, microphone, and speaker. The camera gives the user's point of view (PoV). Hands-free interfaces have been reported to be more intuitive for users with or without visual impairment, especially when speech recognition and audio feedback are combined [83, 151].

Smartphone and tablet

Smartphones and tablets are multipurpose mobile computing devices that integrate several electronic components. Common components in conjunction with a mobile network transceiver are wifi transceiver, camera, inertial sensor, Bluetooth, compass, GNSS, and USB connection. The user interface includes a touchscreen display, sound speaker, microphone and vibration motor. Although less portable, tablets usually have more processing power and bigger screens.

When used for NAVI systems, smartphones and tablets are convenient out-of-the-shelf mobile options with a variety of useful sensors for outdoor navigation and communication. It is possible to connect other sensors and devices by Bluetooth or USB interface. Persons with visual impairment can use smartphones and tablets when a screen reader is built into the operating system, e.g. Apple's VoiceOver and Android's TalkBack. Although some proof-of-concept NAVI projects require the user to connect smartphones and tablets, these devices were not designed to be worn.

Handheld

Handheld devices are similar to smartphones but have physical buttons, less computing power, and generally no touchscreen display. Although physical buttons are preferred by users with visual impairment, the limited computing power and lack of more advanced components make it less convenient to use them as a platform for NAVI systems.

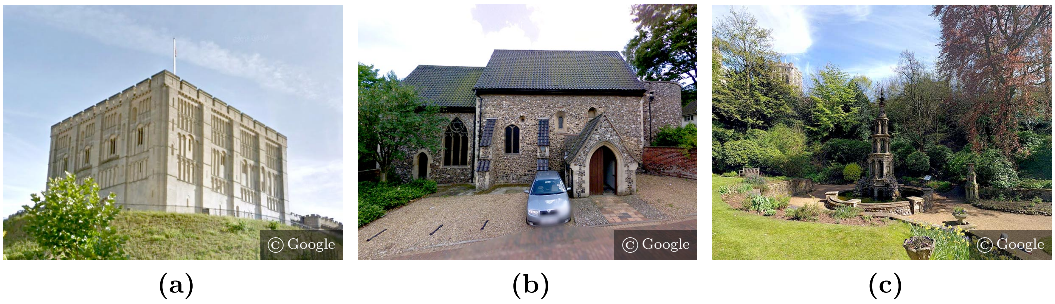
Laptop

Laptops have more processing power available compared to smartphones and tablets, but the core components of NAVI systems are not integrated, e.g. GNSS, compass, inertial sensor, and mobile network transceiver. Laptops are widely used for proof of concept of NAVI systems, usually carried by the user in a backpack. They are unlikely to be adopted as an end-user solution.

2.4 Image processing techniques for geolocation

Image geolocation is the problem of determining the location on Earth where a particular image was captured. Can the origin of a photograph be identified exclusively based on its pixels? Geolocating a photo even within a city can be challenging. Consider the photographs in Figure 2.2, taken in Norwich. The first one (a) is easy – it is the Norwich castle. The second picture (b) looks like one of the many medieval English churches. The last photo (c) is the most challenging. Probably, all that could be said is that it is a formal garden.

The online game Geoguessr.com brings this challenge to another level. In this game, the player is placed on a street somewhere in the world and they have to guess where it is. The player has a 360° view of the street and can walk



(a) **(b)** **(c)**

Figure 2.2: Are you able to say where these photos were taken?

along the road. In the absence of obvious and notable landmarks, humans use their own experience and multiple cues to guess the location of a photo. Street signs, plantations, weather, written texts, building styles and people's clothes can narrow down the number of possible locations. Traditional computer vision algorithms usually do not use this semantic information, relying on visual features available during training. Feature extraction methods have evolved significantly over time, with deep learning techniques representing a significant advancement in the field.

Traditionally, local feature extraction relied primarily on highly tailored techniques such as Histogram of Oriented Gradients (HOG) [32], Scale-Invariant Feature Transform (SIFT) [88], and Maximally Stable Extremal Regions (MSER) [106]. These methods aim to capture specific patterns, textures, or areas in images. They require manual design of feature extraction algorithms tailored to specific tasks.

Deep learning, particularly convolutional neural networks (CNN), gained prominence in feature extraction around the mid-2010s [57]. CNNs revolutionised the field of computer vision by automatically learning hierarchical representations of data directly from raw input through convolutional, pooling, and fully connected layers [91]. Recent methods such as SuperPoint [33] and SuperGlue [128] outperform more traditional local features algorithms such as SIFT in terms of repeatability of detected features and accurate descriptor matching.

In general, each layer of a CNN progressively learns to extract more abstract and high-level features from the input data. The lower layers capture low-level features, such as edges and textures, and the higher layers capture more abstract concepts, such as object parts and semantic attributes. This hierarchical learning enables better generalisation of unseen data and improved performance on complex tasks.

Deep learning methods automatically learn feature representations directly from the data, which is particularly advantageous in computer vision and natural language processing, where manually designing feature extraction algorithms is challenging or impractical. Deep learning models are also highly adaptable and can be trained on diverse datasets and tasks with minimal modifications. They can learn complex patterns and relationships from large-scale data, leading to superior performance compared to traditional local features extraction and shallow learning methods.

In this section, we examine the key methods and techniques used in single image geolocation and highlight the differences between them.

2.4.1 Convolutional Neural Networks

HLoc

The hierarchical localisation method (HLoc) proposed by Sarlin et al. [127] is based on HF-Net, a convolutional neural network trained to detect robust local features for camera pose estimation. This method aims to calculate the camera position within a Structure-from-Motion (SfM) model, with local coordinates. Broadly speaking, the process is divided into the following steps:

1. Offline, processing of the image dataset:
   (a) Extraction of local features and global descriptors from the image dataset using HF-Net.
   (b) Index the global image descriptors based on co-visibility.

   (c) Construction of a Structure-from-Motion model of the scene using local features.

2. Online, processing of the query image:

   (a) Extraction of local features and global descriptors from the query image using HF-Net.

   (b) Retrieve k nearest neighbours (kNN) from the index of global descriptors. These k images, called frames, are used as candidate locations on the map.

   (c) The frames are clustered by the co-visibility of 3D structures. These connected frames are called places.

   (d) For each place, the local features extracted from the query image are matched against the 3D points present in the place using SuperGlue [128]. The camera poses are estimated with a geometric consistency check with the perspective-three-point algorithm [79].

   (e) The process ends when a valid camera pose is successfully estimated.

In the end, this method returns the camera pose in local coordinates, i.e. its position within the SfM model. The model itself is not geolocated, and there is no attempt to estimate global coordinates (latitude, longitude and altitude).

The HF-Net neural network is structured around a single encoder, based on MobileNet [126], and three prediction heads: local feature detection scores, dense local descriptors, and a global image-wide descriptor.

Local features and descriptors are decoded using the SuperPoint architecture [33], which decodes local features and descriptors using a fixed non-learned mechanism. This method has the advantage that the execution time does not depend on the number of local features detected.

For the global descriptor, HF-Net uses the NetVLAD [126] layer applied to the last feature map of the encoder. This layer aggregates local descriptors to create a compact representation of the entire image.

This method was evaluated using the Aachen, RobotCar and CMU datasets [131]. Together, the three datasets have over 100,000 day and night urban and suburban images. The latest version of HLoc [129] is able to localise 89.6 % of the daytime images in the Aachen dataset within 25 cm, and 95.4 % within 50 cm. Taking into account only images at night, it localises 86.7 % of the images within 25 cm and 93.9 % within 50 cm.

PlaNet

Weyand et al. [160] published their work on PlaNet, a convolutional neural network trained with 126 million geotagged photos for ten weeks on a supercomputer. The geolocation task is reduced to a classification problem, with the Earth's surface subdivided into a set of 26,263 regions (the classes of the model) and almost 100 million parameters.

In this classification approach, the output classes are the geographical cells. This CNN receives an input image and outputs a probability distribution over the world. Although the output is not expressed in latitude and longitude coordinates, this formulation expresses its uncertainty assigning a confidence level to each cell.

The S2 geometry library is used to create non-overlapping cells that cover the Earth's surface employing a hierarchical partitioning technique. This is achieved by projecting the surfaces of a cube onto it. As the distribution of photos available on training is not uniform across the globe, the subdivision recursively descends each quad-tree and subdivides cells until all cells contain up to 10,000 photos. This partitioning is illustrated in Figure 2.3.

The authors argue that this adaptive tiling makes training classes more balanced, uses the parameter spaces more efficiently by focusing in densely populated areas, and has the potential to reach street-level accuracy in areas where cells are small enough. Despite such massive effort, PlaNet is able to localise just 3.6 % of the dataset at the street level and 10.1 % at the city level. One reason for that is exactly the use of location discretisation, which hurts its accuracy in sparse areas. Although the S2 library allows for a representation of every square centimetre on Earth, it would be necessary to have a couple of images of each cell to obtain an acceptable accuracy at such a level.

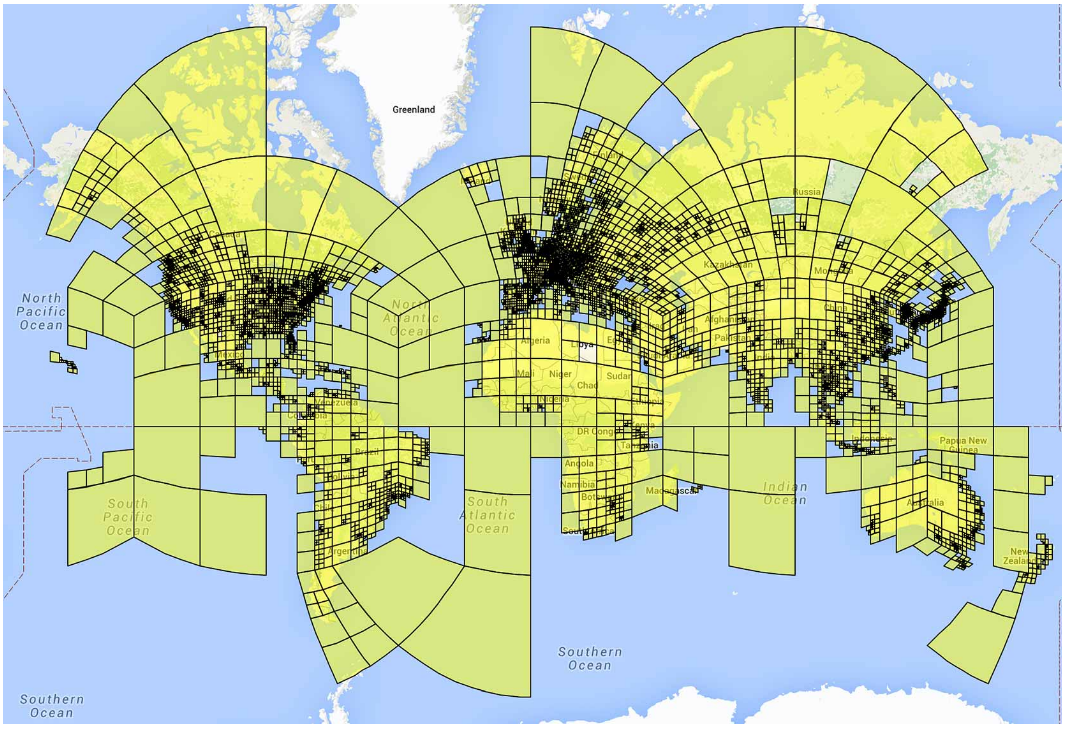


Figure 2.3: Hierarchical partitioning of the Earth's surface into over 26.000 non-overlapping cells. Adapted from Weyand et al. [160]

2.4.2 Multiple Nearest Neighbour

Recently, large-scale image geolocation methods that employ techniques based on image matching and retrieval have attracted a great deal of interest [64, 119, 124, 169]. In such methods, the geolocation of a query image is estimated by finding a match in a dataset of geotagged photos.

Hays and Efros [64] propose the IM2GPS, an algorithm based on scene matching. For this task, six million geotagged images from Flickr [44] are used as a reference dataset. In an effort to exploit the correlation between the images' properties and geographic location, a range of features are extracted from the images:

Tiny images. Images are reduced to 16×16 pixels to reduce computational processing and make the algorithm less sensitive to exact alignment.

Colour histograms. Images are transformed into the CIELAB colour space. The ranges of dimensions $\boldsymbol{L}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are reduced to 4, 14 and 14, respectively. The intensity dimension $\boldsymbol{L}$ has fewer bins because other descriptors already capture information on the intensity distribution of images.

Texton histograms. A universal texton dictionary with 512 entries is built using a bank of filters. Texture features are useful in distinguishing between geographically correlated characteristics such as terrain types, vegetation, building materials and ornamentation styles.

Line features. The use of histograms based on line angles and lengths enables the discrimination between natural and man-made landscapes, as well as the identification of scenes with similar vanishing points.

GIST descriptor. This descriptor is reported to be efficient at scene categorisation and retrieval of scenes semantically and structurally similar.

Geometry context. Image regions are classified into three classes – ground, sky and vertical – using geometric class probabilities.

Given a query image, all features above are extracted and the feature distance is calculated for every image in the dataset. The distances are then normalised in such a way that each feature contributes equally to the ordering of scene matches. Based on the aggregated feature distances, the algorithm then identifies the nearest neighbours [102] in the dataset for the query image and estimates the geolocation based on the geolocation of those neighbours.

The authors argue that the first nearest neighbour is not robust enough to estimate geolocation. They select 120 nearest neighbours, use a mean-shift bandwidth of 500 km and ignore clusters with fewer than four matches, leading to the formation of 6 to 12 clusters that contain approximately two-thirds of the original matches. In the end, the cluster with the highest cardinality is reported as the query image's geolocation.

The mean-shift heuristic localised about 25 % of the test images within the scale of a country (750 km) and 12 % at a city level (25 km). An updated version of the algorithm reaches an accuracy of 2.5 % at street level (1 km) [65].

Despite the huge number of photos in their dataset and the wide range of features extracted, the IM2GPS algorithm still has low accuracy at the street level. We can highlight the inefficient cluster approach to select the most suitable matches among the nearest neighbours.

Compared to the previous approach, IM2GPS has some shortcomings. On PlaNet, clusters are defined beforehand, based on the number of images available for training. In this approach, clusters are formed based on their visual similarity with the query image. Although both approaches are prejudiced on areas with few reference photos, the IM2GPS is even more affected in this case. Its clusters tend to be more sparse and unpredictable, enclosing matches in a 500 km range. Therefore, the error is expected to be in the same order of magnitude.

2.4.3 Generalised Minimum Clique Problem

Zamir and Shah [169] propose an elaborated framework for geolocating an image based on image retrieval using a multiple nearest neighbour local feature matching method and reducing the problem to a Generalised Minimum Clique Problem (GMCP).

In this method, SIFT features are extracted from a query image (query features) and k nearest neighbours (NNs) [102] for each feature are retrieved from the reference image set (reference features). To accelerate the matching process, the query features that do not have distinctive NNs are coarsely removed at this point. The remaining query features and their corresponding k NNs reference features are organised in a graph structure. For each query feature, a single reference feature among the k NNs is selected using a GMCP–based feature matching such that all matches are globally consistent. Finally, the location of the reference image with the greatest number of feature matches is returned as the location of the query image. Algorithm 2.1 gives an overview of this framework.

Explaining the Algorithm 2.1 in more detail, the query image Q is given as an input with the number of nearest neighbours k. In addition, the program

Algorithm 2.1: Single image geolocation using GMCP

```
Input  : Query image $Q$, number of nearest neighbours $k$
Output : Geolocation estimative of $Q$
Data   : Reference image dataset $I$

$k$d-tree $K \leftarrow$ extract local features from images in $I$;
$q^i \leftarrow$ extract $M$ local features from $Q$;
$v^i_m \leftarrow$ find $k+1$ nearest neighbours for each $q^i$;

/* Pruning                                                    */
foreach feature $q^i$ do
    if $v^i_1$ and $v^i_{k+1}$ are more than 80 % similar then remove $q^i$;
end

/* Define graph and clusters                                  */
graph $G \leftarrow (\mathbf{V}, E, \overline{\omega}, w)$, where
    $\mathbf{V} \leftarrow$ nodes corresponding to each $v^i_m$
    $E \leftarrow$ edges between all nodes as long as they do not belong
        to the same cluster,
    $\overline{\omega}(v^i_m) \leftarrow$ node cost as the similarity between the node $v^i_m$ and
        its corresponding query feature $q^i$,
    $w(v^i_m, v^j_n) \leftarrow$ edge weight as the distance between the geolocation
        coordinates of images from where local features $v^i_m$ and $v^j_n$
        were extracted;
$C_i \leftarrow L$ disjoint clusters containing $v^i_m$ nodes;

/* Compute cost of feasible solutions                         */
subgraph $G_S \leftarrow (\mathbf{V}_S, E_S, \overline{\omega}_S, w_S)$ representing each feasible solution in
    which one node $v^i_m$ is selected from each cluster $C_i$;
$C(\mathbf{V}_S) \leftarrow$ cost function of the graph $G_S$ induced by its nodes and edges;
$\hat{\mathbf{V}}_S \leftarrow$ solution with minimal cost, i.e. $\mathrm{argmin}_{\mathbf{V}_S} C(\mathbf{V}_S)$;

return geolocation of reference image with the highest number of
    feature nodes in $\hat{\mathbf{V}}_S$;
```

has access to a reference image dataset I. At the end, the algorithm returns the geolocation estimative of Q.

At first, all images in the reference set I are processed and local features are organised in a kd-tree K. The query image Q is then processed and M local features are detected. The local descriptor of the ith query feature is referred to as q^i , where $i \in \mathbb{Z}^+ : 1 \leq i \leq M$.

Each query feature q^i is then compared to the reference features stored in the kd-tree K, and $k + 1$ NNs are retrieved for each q^i. In this way, the mth NN reference feature of the ith query feature is called v^i_m, where $m \in \mathbb{Z}^+ : 1 \leq m \leq k + 1$.

The query image contains numerous local features that are not relevant to this task, such as those identified on passing objects, trees, or on the ground. Such features are identified and removed based on their similarity to the last NN retrieved, i.e.

$$\begin{cases} \text{remove } q^i, & \text{if } \dfrac{\|q^i - \zeta(v^i_1)\|}{\|q^i - \zeta(v^i_{k+1})\|} > 0.8 \\ \text{retain } q^i, & \text{otherwise,} \end{cases} \tag{2.1}$$

where $\|.\|$ represents the Euclidean distance between vectors and $\zeta(.)$ retrieves the feature descriptor of the argument node. In this framework, the ith query feature is pruned if its first and $(k + 1)$th NNs are more than 80 % similar. In other words, a query feature q^i is considered uninformative and pruned if the first and last NN reference features retrieved from the kd-tree K have a similarity ratio greater than 80 %. After the pruning, only L features are kept for the next step, where $L \in \mathbb{Z}^+ : 1 \leq i \leq M$.

At this point, the problem starts to be reduced to a Generalised Minimum

Clique Problem (GMCP). A graph $G = (\mathbf{V}, E, \overline{\omega}, w)$ is defined, where $\mathbf{V}$, E, $\overline{\omega}$ and w are the set of nodes, edges, node costs and edge weights, respectively. Each element is defined as follows:

- $\mathbf{V} = \{v^i_m\}$, i.e. the nodes in $\mathbf{V}$ correspond to the reference features v^i_m
- $E = \{(v^i_m, v^j_n) | i \neq j\}$, i.e. the edges in E connect every possible pair of node v^i_m as long as they do not belong to the same cluster
- $\overline{\omega}(v^i_m) = \|q^i - \zeta(v^i_m)\|$, with $\overline{\omega} : \mathbf{V} \longrightarrow \mathbb{R}^+$, i.e. the node cost $\overline{\omega}$ is the local feature similarity between the node v^i_m and its corresponding query feature q^i
- $w(v^i_m, v^j_n) = \|\rho(v^i_m) - \rho(v^j_n)\|$, with $w : E \longrightarrow \mathbb{R}^+$, i.e. the edge weight between two nodes is the similarity between the global features of their parent images, which is retrieved by $\rho(.)$

Now the graph G is created, L disjoint clusters C_i are formed with the nodes in $\mathbf{V}$, with each cluster corresponding to a query feature q^i. The subset of nodes in each cluster C_i represents the k corresponding NNs to the query feature q^i. Figure 2.4 shows an example of graph G with five disjoint clusters.

A feasible solution to this Generalised Minimum Clique Problem is represented by a clique with a single node from each and every cluster. In other words, a feasible clique is a subgraph G_S of G in which one node v^i_m is selected from each cluster C_i. This subgraph is defined as $G_S = (\mathbf{V}_S, E_S, \overline{\omega}_S, w_S)$, where $\mathbf{V}_S = \{v^1_a, v^2_b, v^3_c, \dots\}$, i.e. ath node from C_1, bth node from C_2, and so on, edges $E_S = \{E(p, q) | p, q \in \mathbf{V}_S\}$, node costs $\overline{\omega}_S = \{\overline{\omega}(p) | p \in \mathbf{V}_S\}$ and edge weights $w_S = \{w(p, q) | p, q \in \mathbf{V}_S\}$. To keep it simple, the set of nodes $\mathbf{V}_S$ can be referred to as a feasible solution as it is enough to form G_S.

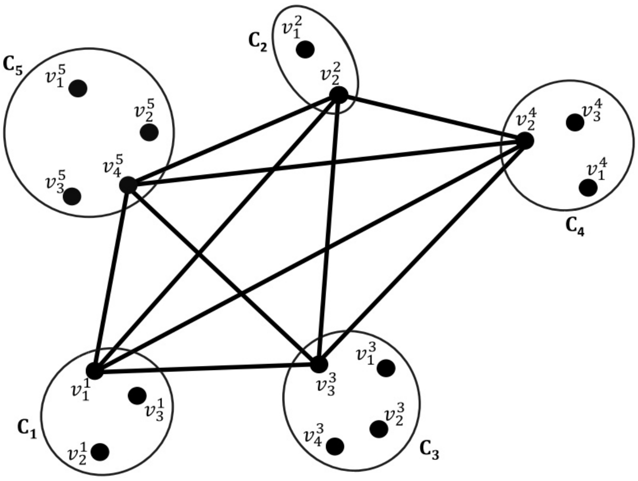


Figure 2.4: An example of Generalised Minimum Clique Problem (GMCP) applied to a graph G with five disjoint clusters C_i. A feasible solution is a clique with one node from each cluster. From [169].

In this way, the cost C of a feasible solution $\mathbf{V}_S$ is defined as

$$C(\mathbf{V}_S) = \frac{1}{2}\sum_{i=1}^{L}\sum_{\substack{j=1,\\ j\neq i}}^{L}\left(\frac{1}{2}\alpha\overbrace{\left(\overline{\omega}(\mathbf{V}_S(i)) + \overline{\omega}(\mathbf{V}_S(j))\right)}^{\text{local features}} + (1-\alpha)\underbrace{w(\mathbf{V}_S(i),\mathbf{V}_S(j))}_{\text{global features}}\right), \tag{2.2}$$

i.e. the sum of all weights of nodes (global features) and edges (local features) of G_S, balanced by a factor $0 \leq \alpha \leq 1$. A larger α increases the contribution of node weights (global features) to the cost of the solution $\mathbf{V}_S$, while a smaller α increases the contribution of edge weights (local features) instead.

Thus, low values for edge and node weights mean a low solution cost and therefore a high global consistency. Finally, the optimal solution with the minimum cost is expressed by $\hat{\mathbf{V}}_S = \text{argmin}_{\mathbf{V}_S} C(\mathbf{V}_S)$, which is the solution for the Generalised Minimum Clique Problem. The geolocation of the reference image with the highest number of feature nodes in $\hat{\mathbf{V}}_S$ is given as the geolocation estimation of the query image.

Distance function

This GMCP-based method employs a function D to measure the distance between global features. The choice of function is important to reduce the impact of outlier nodes from disjoint groups. Thus, Zamir and Shah [169] use the function

$$D(x, y) = \sqrt{2 - 2e^{-\frac{\|x-y\|^2}{2\sigma^2}}}, \tag{2.3}$$

which boosts short distances and caps the effect of outliers to a constant value $\tau = \sqrt{2}$. The distance function D is plotted in Figure 2.5. Using this function makes large distances contribute equally to the cost function, while tight groups of global features reduce the overall cost.

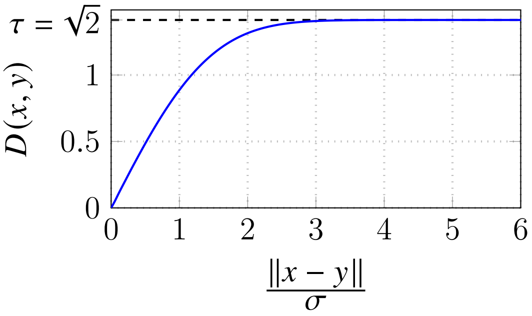


Figure 2.5: Plot of the distance function D defined in (2.3). It dumps large values and boosts short ones.

This GMCP-based geolocation method reaches an accuracy of 40 % in a 50 m range and 57 % in a 200 m range with the robustified distance metric D (2.3). These results are achieved using SIFT features to match the query and the reference images. No other local feature algorithms are considered.

2.5 Conclusion

In this chapter, we presented the state-of-the-art technology for helping outdoor navigation of persons with visual impairment. The functionality of these systems is, however, often limited to obstacle detection and navigation assistance based just on GNSS information presented to the user in limited forms. We also examined the key image processing methods and techniques used for single image geolocation.

Analysing the needs of persons with visual impairment for locomotion and social inclusion, we recognise the need to analyse the requisites of a NAVI system that incorporates new technologies and vision-based solutions to support outdoor navigation.

Chapter 3

Desired features of a vision-based navigation assistant

Part of the content of this chapter features in the following published article:

- R. Busatto and R. Harvey, ‘Outdoor Navigation Assistants for Visually Impaired Persons: Problems and Challenges,’ *Journal on Technology and Persons with Disabilities*, vol. 10, pp. 184–205, 2022.

In this chapter, we present the desired features and user interface of a vision-based NAVI system for outdoor navigation, with a focus on addressing the needs of individuals with visual impairments. We consider the equipment and sensors currently available that could make it possible to build a system with such features, identifying key technology gaps and areas that need further research.

3.1 Methodology

The methodology for this requirement analysis began with active engagement with end users from Action for Blind People, a division of the Royal National Institute of Blind People (RNIB). A semistructured focus group session was conducted on October 16, 2015, at the Action for Blind People's office in Cringleford. Three associated members who are blind participated in the session. They were asked to narrate their experiences navigating outdoors as individuals with visual impairment.

The session lasted two hours, and notes were taken to register the information shared by the participants. The use of a narrative methodology allowed to explore the challenges faced by individuals when navigating outdoors. For example, a participant shared their experience trying to get to a dentist appointment, illustrating the practical difficulties faced in everyday situations. As soon as they leave home, they need to (i) walk to the bus stop; (ii) catch the right bus that takes them as close as possible to their destination; (iii) get on the bus and find a seat; (iv) get off the bus at the right stop; (v) find their way to their destination; (vi) find the right building; (vii) find the entrance; and (viii) find the reception. From there they are more likely to receive human assistance to find their way indoors.

This journey presents some risks, e.g. walking off the pavement and collisions, that must be safely avoided at all times. A step on the road exposes the user to life-threatening risks. Some other tasks, e.g. identifying buildings, shops, buses, entrances, signs and written information, are hardly possible without the assistance of an electronic device or a person with no visual impairment.

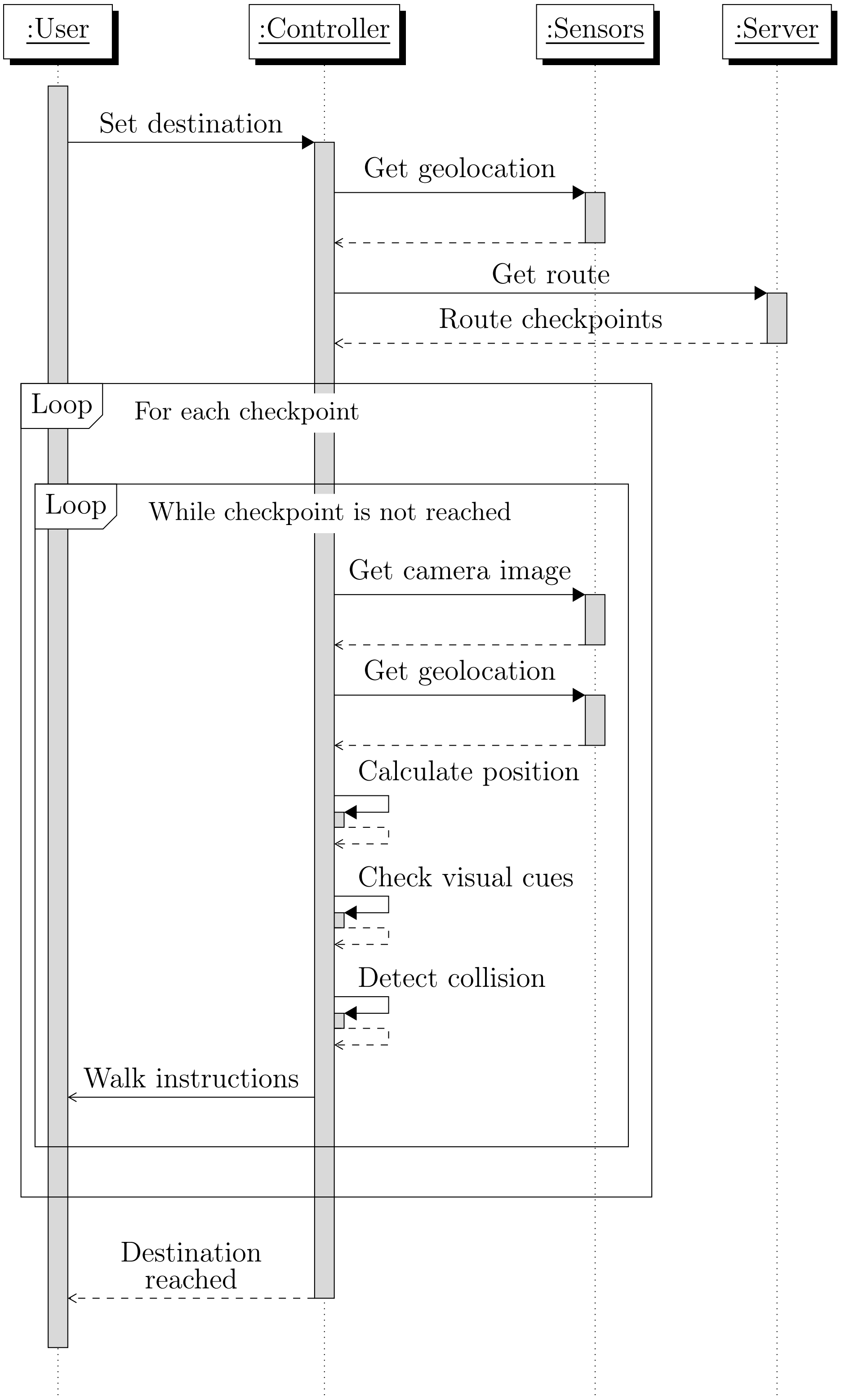
The insights gathered from the focus group session were integrated with the literature review reported in Chapter 2. By analysing the findings from both the focus group session and the literature review, a comprehensive understanding of the requirements for a navigation assistant system emerged, which is presented in this chapter. This integrated approach ensured that the identified features are grounded in the lived experiences of end users, while also being informed by the latest developments in assistive technology.

3.2 Features and user interface

Algorithm 3.1 shows a sequence diagram for the task of vision-based outdoor navigation. First, when the user requests assistance to reach a destination, the system (controller) must retrieve the user's geolocation and request a route to the online server. An internet connection allows making use of more complex algorithms and access to services such as the Google Directions API [52] and the Apple WebKit [12] for up-to-date information about maps, roads, and available routes. However, local processing is preferred when possible.

The evaluation of possible routes must consider: (i) time and length of journey, (ii) accessibility and safety of the route, considering if it is well signposted and paved, (iii) easy access to public transport like bus, tube, train, and tram, and (iv) roadworks and closed ways. The route retrieved from the server is segmented, and the user is expected to reach checkpoints in sequence to arrive at the destination.

The route segments and checkpoints need to be carefully chosen. A segment that involves crossing a street, for example, must be broken down into more specific steps: 'approach the pedestrian crossing,' 'activate pedestrian traffic light,' 'cross the street' and 'reach the sidewalk'. These steps are usually implicit for persons who are sighted, yet they are essential to allow autonomous journeys of persons with visual impairment. It is not just about giving more instructions; the quality of instructions is essentially different. Even when the same route is followed by persons with and without visual impairment, the instructions must consider the individual needs of each user.



Algorithm 3.1: Sequence diagram for the vision-based outdoor navigation task.

For each checkpoint on the route, the system must guide the user by audio and keep estimating the user position in real-time, evaluating whether they are on track or the route needs to be recalculated. Ideally, the camera should identify obstacles in real time and assess imminent collision risk. In a scenario of crossing a street, for example, the camera allows the identification of pedestrian crossings, traffic lights, cyclists and cars to decide when it is safe to cross. In addition, the system must also identify and prioritise recognised text to announce relevant information only according to the context. A big sign far away may be less relevant than a small street sign near the user. With no priority classification, the user may be overwhelmed by irrelevant announcements.

The walk instructions are ideally given by both spoken and audible signals. The audio feedback must not block signals from external sources, which helps in keeping the safety of users. With the use of smart glasses and binaural audio, users may receive audio instructions to aim their heads towards locations where they expect important visual targets [26]. In this way, there is no need to train users beforehand. A heads-up display may be used by persons with low vision, enabling announcements both by audio and on the display. Yet, the use of a display is secondary and is not in any way essential.

The use of non-visual references is fundamental when giving navigation instructions. A simple instruction such as ‘walk 20 m’ is hard to follow because it refers to a measure not easily verifiable in such a situation. Alternatively, saying ‘walk twenty steps’ is more intuitive and easily verifiable. Although it is not as accurate, checking the user’s position in real-time allows route correction and follow-up instructions such as ‘walk two more steps’ or ‘you have reached the street corner, turn left now’.

The geolocation accuracy requirement should ideally be under one meter to ensure the system can accurately guide users with visual impairments through complex environments. Precise location is critical for safe navigation in dense urban environments where small errors could lead to potentially hazardous situations. Additionally, with accurate geolocation, the system can provide precise instructions such as indicating the exact point to cross a street or reach a specific checkpoint. This level of accuracy enhances the user's confidence in following the navigation instructions and reduces the chance of errors.

When the destination is finally reached, the system may learn the user's preferences considering the journey actually undertaken. The more people use it, the better it would get at suggesting convenient routes to everyone.

3.3 Technology gaps

Table 3.1 summarises the desired features of an outdoor NAVI system as described and highlights their current development status reported in the literature.

The first task in Table 3.1 is to localise the user with high accuracy (F1). Although there are some subtleties in certain scenarios, it is a standard task, so we have marked it as such. Although GNSS provides only around 10 m accuracy, which is either sufficient for many purposes (e.g. maritime navigation) or can be combined with tracking models to give improvements. In autonomous vehicle navigation, for example, systems assume that the vehicle is located on the road using an up-to-date map. Thus, cross-track errors can be zeroed. Pedestrian navigation is more challenging because people roam off streets. With good tracking, newer GNSS components such as Galileo, SBAS systems such

Table 3.1: Features of the NAVI model for outdoors navigation.

	Feature	Development	Solved?
F1	Localise user with high accuracy	Current accuracy is approximately 10 m with GNSS; Maximum required error is 0.5 m	Partially
F2	Calculate the best route to reach a point of interest	Pedestrian routing is freely available on smartphones map apps [11, 52]	Yes
F3	Define micronavigation instructions	Further studies needed on Human Computer Interface (HCI) [23]	No
F4	Recognise public transport vehicles	Solved by Computer Vision detection and classification algorithms	Yes
F5	Locate doors and entrances	Solved by Computer Vision algorithms with 97.96 % accuracy [113]	Yes
F6	Recognise relevant signs and labels	Partially solved by Computer Vision detection, classification and OCR algorithms [110]	Partially
F7	Identify the pavement	Solved by Computer Vision segmentation algorithms [109]	Yes
F8	Access collision risk	Solved by Computer Vision algorithms	Yes
F9	Perform visual navigation	Further studies needed in Computer Vision	No
F10	Interact with the user in a natural and intuitive way	Further studies needed on Human Computer Interface (HCI) [23]	Partially

as WAAS and EGNOS, combined with vision-based solutions, it is reasonable to assume that determining outdoor pedestrian geolocation might be solved within 1 m. Indoor navigation will either require significantly greater antenna gain at the receiver using larger and more complex receptors, or wide-scale deployment of indoor GNSS augmentation systems. Both seem unlikely within the next ten years. Hence are other ‘partially’ ratings.

Calculating the best route (F2) can be considered solved for outdoor pedestrian navigation. Google and Apple services provide pedestrian routing freely available online and with vast documentation [12, 53]. These services consider factors such as time, walking distance, accessibility, and roadworks.

Defining the micronavigation instructions (F3), on the other hand, remains a challenging open problem. The route segments retrieved from online routing services must be broken down into more specific segments. Navigating in large open spaces, in a park, for example, is hardly a problem for persons who are sighted. Persons with visual impairment, on the other hand, may get lost or go astray without accurate navigation instructions and constant rerouting assessment. Furthermore, there is every reason to think that instructions need to be personalised since visual disabilities are diverse.

Recent advances in computer vision make viable the recognition of vehicles (F4), doors (F5), signs (F6) and sidewalks (F7) with high accuracy. Recognition of labels using Orcam MyEye and Seeing AI, for example, is reported to achieve greater than 95% accuracy in text recognition for flat, plain word documents [54]. Recent research has been conducted on the detection of signs specifically for the navigation of persons with visual impairment [29]. When recognition is associated with tracking objects, it becomes possible to access collision risks in real-time (F8) without the need for extra sensors.

Despite notable improvements in visual recognition, performing visual navigation (F9) remains a very significant problem for NAVI systems. Current solutions involve building 3D maps a priori [36], which is not desirable in outdoor navigation. Even if the user position could be calculated with enough accuracy, the information retrieved by the camera and processed by computer vision algorithms still needs to be classified and organised into instructions to the user. This is the basis for naturally interacting with users (F10).

It is important to recognise that this analysis is written from the traditional research perspective, which focuses on the lower Technology Readiness Levels (TRLs 0 to 4). It goes without saying that to be useful, all research needs to be pushed to higher TRL levels, which requires commercial or government investment.

The main focus of this thesis is to investigate vision-based geolocation methods to localise the user with high accuracy (F1). In Algorithm 3.1, this corresponds to the steps 'get camera image', 'get geolocation', and 'calculate position.' This is an essential requirement for navigating safely through urban environments. Vision-based approaches have the potential to achieve sub-meter accuracy, as they can use visual cues present in the environment to determine the user's location. Unlike traditional methods relying on GNSS or inertial sensors only, vision-based techniques can reduce localisation errors caused by signal occlusion, multipath effects, or urban canyon areas (as discussed in the next chapters). Vision-based geolocation methods show potential to advance the state-of-the-art in outdoor navigation systems, potentially improving the safety, efficiency, and user experience in real-world settings.

3.4 Conclusion

In this chapter, we explored the requisites of building vision-based outdoor navigation assistants for persons with visual impairment. We presented the requisites of building a NAVI system integrating all journey stages and exploiting the current technology to its full potential. Finally, we highlighted the areas that need further research and the problems that need to be solved to make such a system possible. An accurate estimate of the user position, for example, is essential for a safe NAVI system. In the next chapters, we explore vision-based geolocation methods as an attempt to geolocate the user outdoors.

Chapter 4

GNSS geolocation refinement using geotagged street-view images

Vision-based geolocation methods can be classified into two main categories based on whether they construct an internal 3D structure or not. Methods that fall into the first category construct an internal 3D model of the environment, which can be used to estimate the position and orientation of the camera. Algorithms in this category employ techniques such as Structure from Motion (SfM) [132], which uses a series of 2D images of a scene taken from different viewpoints to construct a 3D model of the environment. The SfM works by identifying common features in multiple images and using them to estimate the relative position and orientation of the cameras. Once the camera positions are estimated, the 3D structure of the scene can be reconstructed by triangulating the position of the features observed in the images. It is usually computationally expensive to both build an SfM model and to match features from a query image against large SfM models [119].

The second category of vision-based geolocation methods does not involve constructing an internal 3D model of the environment. Instead, these methods are usually based on image retrieval, which involves matching the query image captured by the camera with a dataset of images with known geolocation information. The feature matching is based on the similarity of features extracted from the query image and the dataset images. Although computationally faster, these methods are usually less accurate, as they do not exploit the epipolar geometric relation between local features and their projections onto 2D images [119].

The choice of local feature detectors and descriptors is crucial for algorithms in either category. The Scale-Invariant Feature Transform (SIFT) has traditionally been used in various applications due to its partial invariance to changes in illuminance, occlusion, and point-of-view. However, SIFT features may fail to match when the changes are significant or multiple aspects change simultaneously. Recently, with the advance of neural network algorithms, new local feature detectors and descriptors have been proposed [33]. These 'deep features' learn changes caused by challenging factors, such as illuminance and changes in point-of-view, resulting in the ability to match features that look dramatically different.

In this chapter, we present a method for refining GNSS geolocation through the use of geotagged street-view images. This method builds upon the hierarchical localisation (HLoc) framework proposed by Sarlin et al. [127] with several key enhancements. Unlike the original HLoc method that aims to estimate the camera pose in local coordinates within a large SfM model (see Section 2.4.1 for more details), the primary objective of our approach (HLoc+SV) is to accurately estimate the world coordinates of a user's camera by integrating geotagged street-view images into the SfM model.

There are several novel aspects of our approach presented in this chapter. Firstly, the original HLoc method does not attempt to use any geolocation information. On the other hand, we progressively add geotagged street view images to the SfM model and use them as ‘anchors’ to geolocate the whole model. Secondly, instead of using global visual features to search for pictures taken nearby, we use the user’s GNSS geolocation to dynamically download street view images near the user. In this way, we can build the SfM model on-the-fly rather than having to process a pre-existing dataset of images and build the SfM model offline. Additionally, to overcome the challenge of evaluating results without a high-accuracy GNSS receiver (e.g. RTK), we estimate ground truth by identifying clear marks on the ground visible on satellite images and manually retrieving their geocoordinates. Lastly, another significant contribution is the creation of a dataset with 58 user images and 80 street view images, covering seven scenes in Norwich and London. Each user image includes GNSS coordinates retrieved by the smartphone and also their ground truth for evaluation purposes. This dataset enables accurate testing and validation of visual geolocation methods in urban scenarios.

4.1 Method

The method proposed in this chapter (HLoc+SV) aims to estimate accurate world coordinates of the user camera. We use the GNSS receiver information to retrieve a set of geotagged street-view images, which are used to geolocate the SfM model and estimate the user geocoordinates. Figure 4.1 shows a diagram of our method, which is based on the hierarchical localisation framework proposed by Sarlin et al. [127] with significant modifications and extensions to integrate world coordinates and geolocation information.

In the original framework [127], a Structure from Motion (SfM) model [132]

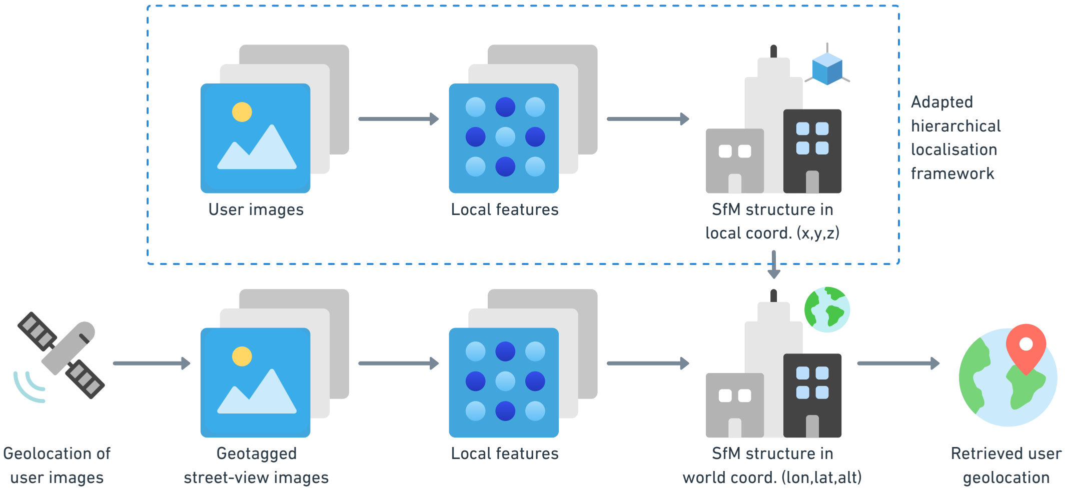

User images
Local features
SfM structure in local coord. (x,y,z)
Adapted hierarchical localisation framework
Geolocation of user images
Geotagged street-view images
Local features
SfM structure in world coord. (lon,lat,alt)
Retrieved user geolocation

Figure 4.1: Diagram of the method HLoc+SV to estimate the geolocation of the user camera.

is built a priori from a set of images using the HF-Net neural network [127] and SuperPoint [33] deep features. A query image is then matched against dataset images using global descriptors, and k nearest neighbours are retrieved. These k retrieved images are clustered based on covisible features in SfM structures. For each retrieved SfM structure, the local features of the query image are matched against the features in that particular SfM structure. The algorithm then estimates the six-degree-of-freedom camera pose using RANSAC [43] to solve the perspective-n-point [79] problem and verify the geometric consistency. The coordinates of their SfM model are local. No effort is made to locate the camera in world coordinates.

In our method, called HLoc+SV, a set of images taken by the user (using a smartphone in the experiments reported below) are converted to greyscale and reduced to a maximum resolution of 1024×1024 pixels. The images are then fed into the HF-Net, and SuperPoint deep features are extracted, with an upper limit of 4096 features per image. For the feature matching step, we use the SuperGlue [128] neural network, pretrained on the Aarchen dataset [131] as provided by the authors [127]. Since we use few user images, all pairs of images are matched exhaustively – i.e. for n images, we consider all $n(n-1)/2$ pairs.

A geometric verification of the matches is then performed to eliminate those image pairs that do not share an overlapping view of the scene, which is done using the COLMAP pipeline [132, 133]. Considering that the camera is not calibrated and its intrinsic parameters will be estimated at the next step, the geometric verification is performed by estimating the fundamental matrix $\boldsymbol{F}$ (instead of estimating the essential matrix, which is suitable for calibrated cameras) [62]. A robust estimation of $\boldsymbol{F}$ is performed using RANSAC [43] to eliminate the several outliers that usually contaminate the feature matches.

The geometrically verified pairs of images are then progressively added to an SfM model, i.e. the 3D reconstruction of the scene. The model is initialised with an image pair from a dense location (i.e. an overlapping area registered by many cameras), and the following images are individually added by triangulating its features' matches with the existing features in the SfM model. This triangulation is the base for solving the perspective-n-point (PnP) problem, which estimates the camera pose and its intrinsic parameters.

We use the COLMAP pipeline to solve the PnP problem using a simple radial camera model, which estimates two intrinsic parameters and covers pincushion and barrel distortions (an introduction to the camera matrix model can be found in [63]). As all user images were taken from the same camera, therefore all images are set to have the same camera parameters. We empirically found this to be the simplest camera model that correctly models the lens distortion in this scenario, accelerating the convergence to find a consistent solution. The SfM model construction finishes when all images are added to the model.

Differently from the original framework, this SfM model does not need to be built beforehand. The 3D model is not created using images stored in a dataset, but from a sequence of images taken by the user a few metres apart from each other. For this reason, the original offline processing step is not needed.

Once the SfM model is created using user images, we add the reference street-view images to the model. Using the user geolocation reported by the user's GNSS receiver, which is embedded in each one of the user photos, we retrieve the street-view images closest to the user from the Google Street View API. Note that, different from similar methods, we do not retrieve images from a dataset using nearest neighbours (NN) algorithms. Instead, we use the GNSS information embedded in the user images to have a rough idea of the user's location and retrieve nearby street-view images. For these experiments, we manually selected the street-view images closest to the user and removed the images that did not have a common overlapping area to any of the user images (pointing to the opposite direction or taken on a perpendicular street, for example). Although the COLMAP pipeline is able to handle images that are not related to the scene, we chose to remove unrelated street-view images to accelerate the process.

The street-view images retrieved by the Google API are added later, in a second step, because each one of them has its own internal camera parameters. Google Street View images are the result of merging images from several cameras attached to the top of a car, merged into a 360° spherical view [10]. The distortion caused by the camera lens is further exaggerated by this post-processing merging. Using their API, we break down each spherical view into eight side sections (at every 45°) with a field-of-view of 60° and a pitch of +10°.

The street-view image sections are progressively added to the SfM model, and the intrinsic parameters of the camera are estimated using a simplified radial camera model. Although the distortions are known to be more severe, we empirically found that the use of more complex models fails to converge to a solution.

4.1.1 Estimation of world coordinates

At this point, all user images and street-view images are added to the SfM model, which contains a 3D map of the scene and the estimation of the camera position of each image in local coordinates.

For the HLoc+SV method, the geocoordinates of the street-view images are used to convert the SfM model from local coordinates (x, y, z) to geodetic world coordinates (WGS-84) expressed in terms of longitude φ, latitude λ and altitude h, i.e. (φ, λ, h). To achieve this, the street-view images' geocoordinates are first converted to an intermediate Earth-centred-Earth-fixed Cartesian coordinate system (ECEF), expressed in (X, Y, Z) coordinates. Both WGS-84 and ECEF coordinate systems are geocentric, i.e. have identical origins at the centre of the Earth. WGS-84 coordinates (φ, λ, h) are converted to ECEF coordinates (X, Y, Z) using the equations [66]:

$$X = (N + h) \cos\varphi \cos\lambda \tag{4.1}$$

$$Y = (N + h) \cos\varphi \sin\lambda \tag{4.2}$$

$$Z = \left(\frac{b^2}{a^2} N + h\right) \sin\varphi \tag{4.3}$$

where a and b are the ellipsoid semi-axes, and N is the radius of curvature in prime vertical, defined as

$$N = \frac{a^2}{\sqrt{a^2 \cos\varphi^2 + b^2 \sin\varphi^2}}. \tag{4.4}$$

We use COLMAP to estimate a 3D similarity transformation from local coordinates (x, y, z) to ECEF coordinates (X, Y, Z). The street-view ECEF geolocations are used as the only reference coordinates (no coordinates from user images are used). This transformation is estimated using RANSAC to be robust to possible outliers, with a maximum error threshold of 8.0 and minimum inlier ratio of 10 %. Although no information about the Google Street View geolocation error or equipment is provided, we assume that it is more precise and reliable than the geolocation retrieved by the user's smartphone.

After applying the similarity transformation, the SfM model is finally geolocated in ECEF coordinates (X, Y, Z). As a result, the camera poses of user images within the SfM model are also expressed in the same coordinates system. As a final step, we perform an inverse geodetic computation [66] to express the user's camera poses in WGS-84 coordinates (φ, λ, h). These coordinates are then returned as a set of user geolocations, one for each user image.

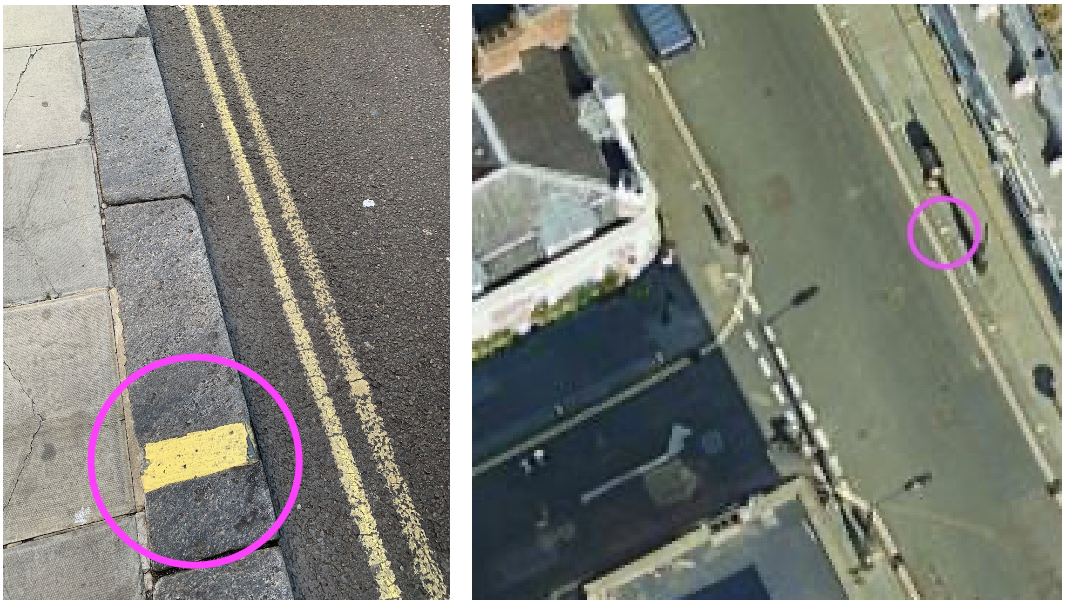
For comparison, we also estimate the user geolocation using the user's GNSS receptor information instead of using the street-view coordinates (reported below as HLoc+GNSS). This scenario simulates when Google Street View images are not available. This similarity transformation is problematic given that the GNSS information retrieved by the user device is often subject to interference in urban centres [156].

For the simpler HLoc+GNSS method, the geolocation embedded in the user images is first converted from WGS-84 coordinates (φ, λ, h) to ECEF coordinates (X, Y, Z) using (4.1) to (4.4). In this method, the user images' ECEF geolocations are used as the only reference coordinates (instead of the street-view coordinates). The 3D similarity transformation from local (x, y, z) to ECEF (X, Y, Z) coordinates is robustly estimated using COLMAP with the same parameters as before. Finally, the camera poses of user images are converted to WGS-84 coordinates (φ, λ, h), which are returned as the result of the HLoc+GNSS method.

4.1.2 Ground truth

There was no high-accuracy GNSS receiver available for this research when these experiments were conducted. Equipment used for land surveys, such as RTK GNSS, is commonly used in research to achieve millimetre-level positioning accuracy. With no such equipment available, the evaluation of results became very challenging.

To overcome this limitation, we estimated the ground truth by carefully choosing locations with clear marks on the floor that are visible on satellite images. We chose locations with clear markings on the ground, such as painted lines and manholes, which are visible on Google Maps' high-resolution satellite images. The user photos were taken with the camera positioned on top of these marks. In this way, the ground truth was retrieved by manually checking the geolocation of such marks on Google Maps satellite images. Figure 4.2 shows some examples of street marks.



(a) Ground mark **(b)** Ground mark seen from satellite

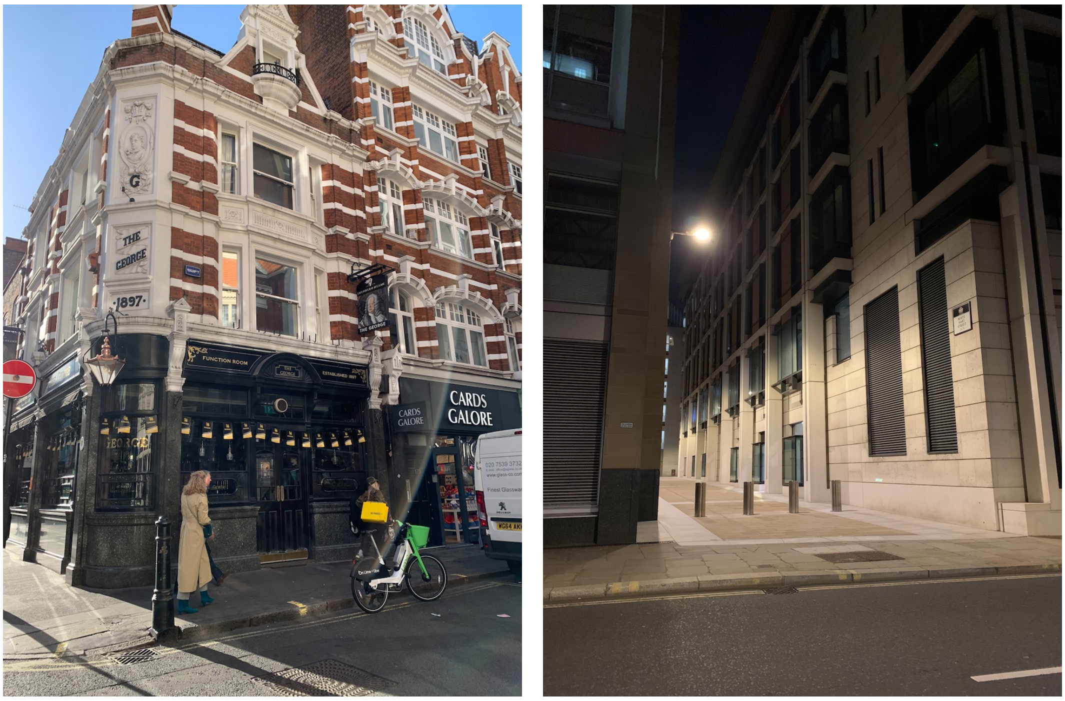
Figure 4.2: Example of a ground mark used to estimate the user's geolocation ground truth. (a) Ground mark from where the user photo was taken. (b) Satellite image of the scene. The ground mark is visible on the pavement.

4.2 Setup

Locations We created a dataset of user images with day and night time photos taken at seven outdoor locations: one in Norwich and six in central London, UK. The number of pictures taken at each location varies from four to twenty-three. Some examples can be seen in Figure 4.3.

Characteristics of each location All pictures in London were taken on the same day over the afternoon and evening. The pictures taken in front of the Norwich Cathedral were all taken on the same afternoon.

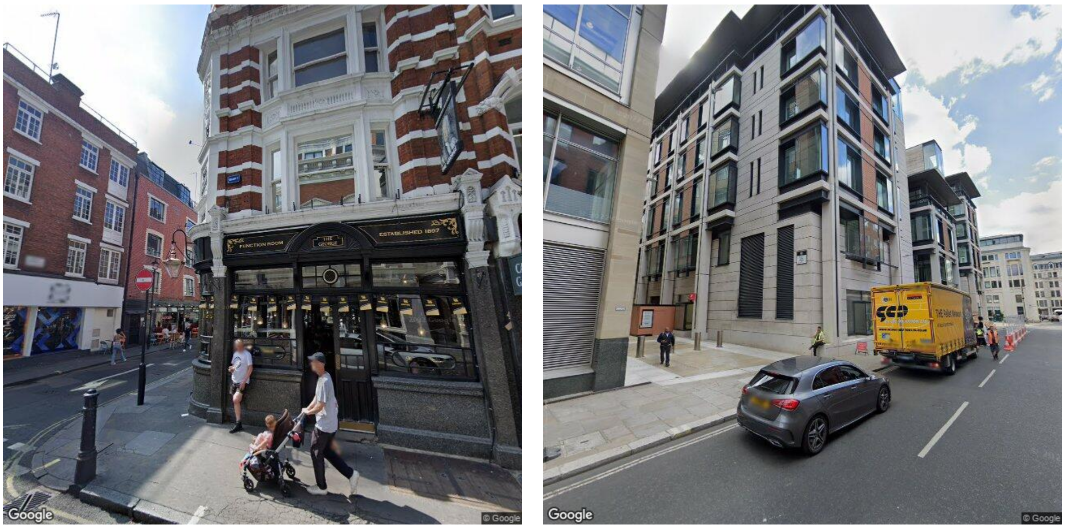
Data collection We collected data in July 2022 (scene S1) and October 2022 (scenes S2 to S7). All images were reduced to a maximum resolution of 1024 × 1024 pixels. For geolocation information, we used the GNSS



(a) User image, scene S5

(b) User image, scene S6

Figure 4.3: Examples of user images taken using a smartphone.



(a) Street-view image, scene S5

(b) Street-view image, scene S6

Figure 4.4: Examples of street-view images retrieved from the Google API closest to the user's GNSS geolocation.

data embedded in the pictures. The native smartphone camera app was allowed to stay open on screen for a minute before each picture was taken so the calculation of the GNSS location would have time to converge. The number of images varies between scenes to assess the robustness of methods under different conditions, specifically with a limited number of user images. As all scenes had similar performance, even with as few as four user images, we also conducted a systematic analysis of the HLoc+SV performance with varying number of images (Section 4.4).

Sensors Only the iPhone camera and embedded GNSS receiver were used. No post-processing or data refinement is done afterwards.

Street-view reference dataset For each location, street-view images from the Google Street View API were downloaded. They have a resolution of 640×640 pixels and geolocation information embedded. Google Street View images are captured with a set of cameras, lasers and GNSS receivers attached to the roof of a car [10]. No information about the specific equipment used is provided. The Google Street View pictures were captured between March 2019 and July 2022. Some examples can be seen in Figure 4.4.

Equipment All experiments reported in this chapter run on an Apple M1 Pro 10-core processor and 16-core GPU. An iPhone XR was used to take user pictures and retrieve the user's GNSS geolocation.

Libraries and source code The hierarchical localisation framework [127] was used as a base pipeline to extract SuperPoint deep features and match using the SuperGlue algorithm with the HF-Net neural network pretrained on the Aarchen dataset. The COLMAP framework [132, 133] was used to create the SfM model and estimate the camera poses. The coordinate transformations were performed using the PROJ software [39]. The code to run the experiments was written in Python 3.9.

Hyper-parameters For the SuperPoint features, we detected a maximum of 4096 keypoints, NMS within a radius of 3 pixels, converted images to greyscale and resized to a maximum of 1024 × 1024 pixels. For the SuperGlue feature matching, we perform 50 Sinkhorn iterations, with outdoor weights. For the COLMAP, we use a radial camera model. For the PROJ, we use the default parameters.

4.3 Results

The results consider both methods – aligning the SfM model using only the GNSS receiver information (HLoc+GNSS) and using the street-view photos geolocation (HLoc+SV). Table 4.1 shows the estimated geoposition mean absolute error (MAE) of the GNSS receiver and both of our methods (HLoc+GNSS and HLoc+SV).

The first row of Table 4.1 shows that, in scene S1, the user pictures were taken during the day and 23 pictures were used to build the SfM model. To geolocate the SfM model, 14 images were retrieved from the Google Street View API. The GNSS receiver embedded in a smartphone had a mean absolute error of 6.65 m (Standard Deviation 3.43). Our method HLoc+GNSS, with no use of street-view images, had a mean error of 3.60 m (SD 2.24). Finally, our method HLoc+ SV, which adds street view images to the SfM model, had a mean absolute error of 0.46 m (SD 0.28).

Table 4.1: Geoposition mean absolute error (MAE) in meters of a smartphone GNSS receiver, HLoc+GNSS and HLoc+SV methods.

		# of images		**Mean absolute error (m)**		
Scene	**Time**	**User**	**SV**	**GNSS**	**HLoc+GNSS**	**HLoc+SV**
S1	Day	23	14	6.65 (SD 3.43)	3.60 (SD 2.24)	0.46 (SD 0.28)
S2	Day	5	9	16.92 (SD 1.87)	16.90 (SD 0.59)	0.54 (SD 0.39)
S3	Day	4	10	5.87 (SD 0.96)	5.84 (SD 0.81)	0.96 (SD 0.44)
S4	Day	4	11	10.97 (SD 1.49)	10.95 (SD 0.75)	0.99 (SD 0.33)
S5	Day	4	6	3.20 (SD 1.19)	3.14 (SD 0.18)	0.57 (SD 0.28)
S6	Night	10	18	14.21 (SD 1.98)	14.15 (SD 1.45)	1.19 (SD 0.12)
S7	Night	8	12	30.21 (SD 1.65)	30.20 (SD 1.30)	1.17 (SD 0.07)
Total		58	80	12.09 (SD 8.67)	10.87 (SD 9.35)	0.77 (SD 0.41)

Taking into account the overall performance of all scenes, with all 58 user images and 80 street-view images, the GNSS receiver achieved a mean absolute error of 12.09 m (SD 8.67). The HLoc+GNSS had an overall mean absolute error of 10.87 m (SD 9.35), and the HLoc+SV, 0.77 m (SD 0.41). Extremely challenging scenarios for GNSS presented the greatest improvement in performance. Scene S7, for example, has nighttime pictures and over 30 m GNSS positioning error. Using HLoc+SV, the positioning error is reduced to 1.17 m.

We run this experiment considering seven scenes: five with pictures taken during the day and two at night. The scenes with night pictures had a greater geoposition error when considering the HLoc+SV method due to the image matching be against street-view pictures taken during the day. Even so, the HLoc+SV has shown a lower mean absolute error in all scenes tested.

Figure 4.5 shows the cumulative distribution of absolute geoposition errors of the GNSS receptor, HLoc+GNSS and HLoc+SV. At a distance threshold of

3 m, for example, 21 % of all images had their camera poses correctly geolocated with HLoc+GNSS and 9 % with GNSS. On the other hand, 100 % of images were geolocated within 1.49 m with HLoc+SV.

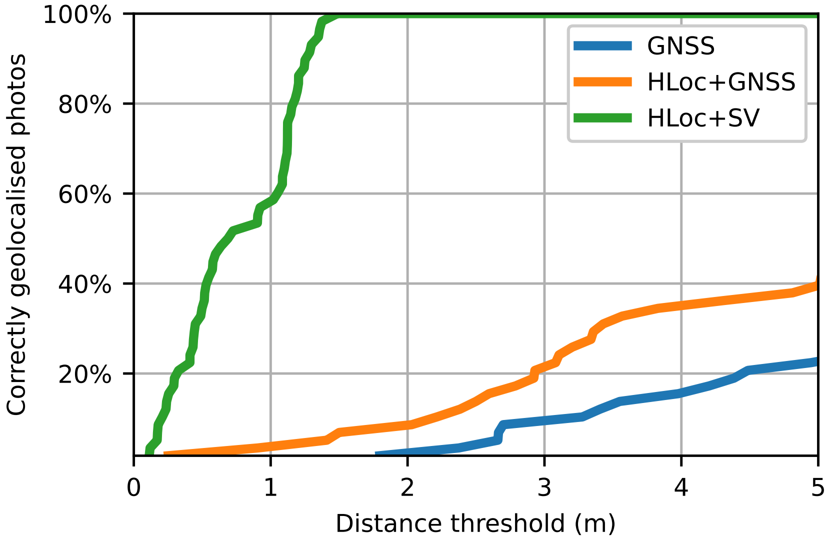


Figure 4.5: Cumulative distribution of absolute geoposition errors. Our method (HLoc+SV) localised all photos within 1.49 m.

Figure 4.6 shows a box plot of absolute geoposition errors. The GNSS error range is between 1.80 m and 33.11 m, with quartiles at 5.35 m (Q1), 9.30 m (Q2) and 15.48 m (Q3). For the HLoc+GNSS, the error range is between 0.25 m and 31.92 m, with quartiles at 3.24 m (Q1), 7.68 m (Q2) and 15.83 m (Q3). Finally, the HLoc+SV error range is between 0.11 m and 1.49 m, with quartiles at 0.43 m (Q1), 0.71 m (Q2) and 1.12 m (Q3).

Figure 4.7 shows map projections of the GNSS and HLoc+SV geopositions in scenes S1 and S2, respectively. The red dots represent the ground truth manually retrieved. The blue triangles represent the geolocation of street-view images retrieved from the Google API, used as a reference to geolocate the SfM model. The GNSS position drift is represented by black lines, showing the

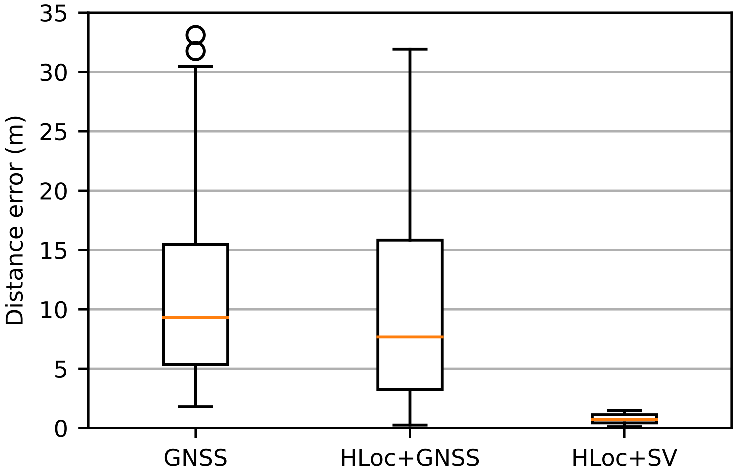


Figure 4.6: Box plot of absolute geoposition errors.

distance between the ground truth geolocation (red dot) and the geolocation obtained by the GNSS receiver (other end of the line). The line in magenta represents the geolocation drift of our method HLoc+SV. The scale of 2 m is represented at the bottom of each map. Note the difference in length between the black and magenta lines. In all cases, the absolute HLoc+SV position error (line length) is lower than the absolute GNSS receiver error.

The photos in scene S1 (top map in Figure 4.7) were taken in Norwich, in front of a cathedral with a clear view of the sky (and consequently a good reception of GNSS signal). All other photos were taken in central London, a highly urban area with tall buildings and urban canyons, more subject to interference and reflection of the GNSS satellites' electromagnetic waves. It is interesting to note that in scene S1 the geolocations reported by the GNSS receiver do not drift in the same direction, but they do in scene S2 (bottom map in Figure 4.7) and in all other scenes.

The most time-consuming step of HLoc+SV is to create the SfM model. The feature extraction runs in real-time at 20 fps. The SfM model building

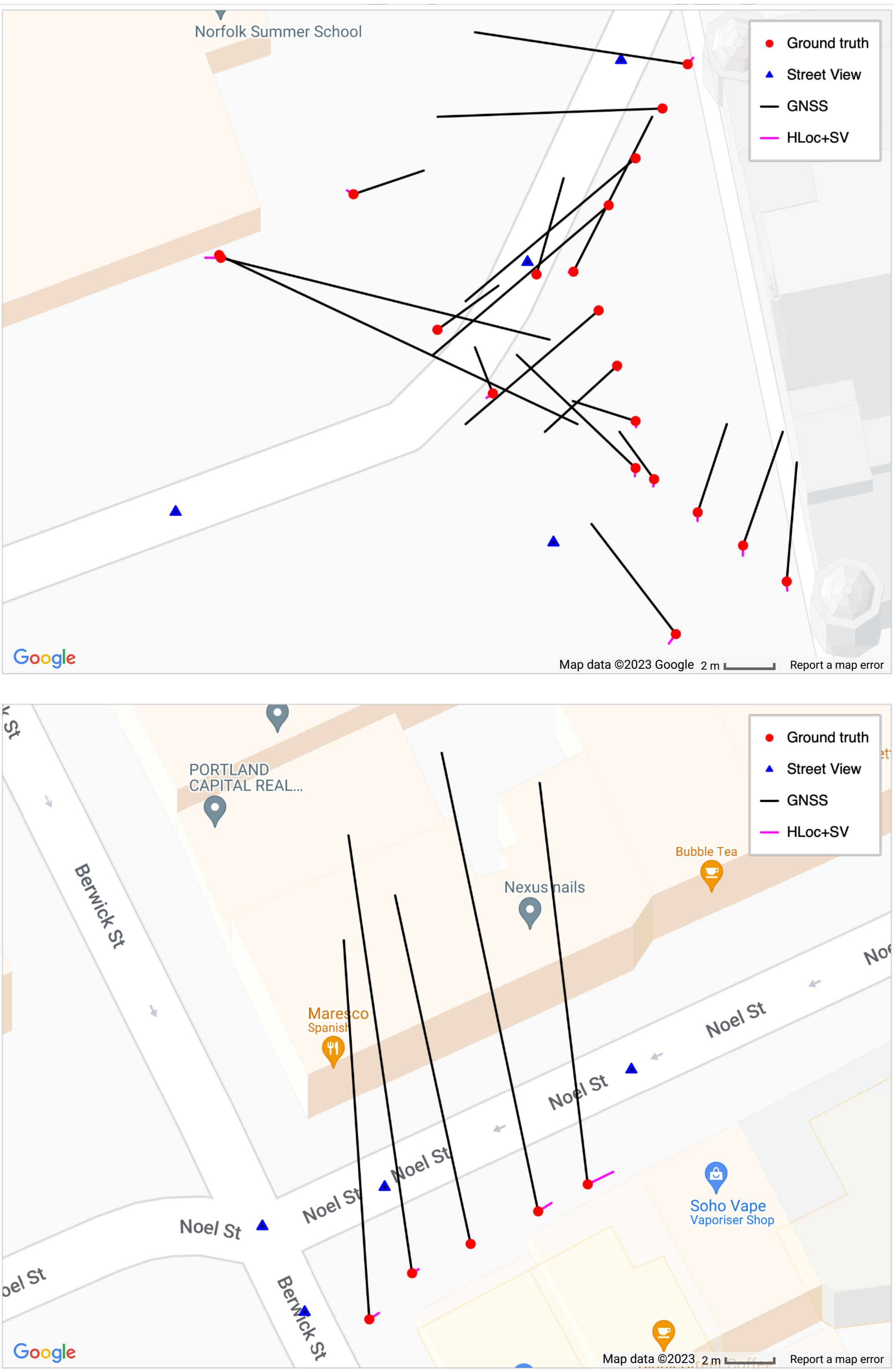


Figure 4.7: Projection on map of geopositions in scenes S1 and S2, respectively. Ground truth position in red. The black line shows the drift between the GNSS position and the ground truth. The Magenta line shows the geoposition drift of our method.

took about 2 min for each scene on an Apple M1 Pro 10-core processor and 16-core GPU.

4.4 Impact of the number of images on performance

The results presented so far showed that HLoc+SV has significantly lower errors even on scenes with as few as four user images and six street view images (scene S5) when compared to using the GNSS information alone (HLoc+GNSS).

For this reason, we conducted a further analysis to investigate how the number of pictures, both user images and street view images, can affect the geolocation performance. Thus, we can discover the minimum number of street view images and user images necessary, and how increasing the number of pictures positively affects performance.

To achieve this objective, we used the following methodology:

1. **Vary the number of images:** The number of user images and street view images was progressively increased from one to the maximum available.

2. **Generate permutations:** For each combination of user images and street view images, up to ten unique permutations were selected.

3. **Build and geolocate SfM model**: The HLoc+SV method run for each permutation of images.

4. **Evaluate model performance:** The mean absolute error (MAE) was computed for each SfM model. We set an error limit of 10 m for cases

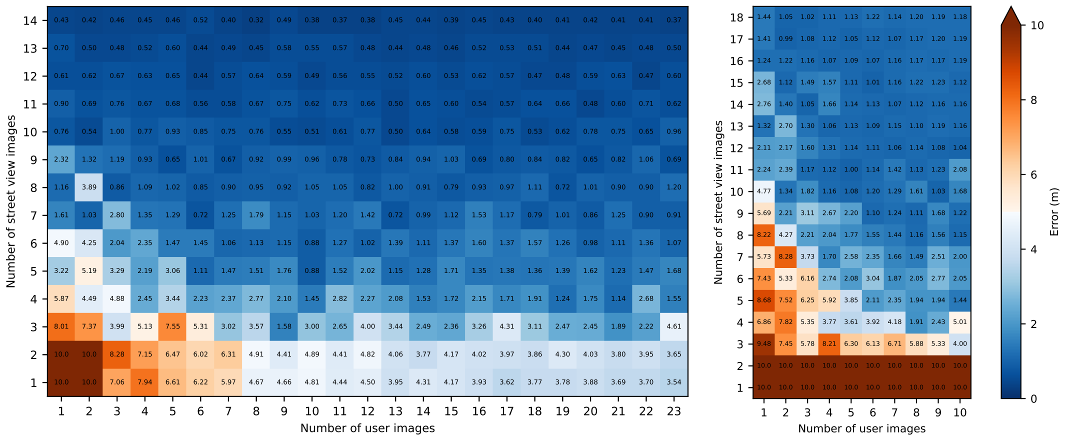
where the geolocation error was higher than 10 m or a model could not be built.

5. **Calculate mean performance:** Finally, we determined the mean MAE of all ten permutations.

For example, to calculate the geolocation error for five user images and seven street view images, we randomly select up to ten unique permutations of user and street view images, run the HLoc+SV method for each permutation, calculate the MAE, and then the mean of all ten MAE.

The experiment results for the two scenes with the highest number of images, S1 and S6, are presented in Figure 4.8, where the x-axis represents the number of user images and the y-axis represents the number of street view images. Each cell in the matrix represents the mean MAE of up to ten image permutations. A gradient colour error bar is used, with white representing geolocation errors within 5 m, blue representing errors from 0 m to 5 m, and red representing errors from 5 m to 10 m.

The results demonstrated a clear and expected trend that an increase in the number of user and street view images corresponded to a reduction in geolocation error. A minimum of three street view images are required to geolocate the SfM model. Below this threshold, it is not possible to geolocate the model using street view pictures. In these cases, the HLoc+SV method falls back to localising the model using only the geolocation of user images, i.e. the HLoc+GNSS method, which is less accurate. In both scenes, using more than ten street view images does not improve the performance significantly. When ten or more street view images are used to build the SfM model, even a single user image can be geolocated accurately.



(a) Scene S1

(b) Scene S6

Figure 4.8: Mean absolute error matrix in metres showing how the number of pictures, both user images and street view images, can affect the geolocation performance.

The observed accuracy trend follows an almost monotonic pattern. The fluctuations are due to the random image selection. In cases when the set of user and street view images show non-overlapping parts of the scene, the construction of an SfM model fails and, consequently, the mean MAE error increases. Despite these variations, the overall consistency in the relationship between the number of street view images and the geolocation performance confirms the expected outcome.

4.5 Discussion

The HLoc+SV presented excellent performance to geolocate user images when street-view reference images are used. Without the use of street-view images, the HLoc+GNSS marginally improves the precision of the user's geolocation due to the robust SfM coordinates transformation using RANSAC.

To illustrate how the HLoc+SV method improves error considerably compared to other methods, consider a specific scenario in which a user captures a series of images in an urban environment with varying levels of GNSS accuracy. We will compare HLoc+SV with GNSS and HLoc+GNSS. Note that the original HLoc method is not suitable for this comparison, as it aims to solve a different problem, i.e. calculate the camera pose within a large SfM model in local coordinates.

In this scenario, the user is walking along the pavement and captures a sequence of 6 images using a smartphone equipped with a GNSS receiver. The GNSS accuracy varies, leading to geolocation errors from 3 m to 30 m. Additionally, a dataset of geotagged street view images of the area is available, which serve as reference points for geolocation.

Using GNSS information alone to estimate the user geolocation is not accurate enough for this navigation task, as it is not possible to know even on which side of the street the user is. It results in inaccurate positioning even when the geolocation error is as low as 3 m,

The HLoc+GNSS method uses the user images to build an SfM model of the scene. It incorporates GNSS information to geolocate the model, which is used to estimate the camera pose in world coordinates. As the GNSS information is the only reference to geolocate the SfM model, GNSS positioning errors propagate into the camera pose estimation process, leading to similar localisation results.

The HLoc$+$SV method integrates geotagged street-view images into the SfM model to refine geolocation accuracy. The GNSS information is used to retrieve street-view images nearby, which are added to the model and serve as 'anchors' to geolocate the whole model.

When street-view images are added to the process, the local coordinates of the SfM model are transformed into world coordinates using accurate geopositions. Although the resolution of Google Street View API images is as low as 640 × 640 pixels, their geoposition is more accurate [10], as demonstrated by the results. In fact, when we align the SfM model using the reference set (HLoc+SV), our method achieves an average mean error of 77 cm (SD 41).

Matching night-time images is a well-known challenge in computer vision. The change in appearance due to illumination, strong shadows, light temperature, and colour makes the appearance of local features change to the extent that they are not recognised as being the same. Associated with that, the change of perspective distorts angles and changes the appearance of local features even further. For small-scale features, the angles change and textures get distorted. For large-scale features, the relative position between objects changes proportionally to the distance between their perspective plan (known as the parallax effect). Combined, these two aspects make it extremely challenging to match local features under such conditions. Nonetheless, the SuperPont deep features and SuperGlue matching algorithm used in HLoc+SV handled well these challenges, achieving a mean absolute error in night scenes of 1.19 m in S6 and 1.17 m in S7, while the GNSS receiver achieved a mean absolute error of 14.21 m and 30.21 m, respectively.

Building the SfM model is the most time-consuming step, which takes about 2 minutes for each scene. The feature extraction runs in real-time at 20 fps. One way of accelerating this process is to build SfM models in advance. There are 3D modelling projects of cities such as London, New York, Tokyo, and San Francisco [167]. Nonetheless, the SfM models must be built using the same feature descriptors (SuperPoint) to be used in this process.

If the mobile internet connection is reliable enough, the features of the user's image can be extracted locally and uploaded to the server, where the building of an SfM model can be done more quickly, potentially in real-time. With the availability of 5G connections in the near future, which is expected to have a very low latency and high speeds, this real-time processing could be possible over the internet. Then, the SfM model can be downloaded by the user and from there the processing can be done locally in real-time.

The HLoc+SV is loosely dependent on the user's GNSS receiver information. The GNSS geolocation is only used to retrieve nearby street-view images. The street-view images' geolocation is then used to transform the SfM local coordinates to world coordinates. The rough geolocation of the user could potentially be obtained in other ways, such as Wi-Fi signal, mobile phone carrier antennas, or even visual identification of the place using only the camera, with no additional sensors.

In scene S1 (top map in Figure 4.7), the GNSS geolocation drifts in all directions. In all other scenes, the GNSS coordinates all drift in the same direction. This behaviour is common in urban centres, where GNSS satellite signals are easily blocked in areas surrounded by tall buildings (called urban canyons) [171]. The use of geotagged street-view images solves this problem. In this way, they act as an anchor to correctly align the SfM model to world coordinates. The estimated camera position on the SfM model after this alignment is regarded as the user position.

There is a marginal improvement in geoposition errors when the SfM model is aligned using geocoordinates reported by the user's GNSS receiver only (HLoc+GNSS). When the set geopositions obtained by the GNSS receptor drift in the same direction, the robust alignment with RANSAC is not very helpful. When positioning is scattered in all directions (e.g. in scene S1), RANSAC is able to fit the SfM model to the position of the user with more precision. Nonetheless, the HLoc+GNSS is not precise enough for visual pedestrian navigation.

Figure 4.8 revealed a clear trend that increasing the number of street view images consistently led to a reduction in geolocation error. While user images provide valuable context and contribute to building consistent SfM models, street view images play a critical role in enhancing accuracy. Error fluctuations occurred due to random image selection and non-overlapping scene parts, which made the construction of SfM models fail. Overall, the trend confirmed a consistent relationship between the number of street view images and geolocation performance. This experiment also revealed that a minimum of three street view images is necessary to build an SfM model and, for the two scenes analysed, using more than ten street view images does not increase the performance significantly.

Finally, the Google Street View is not the only street-view service available. There are other services that can be used instead. The Apple MapKit, for example, provides street-view images for Apple devices only. Recently, the company announced the Apple Maps Server API, for access from any device through HTTP requests. Although similar to MapKit, the Maps Server API does not yet provide access to streetview images. Another alternative is Mapillary, a platform that makes street-level images and map data uploaded by users freely available to everyone. Their coverage is not as wide as Google's yet, and the geolocation of uploaded images are manually set by the users. The Google Street View API also has a partnership program with selected universities to provide high-quality street view images with a resolution of 2048×2048.

4.6 Conclusion

In this chapter, we detailed the construction of the HLoc+SV, a vision-based geolocation method based on a version of the hierarchical localisation framework that exploits information from a dataset of geotagged street-view images. For the experiments, we collected 58 street-view pictures from seven places in the United Kingdom and downloaded 80 geotagged images on demand from the Google Street View service. Comparing our results with a smartphone GNSS receptor, we see that the mean absolute error decreased from 12.09 m (SD 8.67) to 0.77 m (SD 0.41). Extremely challenging scenarios for GNSS presented the greatest improvement in performance, even when night-time pictures were considered. When the mean absolute geoposition error of the GNSS in a night scene is as high as 30.21 m, the error with HLoc+SV is 1.17 m. Overall, a minimum of three street view images is necessary to build an SfM model. Where ten or more street view images are available, even a single user image can be geolocated accurately.

The resources used in these experiments – GNSS receptor, camera, and internet connection – are all available in smartphones, which makes this solution have the potential to be used in end-user applications.

These results add evidence to the argument that vision-based solutions can improve the accuracy of geolocation estimation. Nonetheless, would be interesting to analyse the performance of HLoc+SV on a larger dataset of user pictures, using higher resolution street-view reference images with information about the equipment used, and ground truth data collected from a high precision GNSS receiver (such as RTK GNSS).

The HLoc+SV is a potential solution only suitable for the scenario when the GNSS service is available and relatively accurate. When the GNSS service is unreliable or unavailable, we need to explore other solutions. This is what we will cover in the next chapter.

Chapter 5

Geolocation using image retrieval and light normalisation

A number of applications rely on GNSS service worldwide, although its availability or reliability is far from granted. There are several scenarios in which the GNSS service may suffer interference or be completely blocked [171]. The GNSS signal is easily tampered with by physical obstructions such as mountains, trees, or tall buildings. Even when the GNSS signal is not completely blocked, these obstructions reflect the electromagnetic waves before reaching the receiver antenna, causing an effect called 'multipath'. The reflected signal combines with the original signal and creates interference, resulting in errors in the calculated position. Electrical structures such as telecommunication towers, high-tension equipment, radar systems and even smaller electrical devices may also interfere with GNSS receivers. These effects are easily observed in highly urban areas, where it is common to have inaccurate readings or complete loss of GNSS signal.

Global events can affect GNSS coverage on a larger scale. Weather conditions, such as heavy rain or snow, or atmospheric events, such as solar flares, can disrupt the GNSS service in an entire city or county. The GNSS satellites are also subject to outages due to system malfunction or maintenance.

The GNSS service can be easily disrupted on purpose by jamming or spoofing. A malicious agent can do this by broadcasting electromagnetic waves at the same frequency as GNSS signals. Although illegal in many countries, GNSS jammers can be purchased online and are often used by individuals or organisations that want to prevent tracking or surveillance. In a war, military forces may jam or disrupt GNSS signals to prevent their adversaries from using this technology. Some countries may choose to degrade or even turn off GNSS signals to civilians to prevent enemies from using the GNSS service for navigation or targeting.

There are few alternatives to geolocating an individual when the GNSS service is unreliable or not available. Leaving aside analogical options such as celestial navigation, paper maps, and magnetic compasses, vision-based approaches using images or video frames are a good alternative in this scenario.

In this chapter, we study the problem of geolocating a single image with no previous geolocation information. There is a wide range of approaches in the literature trying to solve this vision-based geolocation problem. Many of them require additional sensors or more information than a colour image [6, 104, 158]. Recent approaches involve the use of convolutional neural networks [160] and nearest neighbour local feature matching [65, 169].

A review of single-image geolocation methods with a brief explanation of techniques is presented in Section 2.4. Despite several attempts to increase the accuracy of geolocating a single picture taken outdoors, the best results we could find in the literature are not suitable for safely building a navigation assistant with the requirements described in Chapter 2 when no geolocation data is available. Furthermore, most proposed solutions in this area do not discuss the role and impact of different local descriptors in this task. The manipulation of illuminance and shadows on photographs is an option to increase the performance of image matching under such a challenging scenario.

The question of estimating and eliminating the effects of illuminants is a highly active research field [8, 41, 56, 153, 157]. These methods produce images partially or fully invariant to illuminants by producing a derived representation of the image that does not change when the illumination changes. Various methods are proposed in the literature for obtaining post-processed images that are invariant to illumination colour, intensity, shading, and specularities. In Appendix A, we review two of these methods and also briefly explain the process of image formation.

This chapter concerns creating and analysing a geolocation method for when there is no GNSS service available based on image retrieval. We use a method based on a Generalised Minimum Clique Problem (GMCP) image retrieval framework, and attempt to further improve it by normalising the illuminance of images with the Self Quotient Image algorithm. We test the geolocation estimation on the UCF Street View dataset, which contains more than 62,000 Google Street View images of three cities in the US. Furthermore, we compare the performance of SIFT, SURF and MSER local descriptors on a smaller dataset of 400 street-view images taken years apart.

When we ran the experiments reported in this chapter, deep features were in an early development stage. For this reason, we do not approach them in this chapter.

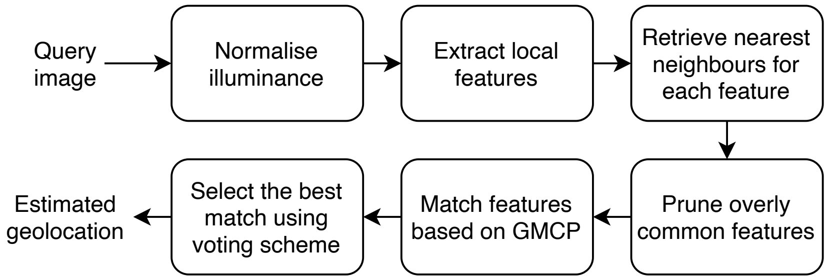

Query image
Normalise illuminance
Extract local features
Retrieve nearest neighbours for each feature
Prune overly common features
Match features based on GMCP
Select the best match using voting scheme
Estimated geolocation

Figure 5.1: Sequence diagram of the image retrieval and geolocation task using the GMCP-based framework and Self Quotient Image to normalise the effect of illuminants.

5.1 Method

We aim to geolocate a query image taken by the user by retrieving the closest street-view image from a reference set. We perform the image retrieval and geolocation task using a GMCP-based framework [169], preprocessing all images with the Self Quotient Image filter to remove the effect of illuminants. Figure 5.1 shows a sequence diagram of this experiment. Please see Section 2.4 for more details on GMCP-based image retrieval and the Appendix A for details on the SQI algorithm.

In the original GMCP-based approach [169], SIFT local features are extracted from all images in the reference set and are organised in a kd-tree. Given a query image, its SIFT features are extracted and, for each feature, the k nearest neighbours are retrieved from the kd-tree [102].

In contrast to the assumption that the first nearest neighbour is the optimal choice for local feature matching, the original GMCP-based method acknowledges that this may not be the case when multiple very similar local features are present (such as undistinctive features present in trees, cars, and roads, for example). For this reason, they exclude local features that are too common in the dataset. The remaining features are then matched to multiple features in the reference set.

A GMCP-based feature matching technique is then employed to identify a single nearest neighbour match for each query feature, ensuring that all matches are globally consistent. Once the GMCP is solved, a voting scheme is employed to elect the strongest reference image match.

The geolocation information of the images in the reference set is converted to Cartesian coordinate values to be used as a global feature. They use a robust distance function to measure the similarity between global features and select the reference image with the greatest number of feature matches. The idea is that clusters of reference images geographically close to each other are prioritised in this voting scheme. In the end, the geolocation of the winning reference image is returned as the estimated geolocation of the query image.

In our method (GMCP+SQI), we first transform the query image and all the images in the reference set into an illuminant-free image space using the Self Quotient Image (SQI) algorithm. In this way, the differences between images due to changes in illuminance are removed, therefore we expect to have a greater number of SIFT features correctly matched. The images processed with the SQI filter are then fed into the GMCP-based framework.

We also analyse two alternative voting schemes: counting the number of matching features (GMCP+SQI w/ weights) and estimating the geometric transformation (GMCP+SQI w/ geom. transf.). We estimate which framework provides the best result (GMCP or GMCP+SQI) considering these two alternative voting schemes, and selected it as the final answer.

For the GMCP+SQI w/ weights, the decision parameter for the voting scheme is the number of matched features considering all images in the reference set. In other words, for every image in the reference set, we count one point for each matched feature remaining in the best GMCP solution. Thus, the image with the greatest cardinality of matched features is the winner. Finally, the last step is to estimate the camera pose.

5.2 Fundamental matrix

Calculating the camera pose involves determining the position and orientation of the camera relative to a reference image (2D) or scene (3D). In this section, we discuss the method to estimate the pose of uncalibrated cameras relative to a reference image using fundamental matrices. A more detailed explanation can be found in [62, 63].

The first step is to detect local features in both query and reference images. A feature matching algorithm is then used to find the corresponding features between the two images. Common methods for feature matching include nearest neighbour search [102] using distance metrics such as Euclidean distance or Hamming distance, followed by a ratio test to remove outliers and ambiguous matches [88].

Using the correspondences obtained from feature matching, the essential matrix (for calibrated cameras) or the fundamental matrix (for uncalibrated cameras) can be computed [62]. The essential matrix relates the camera poses between two views, whereas the fundamental matrix describes the epipolar geometry between two images.

The fundamental matrix conveys the geometric relationship between two views of a scene captured by cameras with arbitrary motion and intrinsics. Formally, the fundamental matrix F relates the corresponding points in two images through the epipolar constraint [62], i.e. given a set of corresponding points $p_i = (x_i, y_i, 1)^T$ in one image and $p'_i = (x'_i, y'_i, 1)^T$ in the other image, the epipolar constraint is such that $p'^T_i F p_i = 0$.

The fundamental matrix can be estimated from these correspondences using various methods [62]. The simplest one is the 8-point algorithm, in which the fundamental matrix F is calculated by solving the linear system $Af = 0$, where A is the coefficient matrix constructed from the point matches and f is the vectorised form of F.

Given a set of N point correspondences $\{p_i, p'_i\}_{i=1}^N$, the goal is to find the fundamental matrix F that satisfies the epipolar constraint $p'^T_i F p_i = 0$ for all i. For N point correspondences, we obtain N linear equations of the form $p'^T_i F p_i = 0$. Since F is defined up to scale, it is usually constrained to be of rank 2 by performing singular value decomposition (SVD) and null space extraction. Methods such as RANSAC [43] and MSAC [146] can also be applied to robustly estimate F from a set of noisy matches. It is possible to calculate F 'when only 7 point correspondences are known' [62, p. 281].

In the normalised version of the 8-point algorithm, the point correspondences are first normalised by transforming the point coordinates such that they have zero mean and standard deviation equals 1. After estimating F in the normalised space, it is denormalised to obtain the final estimate.

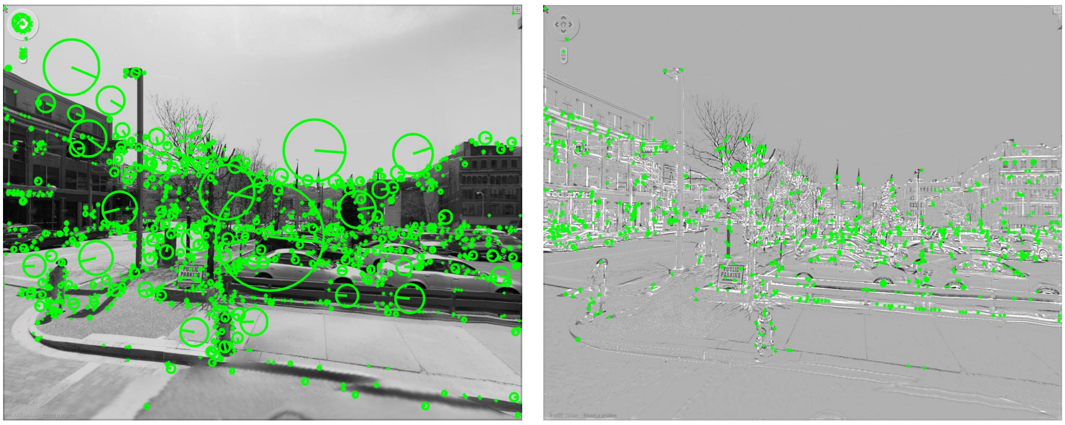
Once the fundamental matrix F is estimated, it can be used to enforce geometric constraints (such as the epipolar lines) and to extract the camera pose and 3D structure of the scene. Given a point p_i in one image, its corresponding epipolar line l'_i in the second image can be computed as $l'_i = Fp_i$. Similarly, for a point p'_i in the second image, its corresponding epipolar line l_i in the first image can be computed as $l_i = F^T p'_i$. Therefore, the epipolar lines establish a relationship between any p_i and p'_i without knowing the actual position of such points in 3D space or the intrinsics of the cameras. The relative pose between the two cameras can be calculated from $\boldsymbol{F}$ [62].

For the GMCP+SQI w/ geom. transf., we attempt to robustly estimate the fundamental matrix $\boldsymbol{F}$ for each image from the reference set with at least three matched feature points using MSAC. The image with the greatest reciprocal condition of $\boldsymbol{F}$ is considered the best image match. When it is not possible to estimate $\boldsymbol{F}$ for any image, we use the original voting scheme (using global features) on the GMCP method to decide the winning reference image.

5.3 Effect of light normalisation on features detection

Before conducting the experiments reported in Section 5.5, we visually checked the impact of SQI light normalisation on the detection of local features. As we can see in Figure 5.2, the Self Quotient Images only contain small-scale SIFT features. This is due to the low contrast of processed images and the SIFT detection method based on the Difference of Gaussian (DoG).

This approach has the advantage that small-scale features focus on details such as doors, windows and textures. This type of information is more stable and is less prone to variation when compared to a broader view of the scene. Large-scale features change with the scene composition of different planes and are more strongly affected by the parallax effect.



(a) Original **(b)** SQI

Figure 5.2: Example of light normalisation using SQI. (a) Original and (b) processed images. SIFT features are illustrated in green. Only small-scale SIFT features are detected in the image processed with SQI.

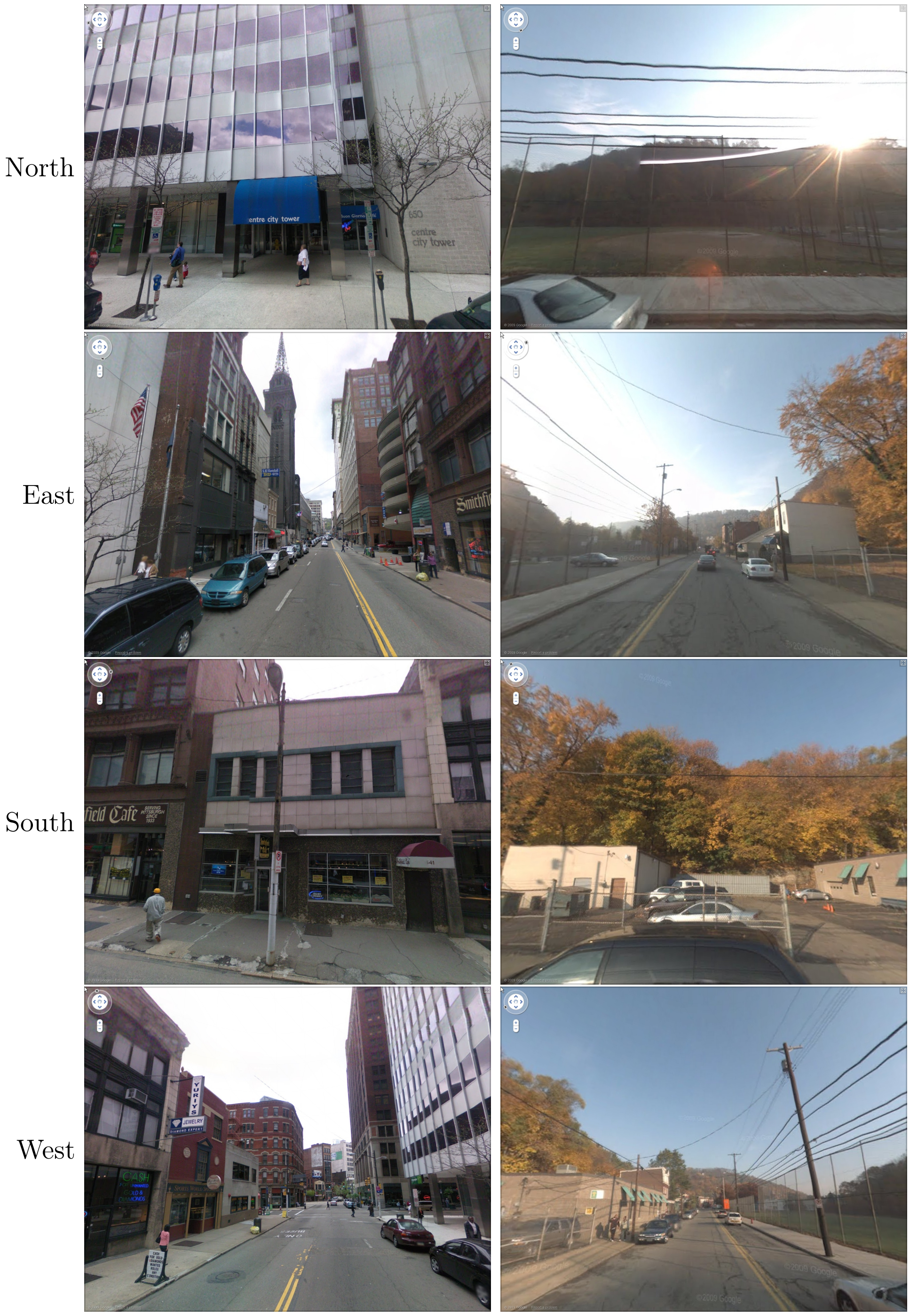
5.4 Datasets

In this section, we introduce the three datasets used in the experiments reported in Section 5.5.

5.4.1 Reference set

We use the UCF street-view dataset [169] containing 62,058 images from the Google Street View service [51] at a resolution of 1281×1025 pixels each. We chose this dataset because it was the one used to test the original GMCP-based image retrieval framework [169]. It allowed us to directly compare the results reported in this chapter with the results achieved by the original GMCP-based geolocation framework. Furthermore, when we conducted the experiments, it was one of the most comprehensive structured street-view datasets publicly available including camera pose information and geocoordinates.

This dataset covers areas in three US cities: Pittsburgh, Orlando, and



(a) 676 Smithfield St

(b) 226 Fort Pitt Blvd

Figure 5.3: Street-view sample images from the reference set taken in Pittsburgh, US. Each column corresponds to a placemark.

Manhattan. Pictures are taken every 10 m along the road. Each street-view placemark is divided into four side views (north, east, south and west). Although these are daytime images, there is no information about the date or time when they were taken. Figure 5.3 shows some examples.

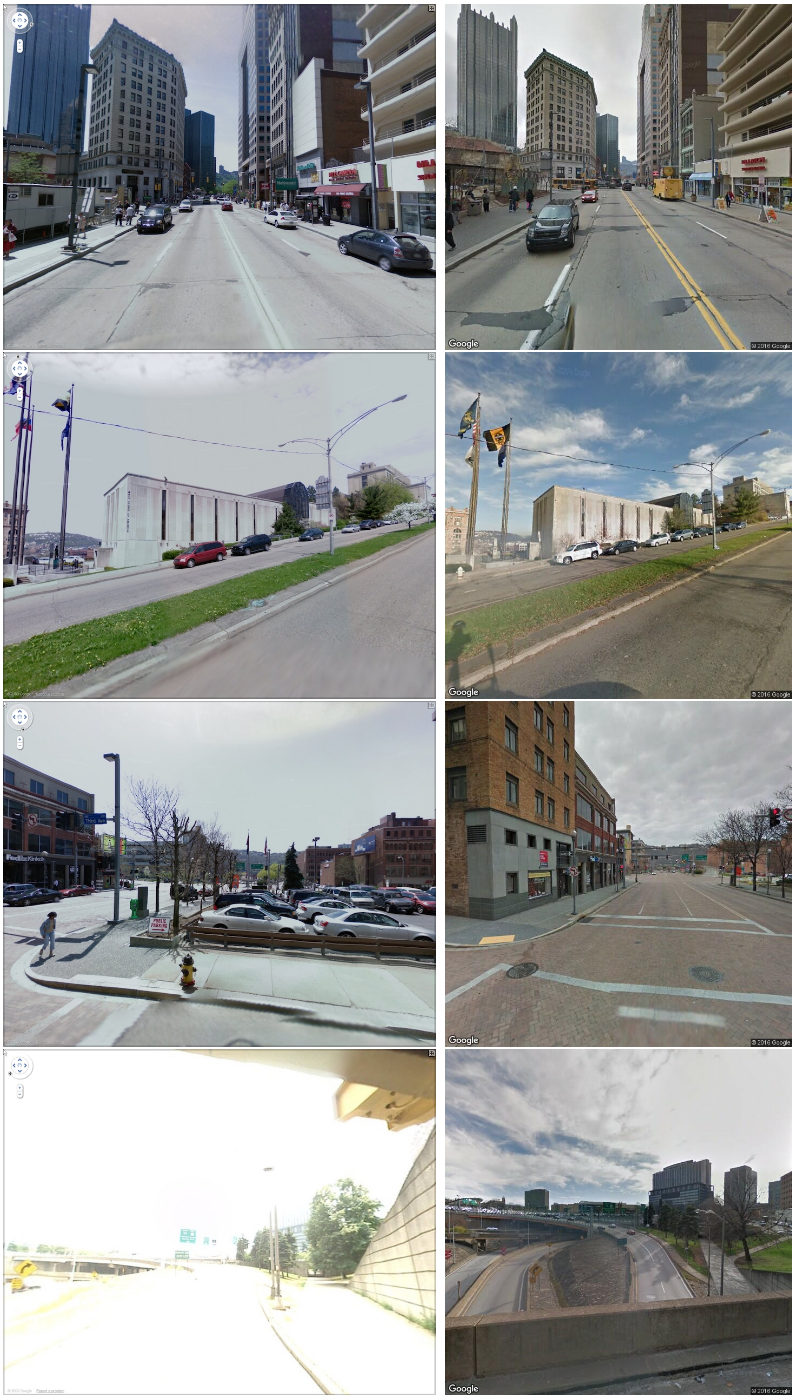
For each street-view placemark (i.e. each spot along the road), a 360° spherical view is broken down into four side views and an upward view.

5.4.2 Query set

We created a dataset with four hundred images retrieved from the Google Street View API. They correspond to a hundred randomly selected placemarks in the reference dataset, four images per placemark (with the camera pointing to the north, east, south and west). All new images have a 640×640 pixels resolution and were taken during the day. Some samples are shown in Figure 5.4.

The illuminance and point-of-view of these new images are remarkably distinct from those in the reference dataset. Although all the photos in both datasets were taken every 10 m along the streets, the position of the placemarks does not necessarily lie in the same place. For each placemark, there is a camera shift of up to 5 m.

In short, the most significant differences identified between the datasets are (i) the illuminance; (ii) the image resolution; (iii) the point of view, which changes up to 5 m; (iv) the camera lens; (v) the light exposition and saturation; (vi) the foliage, depending on the season; (vii) the improvements on roads and pavements; (viii) the changes on buildings; and (ix) the areas occluded by buildings' maintenance and cars.



Reference set Query set

Figure 5.4: Street-view images from the reference and query sets taken in Pittsburgh, US. Each row corresponds to the same placemark and same camera direction.

5.4.3 Test set

The UCF street-view test set [169] has 644 images retrieved from Picasa, Panoramio and Flickr, geotagged by users, and manually corrected afterwards. Seven sample images are shown in Figure 5.5. The resolution of images in this test set is not uniform across all images, although all images have a resolution greater than the resolution of images in the street-view reference set (1281×1025 pixels).

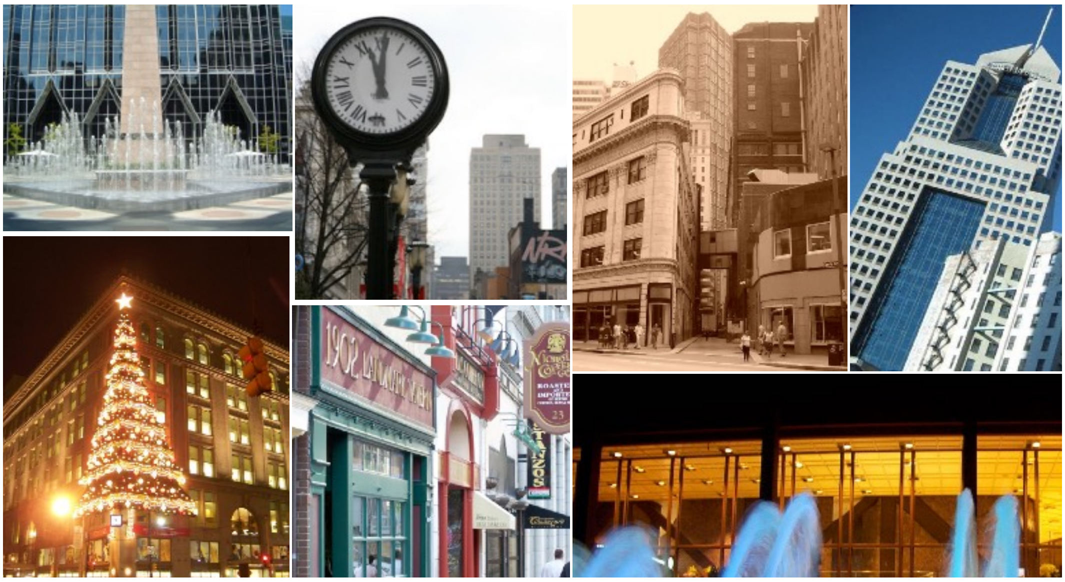


Figure 5.5: Samples from the UCF street-view test set. From [169].

5.5 Results and discussion

In this section, we describe the details of our evaluation and present the results for the feature matching. In Section 5.5.1, we use five images of the same scene to evaluate the performance of SIFT, SURF and MSER, and fine-adjust settings. In Section 5.5.2, we analyse the performance of SIFT, SURF and MSER feature matches on correspondent image pairs in the reference and query sets. Finally, in Section 5.5.3, we evaluate the use of GMCP–based methods with and without SQI normalisation using alternative voting schemes.

In all experiments, the SURF and MSER features were detected using the MATLAB 2016b Computer Vision toolbox [144] and the SIFT features were calculated using the open source VLFeat toolbox [150]. The Large scale GMCP–based image geolocation framework run using the MATLAB 2016b on the UEA High Performance Computing Cluster, optimised for parallel processing.

5.5.1 Single case image matching

A single image was randomly selected from the reference set. Then four similar images were downloaded from the Google Street View service [51] with a progressive change in the camera's perspective and illumination conditions. The settings used to detect the features are listed in Table 5.1. The images retrieved for the single case image matching can be seen in Figure 5.6.

Table 5.1: Settings used on single image pair matching.

Detector	**Descriptor**	**Settings**
SURF	SURF	Threshold of 50 and 15 square filters from 9 to 93 pixels, inclusive.
MSER	SURF	Step size between intensity level of 0.8 %, a minimum area of 9 pixels and maximum area variation between extremal regions threshold of 1.0.
DoG	SIFT	Difference of Gaussians (DoG) detection method, peak threshold of 10^{-4} and estimation of affine adaptation [99] and orientation of features.

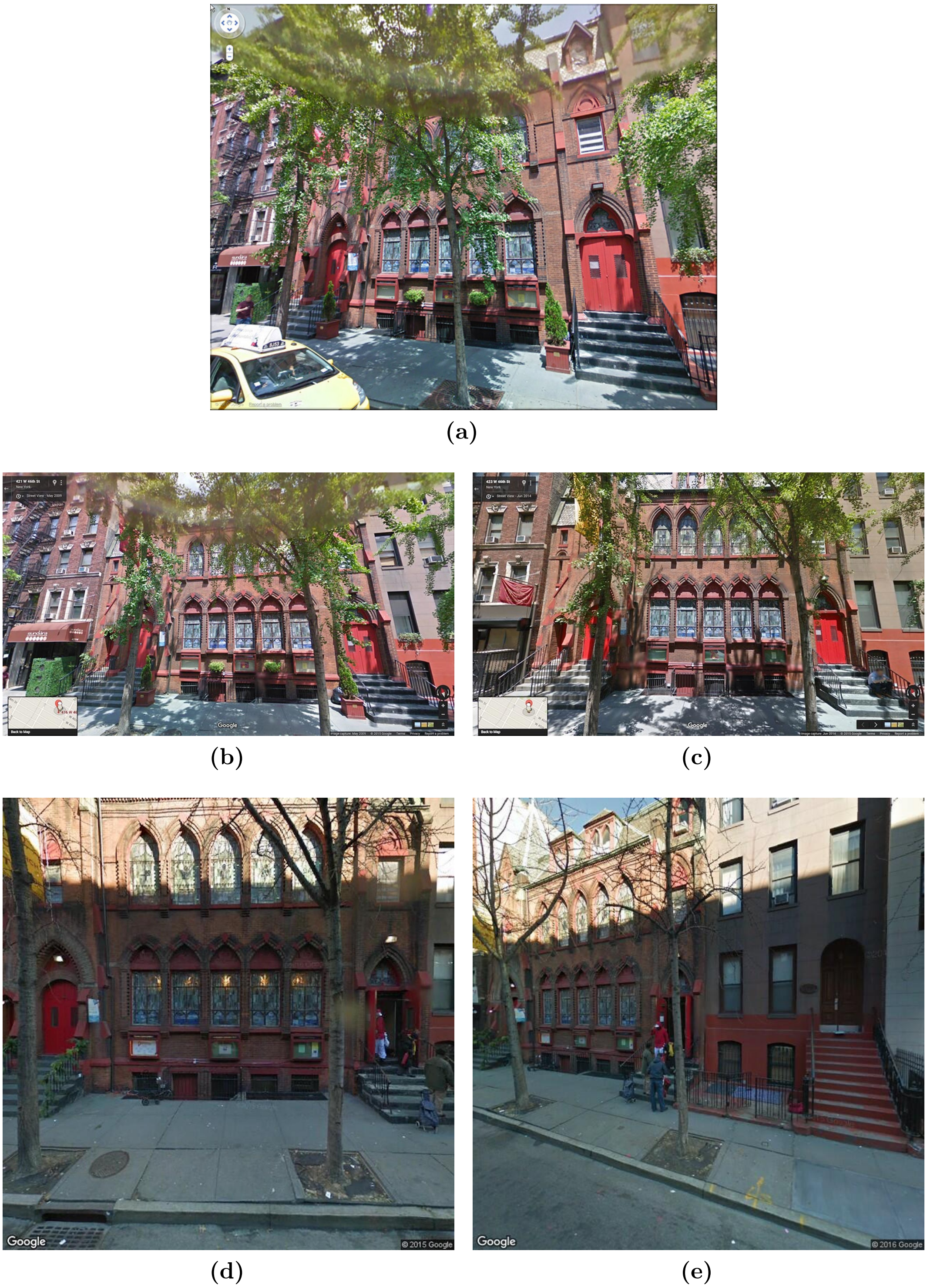


Figure 5.6: Images used in this experiment. (a) Reference and (b-e) query images. The query images are sorted by visual similarity to the reference image.

Fundamental matrices were estimated using the MSAC method with a maximum number of random trials of 8000, confidence of 99 %, and maximum distance from point to projection of 10 pixels. Matrices with a reciprocal condition index lower than 10^{-7} were considered poorly conditioned and discarded. To evaluate the performance of the feature matching, we compare the number of matches in each pair of images. In this way, we observe the descriptors' invariance due to differences in extrinsic conditions, e.g. point of view and illuminance.

A comparison of the results is shown in Figure 5.7. The SIFT descriptor showed the best performance among all evaluated algorithms; in all four cases, it was possible to generate consistent fundamental matrices from SIFT matching features. Although there are 6 to 9 matching features in (d) when SURF and MSER algorithms are used, it was not possible to generate well-conditioned fundamental matrices.

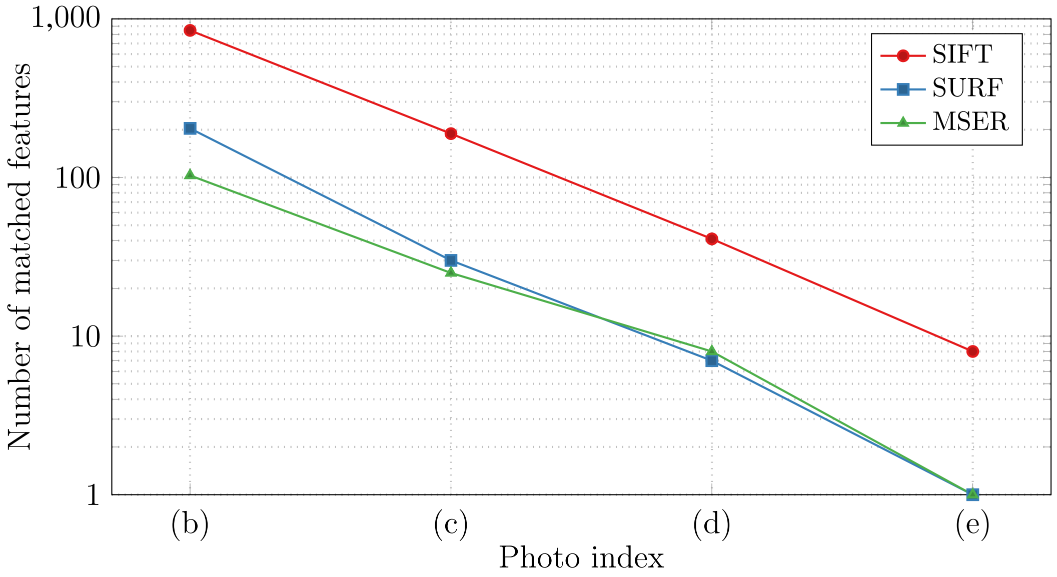


Figure 5.7: Comparison of SIFT, SURF and MSER feature matching on the photos presented in Figure 5.6.

Although we can find in the literature reports of performance decrease when illumination and perspective change separately, these experimental results indicate that the combination of illumination and perspective change has the potential to cause an exponentially negative effect on the performance of all local features analysed.

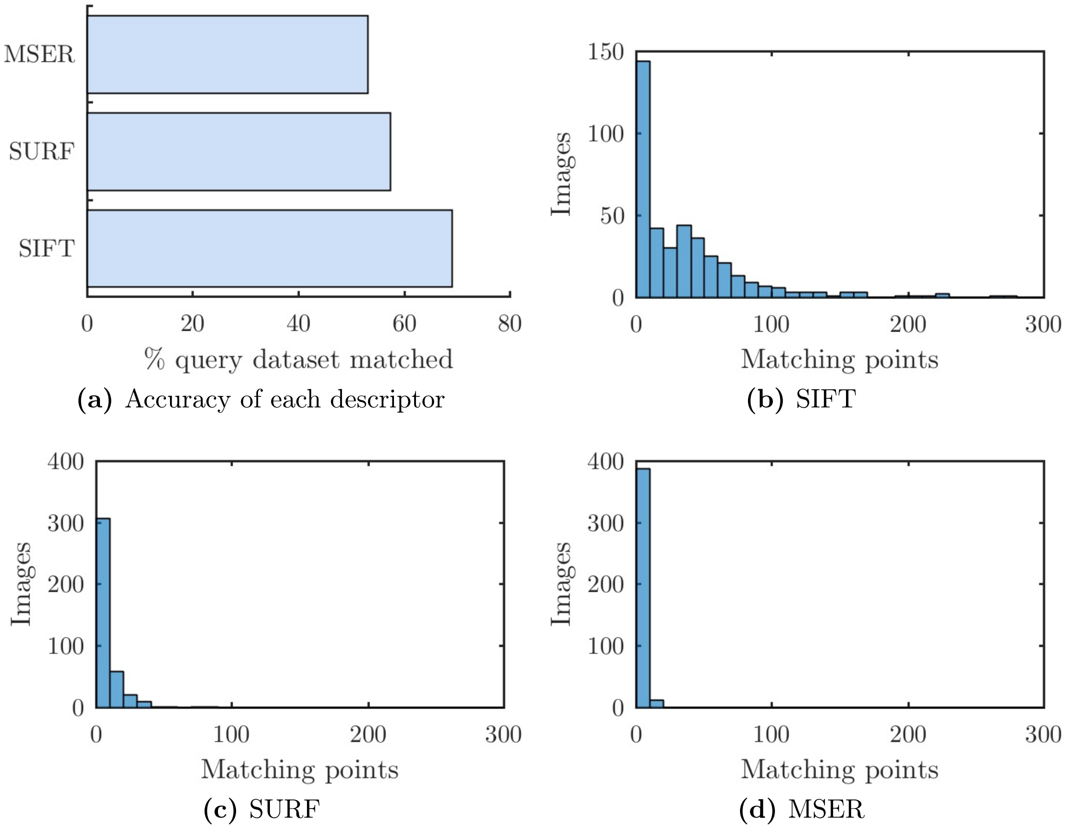
5.5.2 Image matching on query set

This experiment concerns the efficiency of SIFT, SURF and MSER local features detectors and descriptors on matching street-view images with simultaneous changes in perspective, illumination and occlusion. Our approach to exploring this question is to evaluate the accuracy of SIFT, SURF and MSER by measuring their repeatability rate when applied to this problem. Ultimately, we hope to answer to what extent matching out-of-the-box local features can be useful in street-view images taken years apart.

To verify the accuracy of matching features through a range of images, we matched the images in the query set with their corresponding images in the reference set. The parameters were the same as used previously.

The robustness of descriptors was measured by the number of inline matching points. Results are shown in Figure 5.8. The SIFT descriptor (b) was the most robust, generating consistent fundamental matrices for 69 % of the query set, followed by SURF (c) with 57 % and MSER (d) with 53 %. The robustness of SIFT was confirmed on the histogram of inlier features in each image (b). Figure 5.9 shows some examples of matches.

The original GMCP-based method for street-view image retrieval uses SIFT features [169]. Although they do not discuss the reasons for choosing SIFT,



(a) Accuracy of each descriptor

(b) SIFT

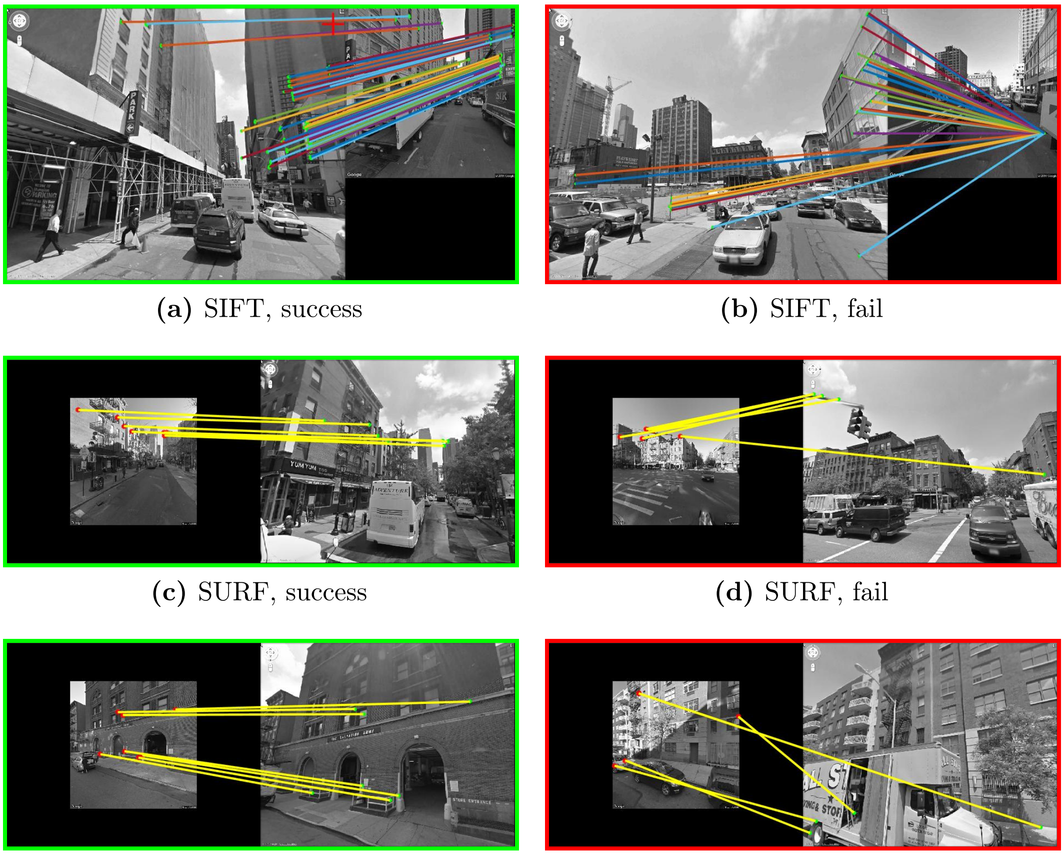
(c) SURF

(d) MSER

Figure 5.8: Comparison of SIFT, SURF and MSER feature matching results for the images in the query set. (a) Bar chart with the accuracy of each method. (b-d) Histograms of the number of matching features in each image pair.

our results confirm that SIFT outperforms other feature detectors and leads to more accurate matching features on street-view images.

Although SIFT outperforms SURF and MSER, 31 % of the query dataset images does not generate consistent fundamental matrices, even when the selected image pairs to be matched are known to be from the same place. We can conclude that any solution based on matching these local features with no image pre-processing is limited to an accuracy of 69 %.



(a) SIFT, success

(b) SIFT, fail

(c) SURF, success

(d) SURF, fail

(e) MSER, success

(f) MSER, fail

Figure 5.9: Sample cases of the query set matching test. Left: successful cases. Right: cases where it was not possible to estimate a fundamental matrix.

5.5.3 Large scale GMCP–based image retrieval

In this section, we evaluate our GMCP+SQI method. For this experiment, we used the following hyper-parameters:

Pruning Multiple-nearest neighbour pruning is used to remove query features that do not have distinctive NNs. The threshold value of 80 % when comparing the similarity between the first and $(k+1)$th feature is empirically found in the literature to be optimal [88, 169], so we set the pruning similarity threshold = 80 %.

Number of nearest neighbours k The optimal value of k for the feature matching process depends on the extent of similarity and repetition in the reference set features. If k is too small, there may be insufficient nearest neighbours considered, leading to a reduced likelihood of identifying the correct one. In contrast, an overly large value of k would result in too many nearest neighbours included in the input graph, increasing the complexity of the optimisation task. The threshold value of $k = 5$ is empirically found to be optimal for this task [169], so we set $k = 5$ in our experiments.

Global feature The geolocation information is converted to Cartesian coordinate values to be used as a global feature.

Distance function The distance function is used to measure the similarity between global features. In these experiments, we use the robust function D and the empirical value $\sigma = 256$ when geolocation is used as a global feature (see Section 2.4 for more details on D and σ).

Figure 5.10 shows the results of evaluating this method compared to the baselines in terms of overall geolocation results using reference and test sets. The solid red curve shows the performance of the original GMCP–based feature matching [169], and the solid blue curve is the performance of our method.

The dotted cyan curve in Figure 5.10 shows the results using the voting scheme with weights. The dotted magenta curve illustrates the results with a voting scheme that considers the conditioning of the geometric transformation matrix. In very few cases, it improved the results compared with the original GMCP–based approach. In general, there were not enough feature matches to

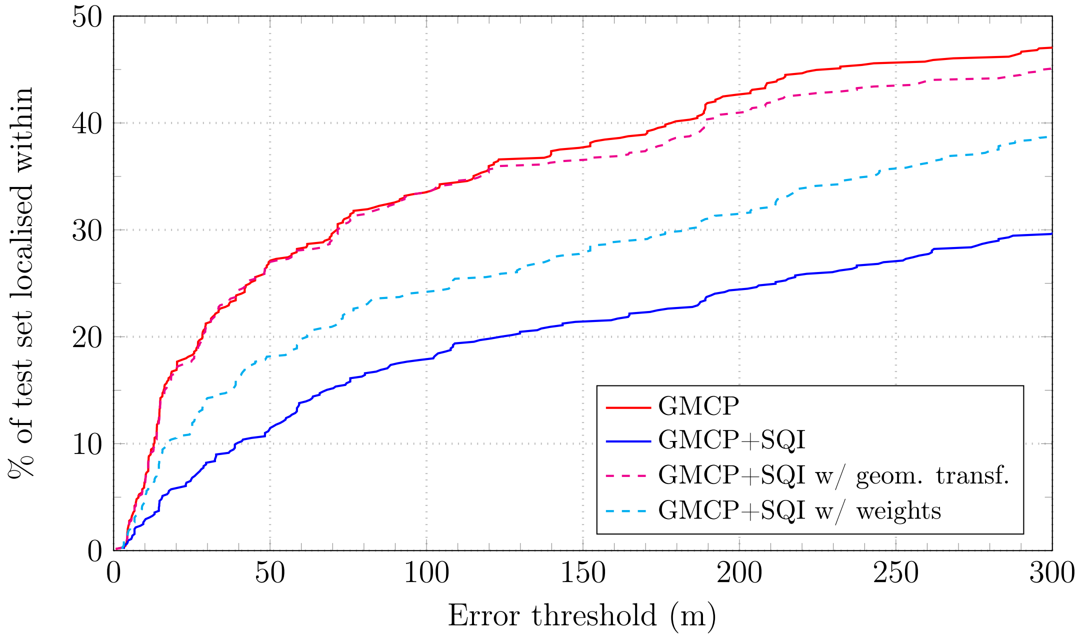


Figure 5.10: Cumulative distribution of absolute geoposition errors using GMCP and SQI, as well as using alternative voting schemes (geometric transformation and weights). The horizontal axis shows the distance threshold and the vertical axis represents the percentage of the test set localised within the distance threshold.

estimate a geometric transformation, which is the first step in estimating the camera pose (see Section 5.2).

The lack of feature matches is an inherent characteristic of this GMCP–based method, which retrieves a single reference image closest to the query image. The process involves identifying corresponding local features across several images in the same location and, as a result, feature points corresponding to the same physical location are scattered across different images. Thus, in most cases, the reference image alone is not enough to calculate the fundamental matrix needed for accurate camera pose estimation.

Large scale features that cover large areas present in the GMCP method can be safely removed using the SQI algorithm. Small areas are more stable and less affected by changes in perspective. The matching needs to be reliable, even if only a few points are associated.

5.6 Conclusion

In this chapter, we explored the use of multiple nearest neighbour feature matching (kNN), generalised graphs (GMCP) and illuminant normalisation with Self Quotient Image (SQI) for the task of single street-view image geolocation. Two street-view datasets were specifically built for these experimental evaluations.

The results show that SIFT performed better in this task, being able to deliver enough points to generate fundamental matrices for 69 % of the query dataset. Nonetheless, the performance of all detectors evaluated drops exponentially when there is a simultaneous change in illumination and point of view.

We found out that there is no performance improvement with use of SQI with the GMCP–based image geolocation framework. Moreover, there was a degradation of 4 % in results when the estimation of geometric transformation was used to combine the results of normalised and non-normalised images. We found that the SQI prevents the detection of large SIFT features due to the reduction of contrast between objects. Finally, it is interesting to see that the estimation of the geometric transformation was prejudiced by the distributed feature matching system introduced by GMCP.

Despite our efforts to calculate the camera pose, it was not possible to estimate a geometric transformation based on the images retrieved using this method. In most cases, there were not enough inlier matches to estimate a fundamental matrix.

Some factors were decisive for this outcome. The significant difference in perspective changes MSER and SURF descriptors to the extent that features from the same points become dissimilar. With SIFT features, on the other hand, perspective distortion is partially mitigated by affine adaptation, although it creates more mismatches. All three methods are partially invariant to illumination, yet their performance drops drastically when both perspective and illuminance change simultaneously.

Illumination plays a crucial role in the performance of feature detection on street-view images, especially when changes in illumination are combined with a perspective shift. Considering that it was not possible to improve the performance of feature matching using images normalised with SQI, in the next chapter, we investigate the impact of SQI on the accuracy and repeatability of local feature detectors and descriptors by progressively removing details in images using a graph-morphological algorithm known as Sieve.

Chapter 6

Impact of illuminance normalisation on local feature matching

There are disadvantages to using illuminance normalisation algorithms on images of urban areas, where there is a predominance of flat areas with sharp delineations between objects and buildings. Broadly speaking, illuminant normalisation algorithms based on retinex [84] compare averages of the scene, which gives the average of the illuminant, with the specifics of the scene. This approach blurs flat areas, destroying useful information. Retinex algorithms work well in scenes illuminated by a single constant diffuse illuminant, where blurring does not affect so much. In a forest, for example, retinex would probably work well, as the trees would diffuse the sunlight. In urban scenes, a segmentation algorithm to process and extract information from naturally well-defined built areas would perform better. In the class of segmentation algorithms, the Sieve is an interesting one, allowing the separation of connected components by scale.

Very small-scale information (such as texture) is unlikely to be reliable for feature matching in urban scenes. On the other end, large-scale information is excessively big for the buildings. There must be a threshold in between that gives us information about this matching process. In most photos, we can identify corners, windows and texture variation in a 10×10 pixels window.

We expect that features used for geometric alignment do not span multiple scales in the Sieve. Very small-scale information is unlikely to be reliable for feature matching, and at a large scale is information that contains the illuminant. In this way, systematically removing those scales serves as a proxy for solving the illuminant. The illuminant is expected to be removed when large-scale information is removed, while small-scale information is intact.

In this chapter, we investigate the impact of normalising the illuminance on images using the Self Quotient Image (SQI) filter on the accuracy and repeatability of local feature detectors and descriptors. We use a graph-morphological algorithm known as *Sieve* to progressively eliminate image details and analyse differences between original images and SQI images at each scale of detail. We aim to learn more about the performance of SIFT and MSER at each scale of detail, as well as the impact of Self Quotient Images on this process. A novel dataset was created to conduct this study on a controlled set of images carefully aligned. We present suggestions for improvement and further research in the last section of this chapter.

When we ran the experiments reported in this chapter, deep features were in an early development stage. For this reason, we do not approach them in this chapter.

6.1 Dataset

We built a new structured image dataset to analyse the challenging factors involved in matching street-view images. A favourable scene for this study has the following desirable characteristics:

Landmark An ideal landmark for this study must be well known so it would be easier to get images under a wide range of conditions and facilitate the dataset expansion in the future.

Almost plane facade A landmark with a planar front makes it easier to study transformations close to projective rather than epipolar. It is easier to manually align the images, obtain a ground truth and check the accuracy of reprojections. Nonetheless, it also allows studying the effect of projective and epipolar projections at the same time, considering structures and objects out-of-plane with the building facade. Buildings with complex structures would take a long time to accurately align.

Variety of sources This allows us to analyse the effect of extreme changes when the landmark is projected on a plane. Images taken from distinct cameras and points-of-view allow a camera-independent approach. At the same time, a variety of light conditions and weather becomes essential to investigate the effect of the illuminance in such scenarios. Therefore, this dataset must have images from a variety of sources, e.g. Google Street View, Flickr, Panoramio, Instagram and Facebook.

Clear view Occlusion is a common problem in image matching, partially solved by the bag-of-features approach. Yet, it becomes a real challenge when it is combined with all the other factors mentioned above.

The historical house of Anne Frank (Figure 6.1) meets all the mentioned requirements. It has a simple and almost completely flat facade, which facilitates a manual alignment. Its neighbouring houses are slightly out-of-plane with some projecting elements, e.g. hangs, windows, surveillance cameras, doors and roofs. It is exhaustively photographed by tourists and shared online on social media. The building has a wide-open area in front, allowing people to take pictures from distinct points of view, including the other margin of the canal. It is easy to find pictures of the Anne Frank house under very distinct light conditions and weather, day and night, from 0° to almost 90°, i.e. front view to almost absolute side view. There is also a tree about 8 m tall in front of it, which helps to study the effect of occlusion consistently.

We so define a new dataset with twenty-two photos of the Anne Frank house in Amsterdam, geocoordinates $52^\circ\,22'\,31.2''$ N, $4^\circ\,53'\,2.6''$ E. The images were collected online, taken from different cameras and under different circumstances. Figure 6.1 shows some examples of images in this dataset.

We analyse the relative variability of our dataset by hand aligning all images and projecting them all onto a canonical image. It can be observed in Figure 6.2 that the Anne Frank house (at the centre of both images) is sharper than its surrounding buildings. This is due to the effect of projective transformation and lens distortions. Elements out of the chosen alignment plan appear with a ghost effect or, at extremes, are completely vanished. The 8 m tall tree in front of the house is an example of that. It is a few meters away from the facade and its appearance changes with the seasons. The large arm and hook sticking out from the top of the building disappear as they are also out-of-plane.

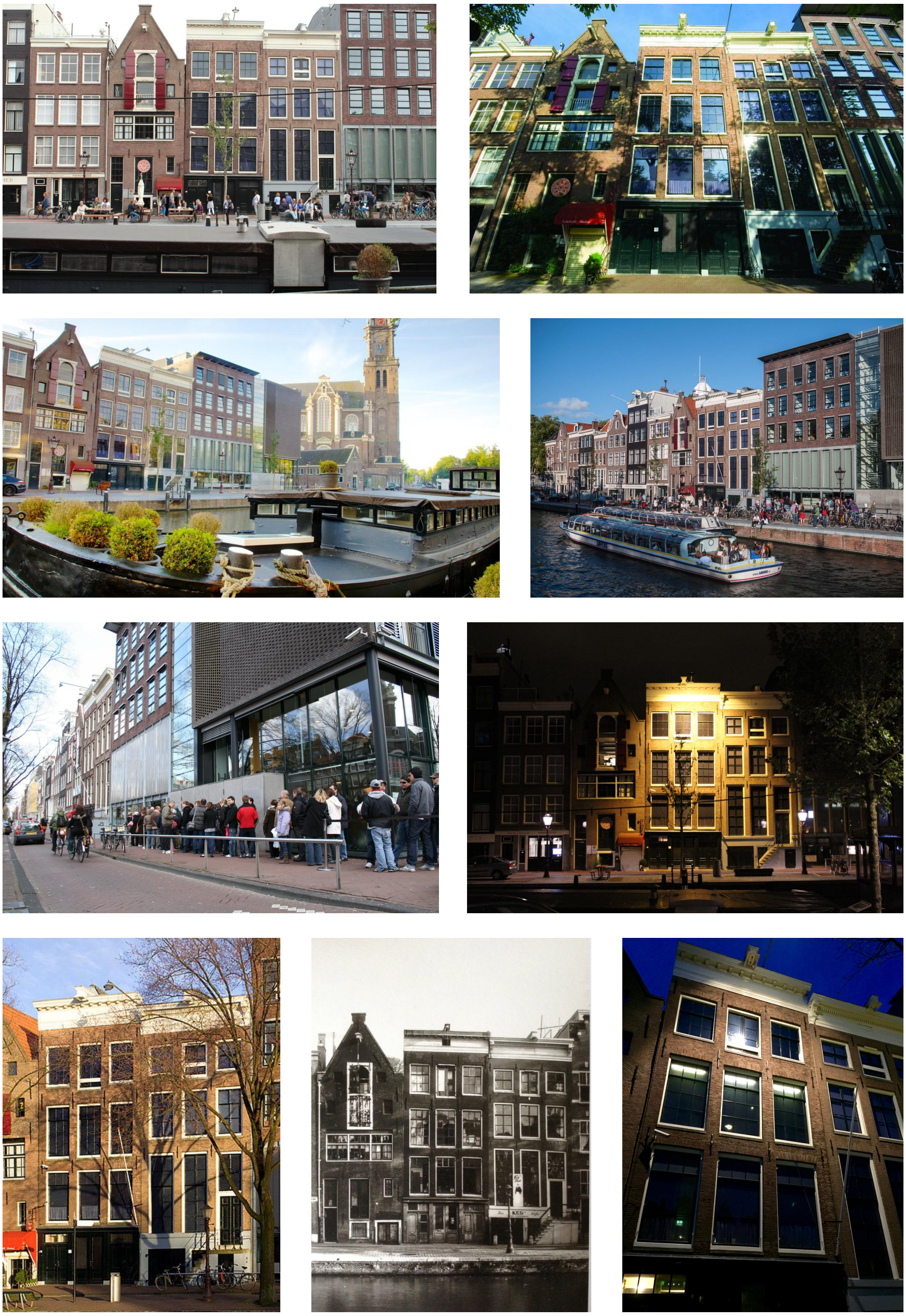
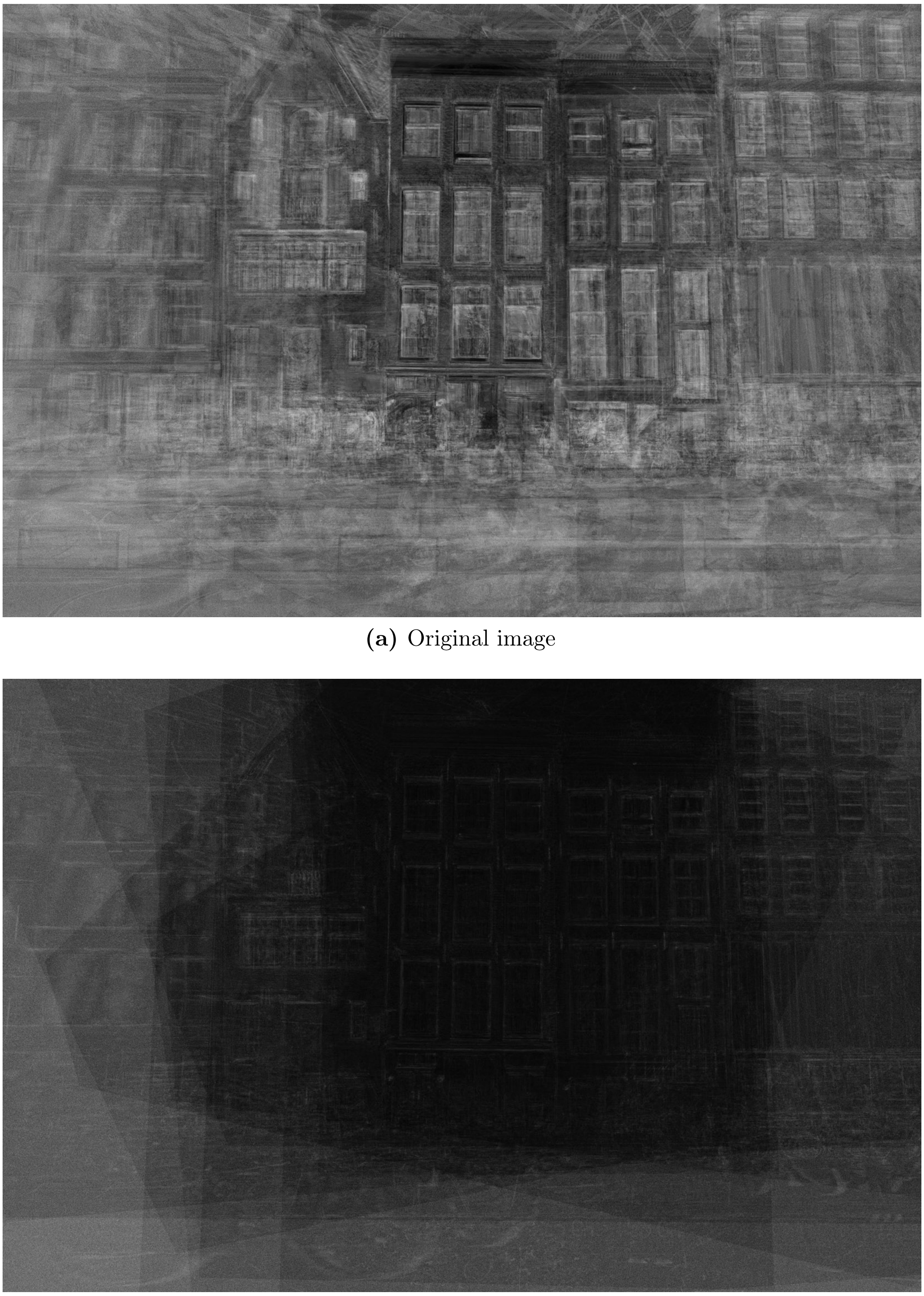


Figure 6.1: Sample images from the Anne Frank House dataset.



(a) Original image

(b) Self Quotient Image

Figure 6.2: Coefficient of dispersion of all images on the Anne Frank House dataset, manually aligned and projected on top of each other. The coefficient of dispersion is calculated as the ratio of the interquartile range to the mean at each pixel.

6.2 Sieve

The Sieves were first proposed as morphological extensions of one-dimensional recursive median filters [18]. These filters were then extended to two dimensions using graph morphology, and since then they have been applied to a range of problems in computer vision including lip-reading [95], stereo vision [100] and image retrieval [50]. A brief explanation of the Sieve algorithm can be found in Appendix B. A more detailed mathematical description of the Sieve, its theorems and properties are available in the literature [15, 16, 49, 58].

In a nutshell, the Sieve is a non-linear decomposition algorithm that identifies intensity extrema by the scale and removes them by slicing off the peaks, up or down, to the next most extreme level. It uses morphological operations to eliminate increasing scale detail keeping the signal's basic structure. Figure 6.3 and Figure 6.4 show an image decomposition using the Sieve algorithm for greyscale and SQI images, respectively.

Selecting the most suitable operator is crucial when using the Sieve for image segmentation. Using $\mathcal{M}$- and $\mathcal{N}$-filters, the sliced regions precisely align with shadow or illumination highlights, i.e. the granules naturally follow the isophotes.

6.3 Overlap comparison

Our objective in conducting the following experiments is to measure the accuracy and repeatability of detectors and to determine to what extent the detected regions overlap in the same scene area.

First, we convert all images to grayscale by transforming them to the CIELAB colour space and retaining only the L dimension. Based on the

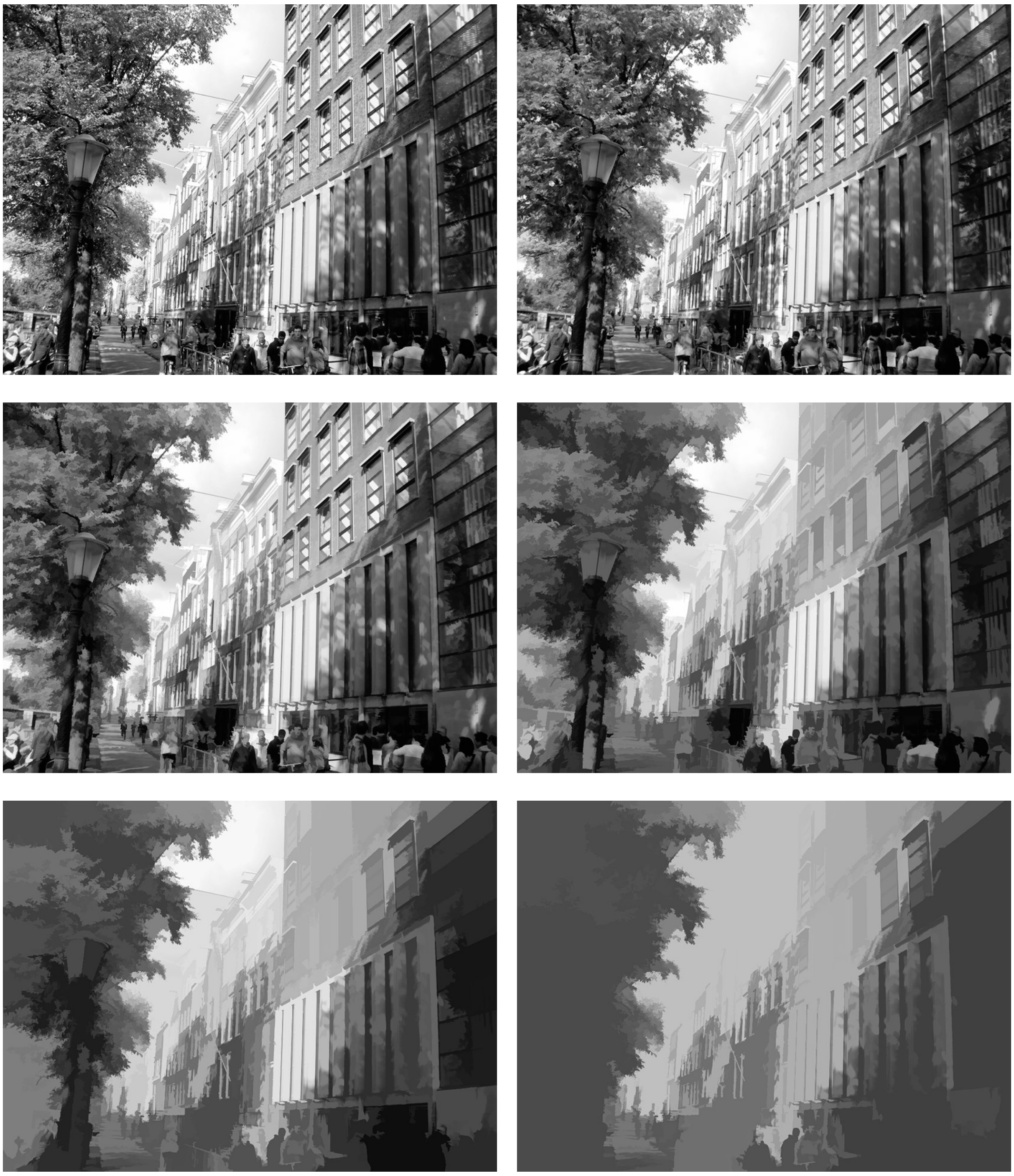


Figure 6.3: Original grey scale image at 1024 × 768 pixels (top left) sieved with an $\mathcal{M}$-filter at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

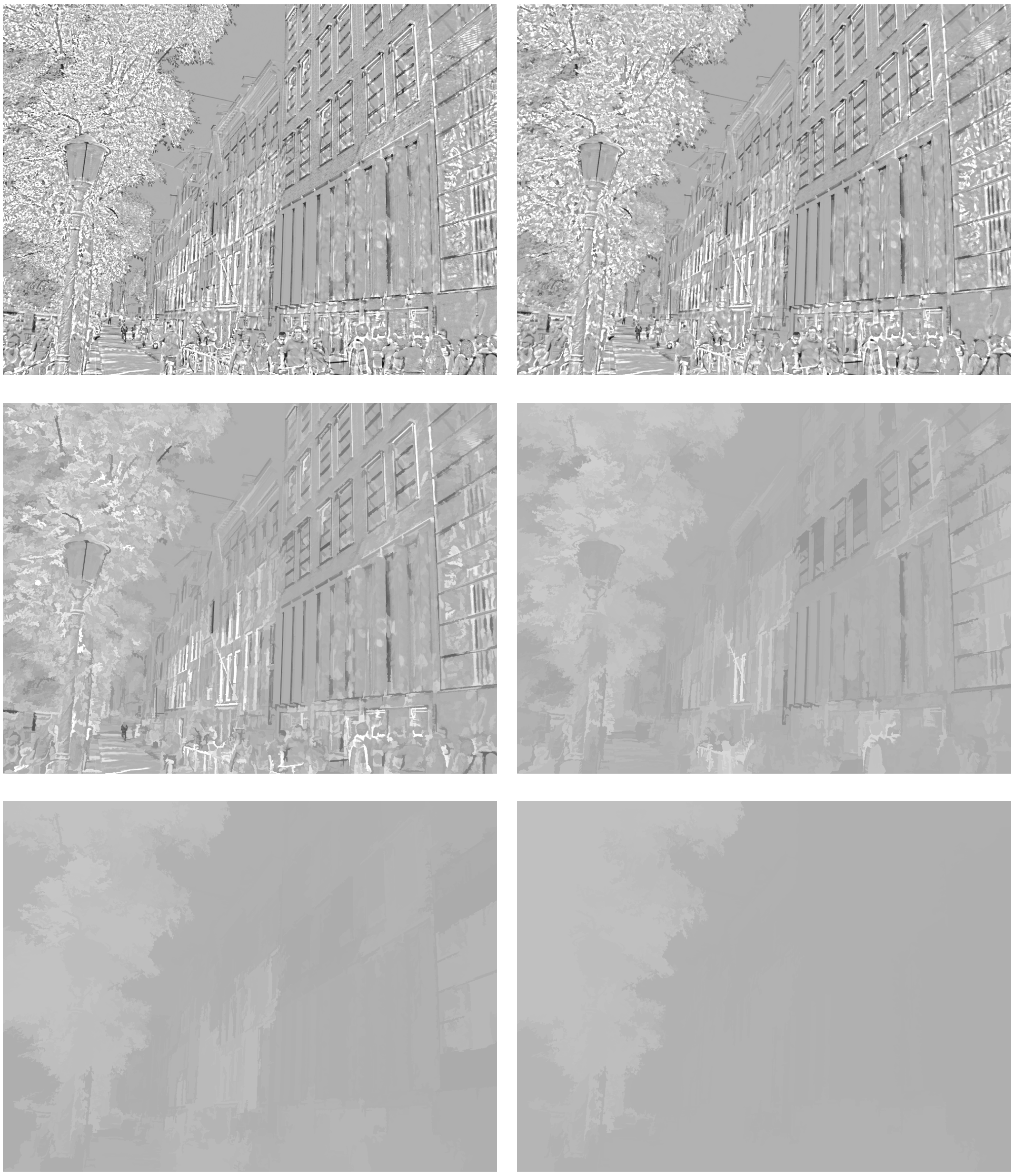


Figure 6.4: Self Quotient Image of the picture in Figure 6.3 sieved with an $\mathcal{M}$-filter to scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

experimental results reported in Chapter 5, we chose to detect SIFT features with affine shape estimation on each pair of images. MSER and SIFT without affine estimation features are also tested for comparison.

To assess the accuracy and reliability of detectors, we establish an overlap error threshold and normalise the sizes of the detected regions. We then test the repeatability of SIFT features by gradually increasing the mesh size of the Sieve algorithm to measure how the number of corresponding regions changes with scale. We record both the actual and relative numbers of corresponding regions. A robust detector should present a large number of correspondences and a high repeatability score.

The ground truth was calculated by hand-mapping four points precisely at the corners of the Anne Frank house in all pictures contained in the dataset. It allowed us to then compute the projective transformation matrices [90] for each pair of images. To evaluate the performance of detectors, we calculate the overlap area between the detected region in a reference image and its corresponding region in the query image. We then use the projective transformation H to project the overlap onto the reference image.

The correspondence of two regions may be measured by the overlap error [98], defined as

$$\epsilon_o = 1 - \frac{R_{\mu_a} \cap R_{H^T \mu_b H}}{R_{\mu_a} \cup R_{H^T \mu_b H}} \tag{6.1}$$

where R_μ gives the elliptical region defined by $x^T \mu x = 1$.

A closer look at the definition of ϵ_o reveals that larger regions are more likely to obtain favourable overlap scores [98]. The overlap performance of a region detector can be increased by merely doubling the size of all regions. To promote a fair comparison, we normalise the regions R_μ fixing their size by applying an arbitrary scaling factor s. This scaling is performed only to calculate ϵ_o, such normalisation is not performed or even desired at other steps.

We normalise the region size by applying a scaling factor s that resizes the regions to a fixed area $s \cdot R_{\mu_a}$. This scaling factor s is also applied to $R_{H^T \mu_b H}$, which is the region detected in the other image mapped onto the reference image. We use these normalised regions to calculate the actual overlap error as described earlier.

To determine the repeatability score for a specific pair of images, we divide the number of region-to-region correspondences by the smaller number of regions detected in the pair of images. In our experiments, we consider only the regions in common for each pair of images.

In the first set of experiments, we measure how the number of correspondences varies with the Sieve mesh size. The overlap error threshold ϵ_o is set to be less than 40 %, and the normalised region size is set to have a radius of 30 px. In general, a detector with a higher repeatability score and a greater number of correspondences is considered to be more robust. This test provides a means of measuring the robustness of the detectors at each Sieve band.

In practical applications, the matching or clustering of regions is based not only on the accuracy and repeatability of the detection but also on the distinctiveness of the detected regions. To assess the effectiveness of region matching, we examine both the number of matches found and the ratio of correct matches to mismatches.

The matching score between two images is calculated in two steps:

1. Two regions are considered correspondent if the overlap error $\epsilon_o \leq 40\,\%$. Only correspondent regions are considered for the next step.

2. To count a match, we check if the first nearest neighbour in the descriptor space retrieves the corresponding area when compared by the Euclidean distance to all other features in that specific image. The matching score is calculated by dividing the number of matches by the smaller number of detected regions in a pair of images.

We also measure the impact of the Self Quotient Image algorithm on the detection of local features. The process described above is also applied to the sieved Self Quotient Images at the same scales and parameters used on the original images.

6.4 Experimental results

For these experiments, we consider both SIFT [89] and MSER [94] features with affine shape estimation [101] and original SIFT without affine estimation for all detected regions, and check which regions correctly match with each detector. To compensate for the affine geometric deformations, we map each elliptical region to a circular region of 30×30 pixels, rotating it based on the estimated gradient orientation.

The results of these experimental tests are shown in Figure 6.5 to Figure 6.12. The six first figures are composed of four plots: (a) repeatability score, (b) number of correspondences, (c) matching score and (d) number of correct matches. The first row of plots, i.e. subfigures (a) and (b), refers to the performance of feature detection. The second row considers the matching of features when the first nearest neighbour is retrieved. The matching score (c) indicates the proportion of features that match, while the number of correct matches is shown in (d). These two plots evaluate the efficiency of feature descriptors, i.e. if the description of a point of interest matches with, and only with, its correspondent. Notice that the matching score is always lower than the repeatability. This happens because the correct correspondence of a pair of features is a prerequisite to consider a correct match. Similarly, the number of correct matches could never be greater than the number of correspondences.

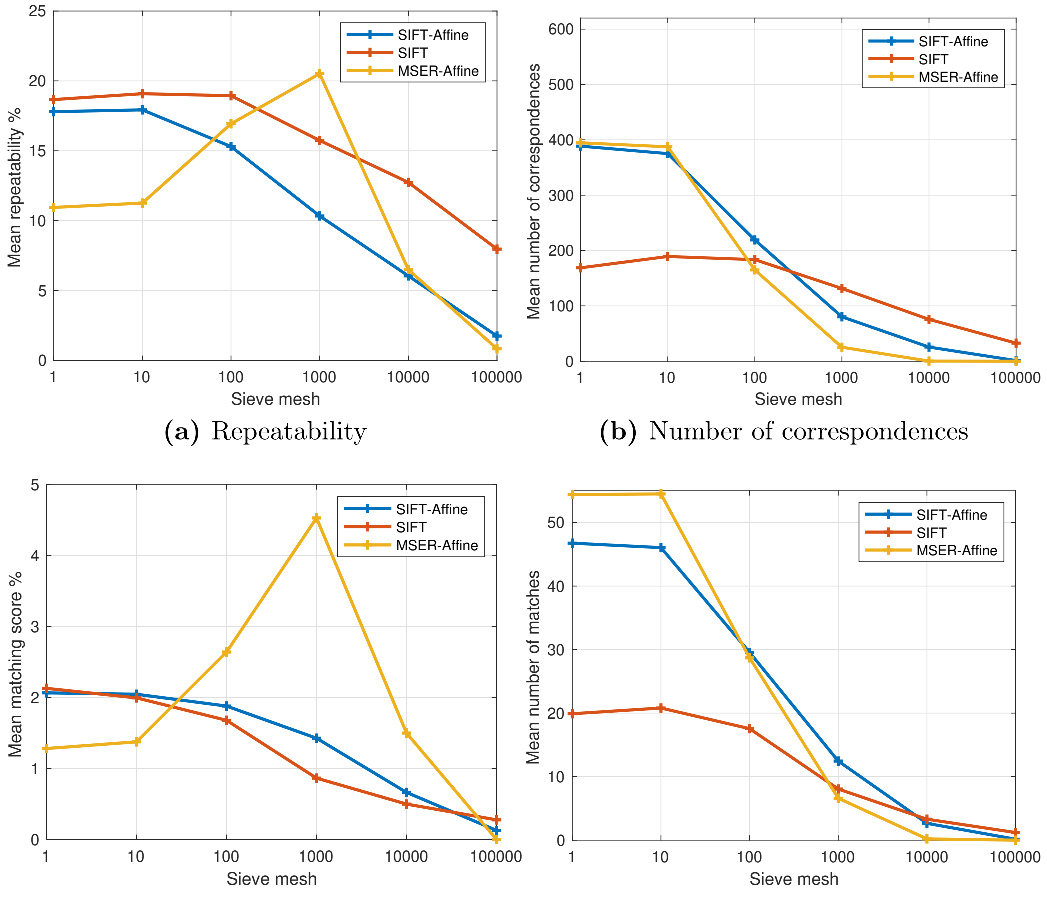
We analyse the performance of SIFT and MSER features with an estimation of affine shapes, as well as original SIFT features with no affine estimation. Figure 6.5 and Figure 6.6 are relative to a low-pass Sieve segmentation of original images and Self Quotient Images, respectively. The following figures show the results of high-pass (Figure 6.7 and Figure 6.8) and band-pass sieving (Figure 6.9 and Figure 6.10) for original images and Self Quotient Images. All results are presented as the mean performance of matching images on the Anne Frank House dataset with a reference image.

For low-pass Sieve (Figure 6.5), Sieve mesh = 1 represents the original image with no Sieve processing. The repeatability score obtained at this Sieve mesh = 1 indicates the performance of each detector on the original photos and the degree to which the detector is influenced by changes in perspective, illuminance, and other factors. Plot (a) shows that the repeatability scores of original images are similar when we use SIFT with or without affine shape estimation. The repeatability score of MSER is 39 % lower. On the other hand, the number of MSER correspondences is almost identical to the number of SIFT-affine correspondences. Therefore, more MSER points have no correspondence on the reference image.

Regarding the number of matches (d), SIFT and MSER reach 50 matches when we use affine shape adaptation (Figure 6.5(d), Sieve mesh = 1). The performance of SIFT with no projection estimation drops 60 % relative to the number of correct matches. The matching score (c), on the other hand, is 61 % higher when we use SIFT, with or without affine estimation.

The slope of the curves reflects how much a detector is affected by progressively removing details with a low-pass Sieve. As we start to remove detail, at mesh = 10, the performance of all four indicators stays the same or even improves slightly. This is due to the removal of fine noise in the images. We

Low-pass Sieve, original images



(a) Repeatability

(b) Number of correspondences

(c) Matching score

(d) Number of correct matches

Figure 6.5: Low-pass Sieve segmentation for the structured scene (Anne Frank house dataset, Section 6.1). (a) Repeatability score for increasing Sieve mesh size (default settings: overlap error ≤ 40 %, normalised size = 30 px). (b) Number of corresponding regions. (c) Matching score. (d) Number of correct nearest neighbour matches.

observe a substantial decrease in performance for Sieve meshes greater than 100. The number of matches is negligible for Sieve meshes equal to or greater than 10,000.

The repeatability curve (a) for MSER does not follow the tendency of other algorithms. Both repeatability and matching scores increase until a Sieve mesh = 1000. This is explained by the fact that MSER detects areas and SIFT detects corners and edges. Increasing Sieve meshes progressively remove detail, and areas of interest become more evident, which helps MSER. Nonetheless, the absolute number of areas detected (b) drops 62 % at Sieve mesh = 100. Therefore, increasing mesh sizes effectively eliminates incorrect MSER correspondences at a higher rate than it eliminates correct ones.

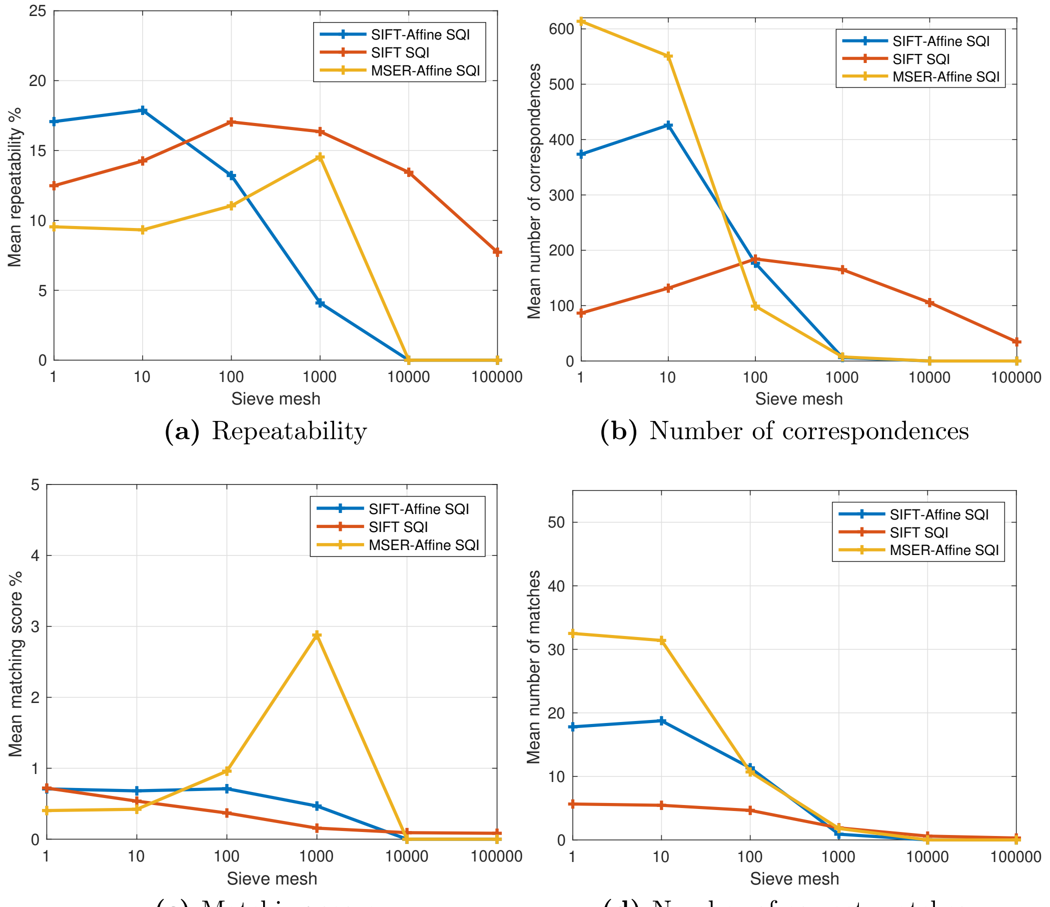
The results for the matching plots (c) and (d) follow the same tendency observed in the corresponding regions (a) and (b), although the former has a performance 87 % lower than the latter. This means that the lack of distinctiveness in the regions detected results in a high number of mismatches. In practice, more complex methods than considering just the first nearest neighbour are used to match region descriptors. Therefore, these results must be considered with caution.

Results are quite different when we apply a low-pass Sieve to Self Quotient Images (Figure 6.6). SIFT-affine is negatively affected when we consider Self Quotient Images. SIFT with no affine estimation has a more stable curve, although its performance also decreases.

MSER benefits from using SQI, as the number of correspondences (b) increases by 52 % when we consider original images (Sieve mesh = 1) and by 41 % with a Sieve mesh = 10. SIFT-affine also has 10 % more correspondences when we apply Sieve with a mesh size = 10.

These results help to understand what happened with the experiments reported in Chapter 5. Self Quotient Images do increase the number of correspondences. Nonetheless, the description of features detected on SQI is not distinctive enough to allow matching correct pairs of corresponding features. The fact is that the SIFT descriptor does not work well with SQI. This is due to the low contrast of SQI, which wipes away most of the information used to

Low-pass Sieve, Self Quotient Images



(a) Repeatability **(b)** Number of correspondences

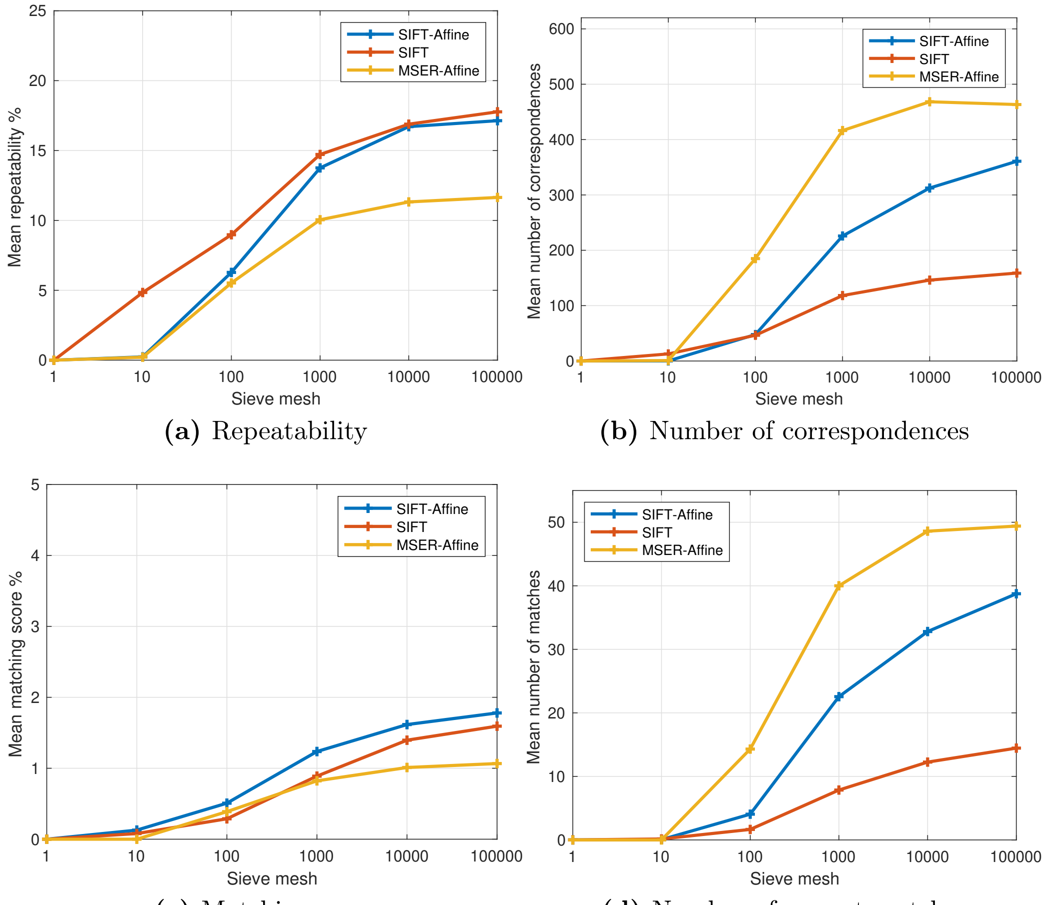
(c) Matching score **(d)** Number of correct matches

Figure 6.6: Low-pass Sieve segmentation for the Self Quotient Images (SQI) of the structured scene (Anne Frank house dataset, Section 6.1). (a) Repeatability score for increasing Sieve mesh size (default settings). (b) Number of corresponding regions. (c) Matching score. (d) Number of correct nearest neighbour matches.

describe points of interest.

Unlike the low-pass Sieve, a high-pass Sieve (Figure 6.7 and Figure 6.8) keeps detail instead of removing it. At Sieve mesh = 1, we just have a mid-grey image for all algorithms. At Sieve mesh = 10, only fine details are kept and everything else is removed. Larger meshes incorporate coarser details until Sieve mesh = 100,000, which is very similar to the original image.

High-pass Sieve, original images



(a) Repeatability

(b) Number of correspondences

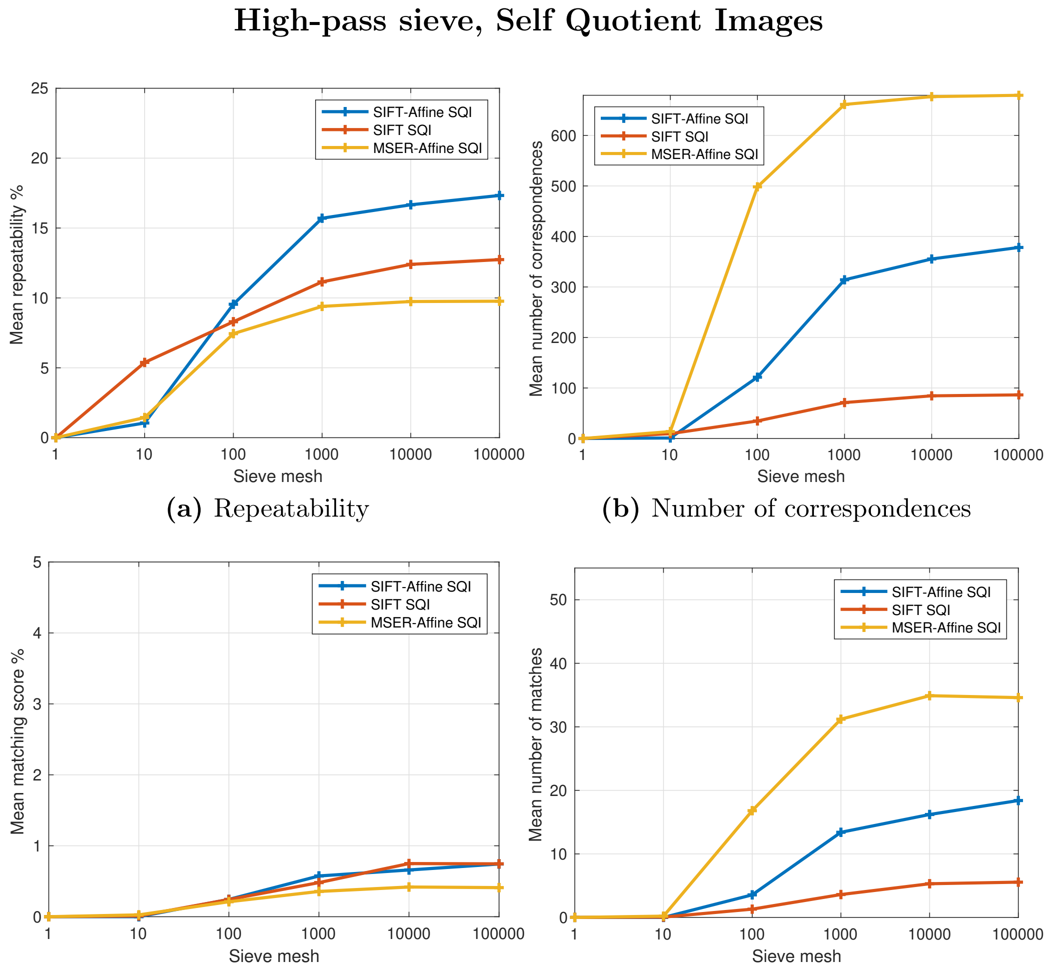
(c) Matching score

(d) Number of correct matches

Figure 6.7: High-pass Sieve segmentation for the structured scene (Anne Frank house dataset, Section 6.1). (a) Repeatability score for increasing Sieve mesh size (default settings). (b) Number of corresponding regions. (c) Matching score. (d) Number of correct nearest neighbour matches.

The value of each pixel processed with a high-pass Sieve can range from −255 to +255, although most images have an effective range of 255 values at all mesh sizes. For this reason, we need to adjust the contrast of images processed with a high-pass Sieve so pixel values lie between 0 and +255. The same happens with the band-pass that we analyse later.

There is no significant increase in performance when we work just with



(a) Repeatability

(b) Number of correspondences

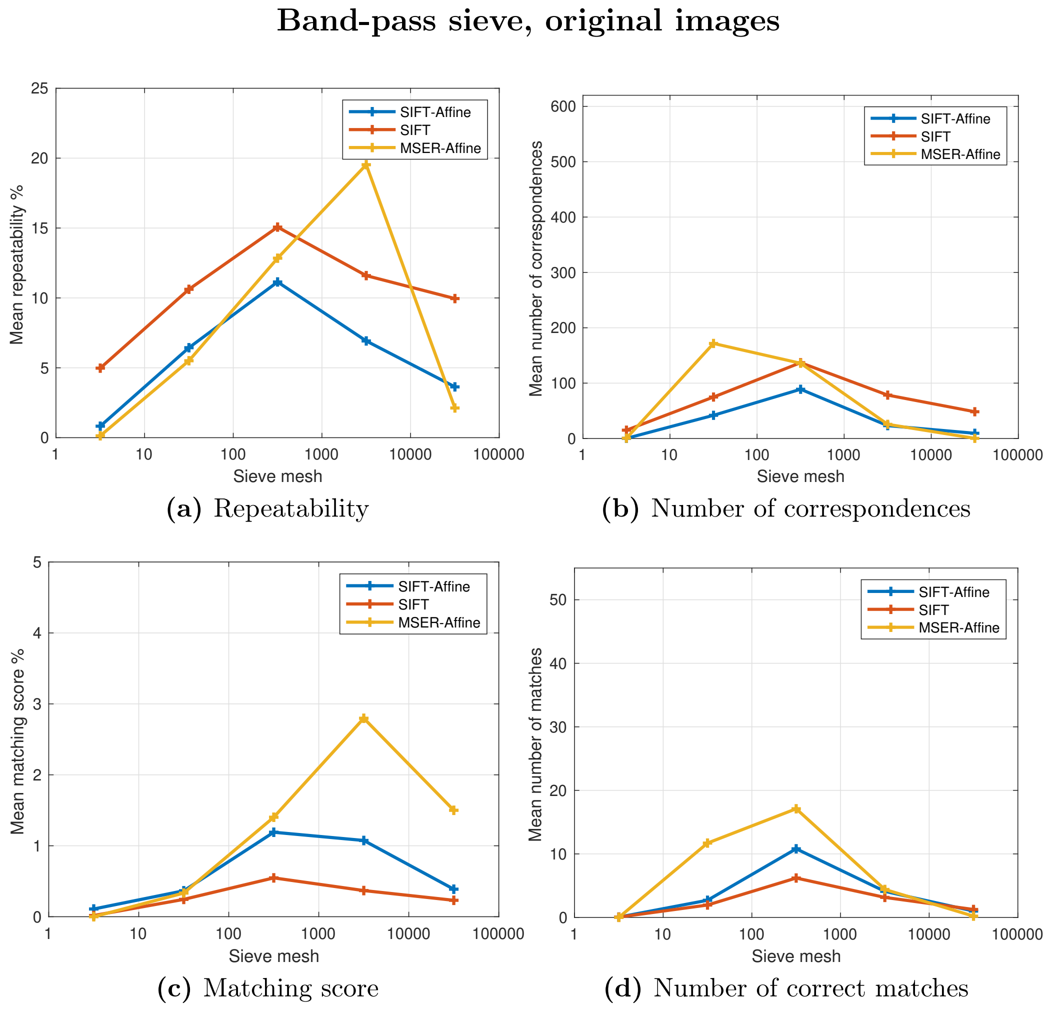
(c) Matching score

(d) Number of correct matches

Figure 6.8: High-pass Sieve segmentation for the Self Quotient Images (SQI) of the structured scene (Anne Frank house dataset, Section 6.1). (a) Repeatability score for increasing Sieve mesh size (default settings). (b) Number of corresponding regions. (c) Matching score. (d) Number of correct nearest neighbour matches.

the details given by the high-pass Sieve. Figure 6.8 shows that the process of detecting features is once more helped by using Self Quotient Images, although the high-pass processing itself does not increase performance.

For the band-pass Sieve, Figure 6.9 and Figure 6.10, we represent the Sieve bands by positioning the values of the graph curves between the ticks on the x-axis. A value between 10 and 100, for example, indicates that is the result for



(a) Repeatability

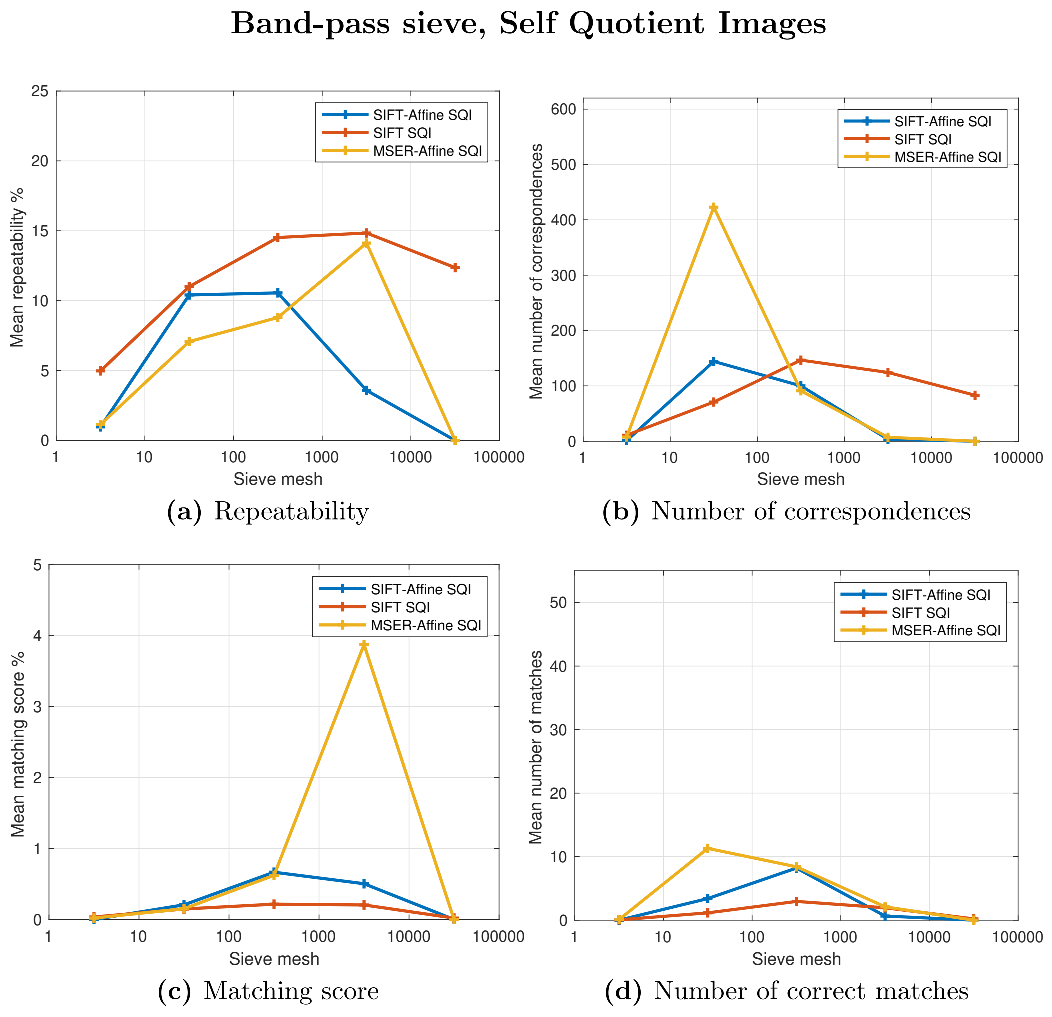
(b) Number of correspondences

(c) Matching score

(d) Number of correct matches

Figure 6.9: Band-pass Sieve segmentation for the structured scene (Anne Frank house dataset, Section 6.1). (a) Repeatability score for increasing Sieve mesh size (default settings). (b) Number of corresponding regions. (c) Matching score. (d) Number of correct nearest neighbour matches.

the band with granules between scales 10 and 100. These results indicate what size of details holds more useful information for the detection and matching of features. Mid-size information retrieved with band-pass 100 to 1000 holds the majority of detected and matched features with SIFT algorithms. The number of MSER correspondences (b) is 20 % higher at Sieve band 10 to 100 when compared to the band 100 to 1000.



(a) Repeatability

(b) Number of correspondences

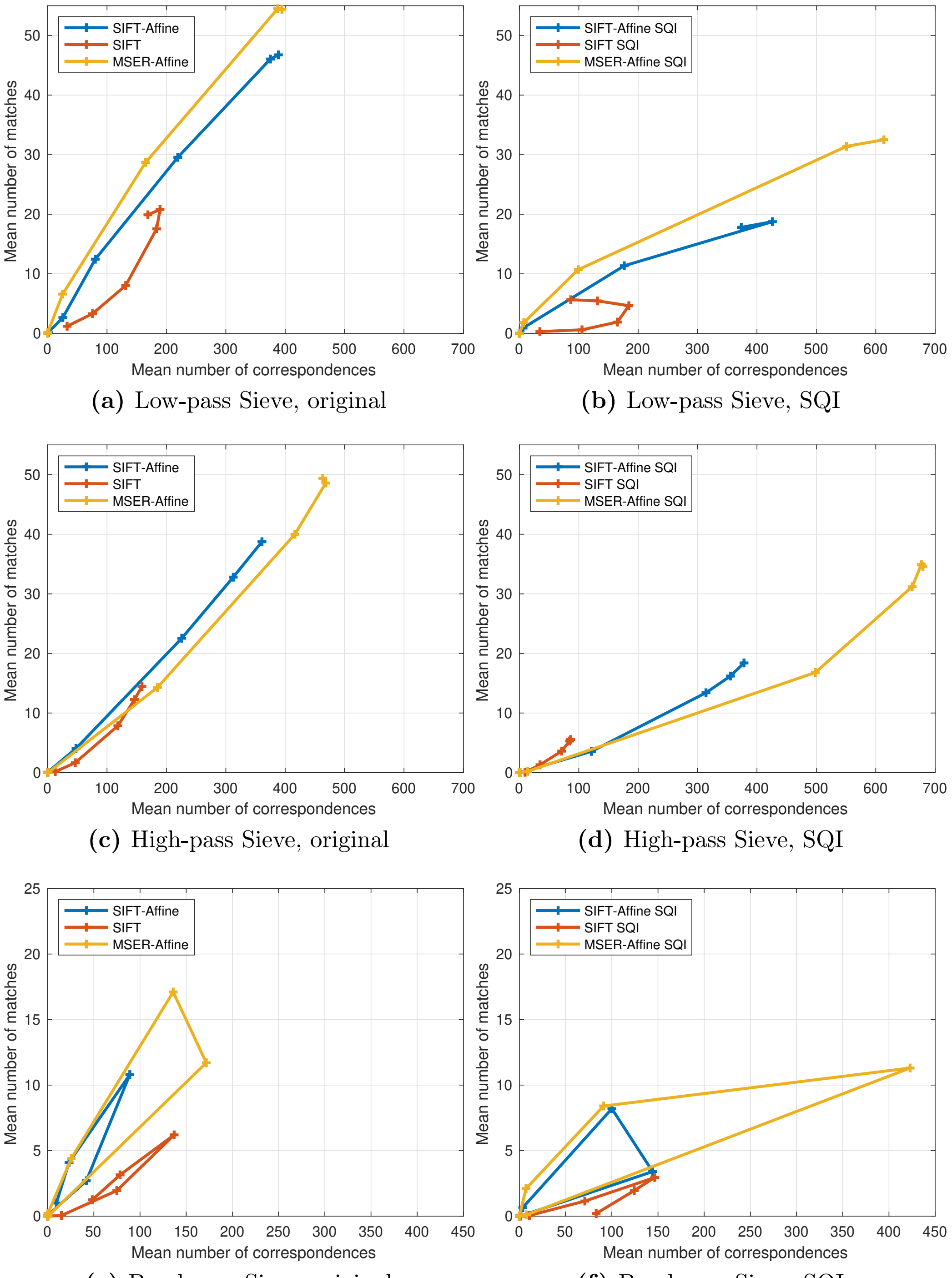
(c) Matching score

(d) Number of correct matches

Figure 6.10: Band-pass Sieve segmentation for the Self Quotient Images (SQI) of the structured scene (Anne Frank house dataset, Section 6.1). (a) Repeatability score for increasing Sieve mesh size (default settings). (b) Number of corresponding regions. (c) Matching score. (d) Number of correct nearest neighbour matches.

On Self Quotient Images, Figure 6.10(b), band-pass sieving reveals that most of the information used to detect features is between Sieve meshes 10 and 100, especially when MSER features are considered.

Figure 6.11 presents plots with the number of correspondences vs the number of matches. These plots help us to compare different image treatments more directly, letting the variation of Sieve meshes implicit on the curves. In all these plots, the closer to the top right corner the better. A greater number of intermediate Sieve meshes would create smoother curves. On plot (a), SIFT-affine and MSER both have approximately 400 correspondences (x-axis), yet the number of MSER matches surpasses by 22 % the number of SIFT-affine matches. All three algorithms in (a) show an optimal Sieve scale, and all three passes have optimal scales as well. The high-pass Sieve displays monotonic lines, which is different from the other passes. Looking at these plots, it becomes clear that the use of Self Quotient Images indeed increases the number of correspondences and decreases the number of matches.



(a) Low-pass Sieve, original

(b) Low-pass Sieve, SQI

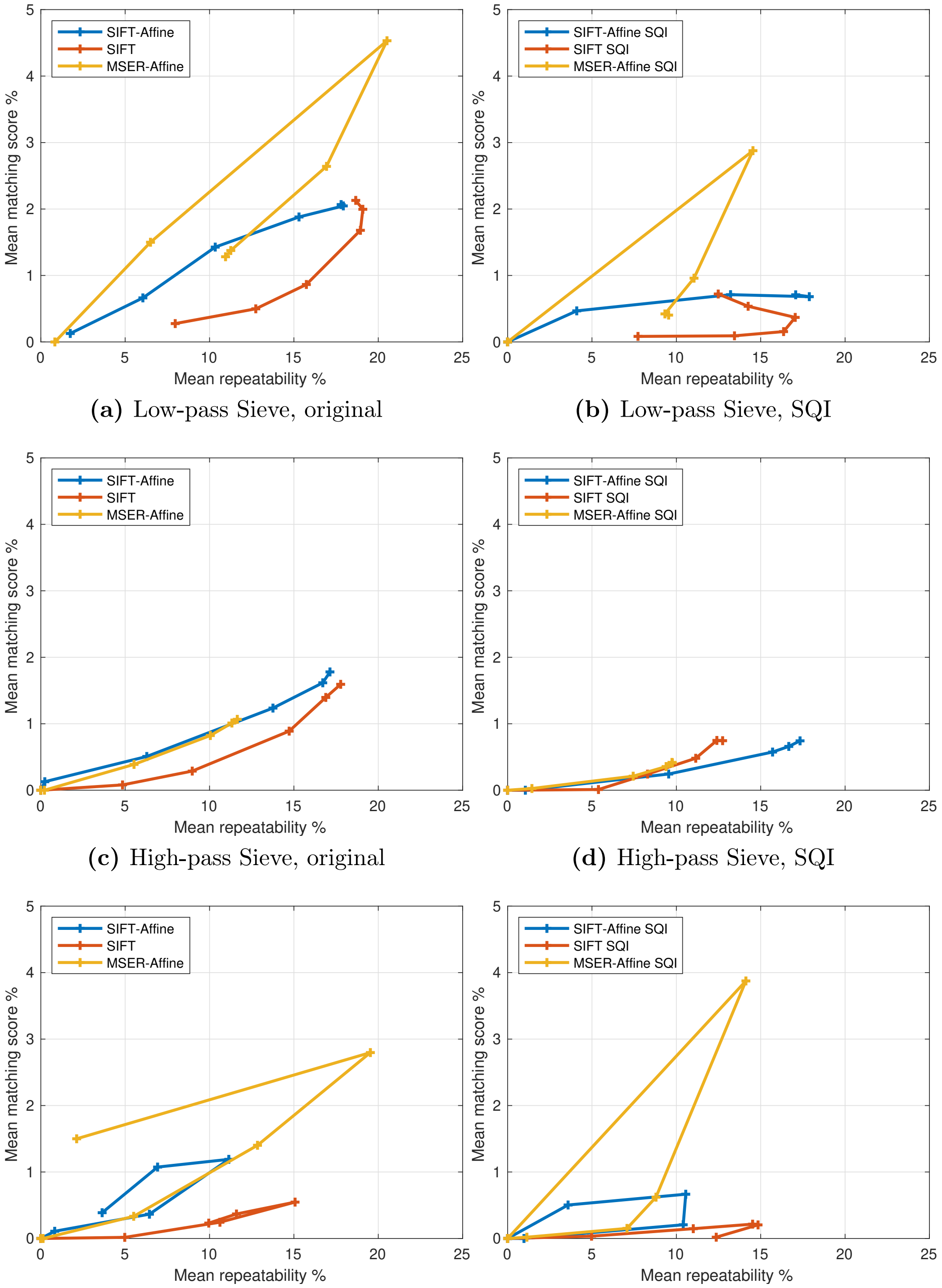
(c) High-pass Sieve, original

(d) High-pass Sieve, SQI

(e) Band-pass Sieve, original

(f) Band-pass Sieve, SQI

Figure 6.11: Number of correspondences versus number of correct matches considering Sieved original images and Self Quotient Images of the Anne Frank house dataset.



(a) Low-pass Sieve, original

(b) Low-pass Sieve, SQI

(c) High-pass Sieve, original

(d) High-pass Sieve, SQI

(e) Band-pass Sieve, original

(f) Band-pass Sieve, SQI

Figure 6.12: Repeatability versus matching score considering Sieved original images and Self Quotient Images of the Anne Frank house dataset.

6.5 Discussion

All plots presented in the previous section show two action points. One is around Sieve mesh = 20, below which we can remove detail without affecting the algorithms. The other is around Sieve mesh = 5000, above which scenes become unrecognisable and the number of features is low.

On average, the MSER detector achieves the best results. The number of corresponding areas using SIFT-affine and MSER at Sieve mesh = 10 are similar to those detected on the original image (mesh size = 1). The highest mean repeatability score of 21 % is given by MSER with a Sieve mesh = 1000. Flat areas produced by the Sieve have a positive effect on the detection of MSER blobs, producing a few areas with a high repeatability score.

MSER does not follow the curves produced by SIFT because it detects blobs of interest rather than points and edges. This method benefits from using the Sieve, which makes areas more evident. SIFT is an averaging algorithm that uses multi-scale windows. It has some tolerance to change in illuminance by design. Nonetheless, it is not robust enough to deal with changes in illuminance observed on real-life street-view images.

The power of MSER becomes evident when we use Self Quotient Images. Although the number of matches drops compared to original images (Figure 6.5 and Figure 6.6), MSER still outperforms SIFT-affine.

Self Quotient Images do not improve the number of correct matches as we expected. Illuminance variation is a major concern for accomplishing this task of matching features, therefore normalising the effects of changes in illuminance should increase the performance of all algorithms that detect and describe features. What we observe instead is that the Self Quotient Images algorithm changes the appearance of images dramatically, as can be seen in Figure 6.3. It is a true challenge to guess even in which country those pictures were taken. There are hardly any matches at Sieve meshes above 1000. Such images become just a composition of abstract grey blobs. Yet, MSER is successfully selecting large regions without losing precision.

In Chapter 5, we observe that Self Quotient Images make GMCP–based image geolocation considerably worse. This geolocation method was designed to use SIFT-affine features. Results presented in Figure 6.5 show that SIFT-affine features reach a correct match ratio of 2 % and 46 correct matches. When we use SQI (Figure 6.6), the performance is less than half of that. Results presented in this chapter clarify that the use of Self Quotient Images successfully improves the detection of features, although matching those features becomes more challenging.

MSER stands out as a better detection algorithm for this problem. When combined with the Sieve, the matching score can increase to 4.5 % at Sieve mesh = 1000. There are just a few good matches at this scale, but as few as seven correspondences are enough to estimate a fundamental matrix (see Section 5.2 for more details on the process of estimating the fundamental matrix).

Using a Sieve to remove a small amount of low-scale artefacts helps to detect and match features. The question of what Sieve scale to use remains for other images not tested during this experiment.

Limitations and future work

In this set of experiments, we focus on the detection and matching of local features at distinct levels of detail using a dataset of twenty-two images. This technique is useful to have insight into such a dataset, but it would not be appropriate to draw more general conclusions without a larger study on a more comprehensive dataset of street-view images. Although such a larger study would take a long time to be conducted, the proposed method can be applied to a larger dataset with little adaptation.

We demonstrated that MSER features are robust for matching street-view images, but further investigation is needed to fully define the optimal settings for use with the Sieve and Self Quotient Images on a real application. The result of processing an image with the Sieve depends on the resolution of the input image. The same occurs with Self Quotient Images. Therefore, the parameters for both algorithms must be adjusted for the input image resolution.

6.6 Conclusion

Image matching over extended periods is extremely challenging, and methods that are routinely used in SLAM or local matching fail catastrophically when applied to images like the ones illustrated in Figure 6.1. But what is the cause of the failure? To help quantify this problem we have devised a new dataset and a scale-based analysis that allows us to identify the nature of the problem and hence develop alternative local feature extraction and matching methods.

We tested the performance of SIFT and MSER, with and without affine shape estimation, on the new Anne Frank House dataset. A graph-morphological Sieve algorithm was used to observe the behaviour of algorithms at different levels of detail. MSER combined with SQI and Sieve proved to be robust for matching street-view images, increasing by 90 % the matching score in some cases when compared to SIFT features extracted from original images.

We observe that illuminant normalisation also needs to be done. SQI might not be the ideal one for this problem, but some sort of illuminant normalisation has to be done because illuminant variation is a recurrent problem and can be quite dramatic in real-life situations. The effects of illuminant variation destroy conventional feature description algorithms. The SIFT algorithm considered in this study is frequently used in academic projects and industry; it is reasonable to suppose that similar algorithms would also fail. However, if we move to another feature detection method, which MSER seems the most promising, and combine that with a method of keeping only mid-scale visual information, we then have a new framework for matching street-view images under a wide range of conditions.

Chapter 7

Conclusion

7.1 Methodology overview

The methodology employed in this thesis aimed to answer the research question: Can we use computer vision techniques to improve the accuracy of geolocation estimation to potentially assist persons with visual impairment navigating outdoors? To address this question, we first reviewed the state-of-the-art navigation assistants for the visually impaired (NAVI) used for outdoor navigation, and conducted a literature review on image processing techniques for geolocation. This review reported in Chapter 2 provided an understanding of the NAVI systems currently available, the equipment most commonly used, and the key image processing methods used for single image geolocation.

Next, we analysed the desired requisites of a vision-based navigation assistant (Chapter 3). In addition to the literature review, a semistructured focus group session was organised with three individuals who are blind to explore the participants' challenges when navigating outdoors. This integrated approach ensured that the identified requirements were based on the challenges experienced by individuals, as well as on the state-of-the-art image processing methods and equipment for outdoor navigation. Among the technical gaps identified, the need for high-accuracy user localisation was highlighted as a critical requirement for safe navigation in urban environments.

We then explored vision-based methods to geolocate the user in urban areas due to their potential to achieve sub-meter accuracy. We considered two scenarios: when the GNSS service is available and when it is not.

For urban areas with GNSS coverage, we presented HLoc+SV, a method for refining GNSS geolocation through the use of geotagged street-view images (Chapter 4). This method builds upon the hierarchical localisation (HLoc) framework to accurately estimate the world coordinates of a user's camera by integrating geotagged street-view images into the SfM model.

The HLoc+SV method presents several novel aspects. Firstly, we progressively add geotagged street view images to the SfM model and use them as 'anchors' to geolocate the entire model. Secondly, we use the user's GNSS geolocation to dynamically download street view images near the user and build the SfM model on-the-fly. Additionally, to overcome the challenge of evaluating results without a high-accuracy GNSS receiver, we estimated ground truth by identifying clear marks on the ground visible on satellite images and manually retrieving their geocoordinates. For this study, we created a dataset with 58 user images and 80 street view images, covering seven scenes in Norwich and London. Each user image includes GNSS coordinates retrieved by the smartphone and also their ground truth for evaluation purposes. Although the results added evidence to the argument that vision-based solutions can improve the accuracy of geolocation estimation, the HLoc+SV is only suitable for the scenario when the GNSS service is available and relatively accurate.

For areas where the GNSS service is unreliable or unavailable, we studied the problem of geolocating a single image with no previous geolocation information.

We reviewed the main approaches in the literature (Section 2.4), but the best results we could find have geolocation errors of the order of hundreds of metres and are not suitable for safely building a navigation assistant with the requirements described in Chapter 2. Furthermore, most proposed solutions in this area do not discuss the role and impact of different local descriptors on the geolocation task.

For this task, we proposed a geolocation method based on a Generalised Minimum Clique Problem (GMCP) image retrieval framework, and attempted to further improve it by normalising the illuminance of images with the Self Quotient Image algorithm (Chapter 5). To understand this problem, we visually checked the impact of SQI light normalisation on the detection of local features (Section 5.3), analysed a single scene with progressive change in the camera's perspective and illumination conditions (Section 5.5.1), and compared the performance of SIFT, SURF and MSER features on matching street-view images with simultaneous changes in perspective, illumination and occlusion on a dataset of 400 street-view images taken years apart (Section 5.5.2). We tested our geolocation estimation method on the UCF Street View dataset, which contains more than 62,000 Google Street View images of three cities in the US. The results obtained in this experiment indicated that illumination plays a crucial role in the performance of feature detection on street-view images, especially when changes in illumination are combined with a perspective shift.

Finally, those results led us to investigate the impact of SQI on the accuracy and repeatability of local feature detectors and descriptors (Chapter 6) by progressively removing details in images using a graph-morphological algorithm known as Sieve. For this study, we built a new structured image dataset with twenty-two photos of the Anne Frank house in Amsterdam, and analysed their relative variability by hand aligning all images and projecting them all onto a canonical image. We then tested the performance of SIFT and MSER, with and without affine shape estimation, on this dataset. The results showed that MSER combined with SQI and Sieve were robust for matching street-view images, increasing by 90 % the matching score in some cases when compared to SIFT features extracted from original images.

In conclusion, the methodology adopted in this research focused on exploring the use of vision-based methods for geolocation, which can potentially be employed in outdoor navigation assistants. We proposed novel vision-based approaches, such as the HLoc+SV and the GMCP-based image retrieval method with SQI illuminance normalisation techniques, which ultimately contribute to the vision-based geolocation research field.

7.2 Summary

In this thesis, we addressed the research question: Can we use computer vision techniques to improve the accuracy of geolocation estimation to potentially assist persons with visual impairment navigating outdoors? The answer is *yes, we can.* We detailed the construction of the HLoc+SV, a vision-based geolocation method based on a version of the hierarchical localisation framework that exploits information from a dataset of geotagged street-view images. In a dataset of 58 user pictures and 80 geotagged street-view reference images, HLoc+SV had a mean absolute geolocation error of 0.77 m (SD 0.41), while a smartphone GNSS receptor had a 12.09 m (SD 8.67) error. Extremely challenging scenarios for GNSS presented the greatest improvement in performance, even when night-time pictures were considered. When the GNSS mean absolute geoposition error in a scene is as high as 30.21 m (SD 1.65), the error with HLoc+SV is 1.17 m (SD 0.07). Nonetheless, the HLoc+SV is a potential solution suitable only for the scenario when the GNSS service is available and relatively accurate.

We also explored solutions for the scenario when the GNSS service is unreliable or unavailable. We proposed and analysed a framework to geolocate an image using a GMCP-based image matching method combined with the SQI illuminance normalisation. Our method first normalises the illuminance of images and then retrieves and matches the local features from a reference set. We found out that there is no performance improvement with the use of the SQI illumination normalisation with GMCP–based image geolocation framework. Moreover, there is a degradation of 4 % in the results compared to the original GMCP-based method when the estimation of geometric transformation was used to combine the results of normalised and non-normalised images. We found that the light normalisation with SQI, when used with SIFT features, prevents the detection of large features due to the reduction of contrast between objects.

These results led us to conduct a careful investigation into the impact of illuminance normalisation with SQI on the accuracy and repeatability of SIFT and MSER detectors and descriptors. We use a graph-morphological operator known as Sieve to progressively eliminate image details and analyse differences between original images and SQI images at each scale of detail. A novel dataset was created to conduct this study on a controlled set of images carefully aligned. In the first set of experiments, we measure how the number of correspondences varies with the Sieve mesh size. We compute both SIFT and MSER with affine shape estimation and original SIFT without affine estimation for all detected regions, and check which regions correctly match with each detector. We also measure the impact of the SQI algorithm on the detection of local features. On average, the MSER detector achieves the best results. The number of corresponding areas using SIFT-affine and MSER at Sieve mesh = 10 are similar to those detected on the original image (mesh size = 1). The highest mean repeatability score of 21 % is given by MSER with a Sieve mesh = 1000. The flat areas produced by the Sieve positively affect the detection of MSER blobs, producing a few areas with high repeatability scores. MSER combined with SQI and Sieve proved to be robust for matching street-view images, increasing by 90 % the matching score in some cases when compared to SIFT features extracted from original images. Although SQI might not be the ideal illuminant normalisation algorithm for this problem, the effects of illuminant variation must be attenuated. Illuminant variation is a recurrent problem that can be quite dramatic in real-life situations.

7.3 Limitations

We evaluated the performance of HLoc+SV on a small set of 58 user images and 80 street-view geotagged images. The street-view reference images had no meta information about the camera, lens, GNSS receptor or any equipment used. Furthermore, ground truth information was manually estimated using ground marks and satellite images instead of using accurate GNSS equipment such as RTK. Ultimately, the HLoc+SV is a potential solution only suitable for the scenario when the GNSS service is available and relatively accurate.

The GMCP-based geolocation method combined with the SQI illuminance normalisation was not designed to operate in real time. Currently, it cannot download street-view reference images on-demand, instead it requires access to all images in a reference set to build the *k*d-tree. The dataset contains street-view images from three US cities only. A larger study would take a long time to conduct due to the need to process all images in the dataset. Furthermore, the low number of feature matches prevents estimating a fundamental matrix, which is necessary to compute the relative camera pose of images.

The investigation on the impact of SQI on SIFT and MSER detectors and descriptors using Sieve was done in a small set of 22 street-view images, which were manually aligned. Finally, the experiments reported in Chapter 5 and Chapter 6 do not consider the use of deep features because they were in an early stage of development when those evaluations were conducted.

7.4 Future work

The research presented in this thesis has provided insights into the development of vision-based geolocation methods. However, there are several areas where future work can be carried out to enhance the accuracy and applicability of these methods.

(a) The HLoc+SV method presented in Chapter 4 can potentially be adapted for a scenario where there is no GNSS signal. Although modern mobile devices use alternative methods to determine the user location, such as wifi triangulation, mobile phone tower triangulation, and IP address lookup, they can have errors in the order of hundreds of meters [170]. The GMCP-based image retrieval method studied in Chapter 5 also has errors of the same magnitude. In these cases, a potential solution could be to pre-build geotagged SfM models using high-resolution street view images where the user is expected to be located. As these models would cover extensive areas (potentially entire towns and cities), the global feature matching step of the original hierarchical localisation method could be used to accelerate the estimation of the camera pose. Working with large-scale SfM models can be computationally intensive, which has the risk of increasing both processing time and resource requirements, potentially making it impractical to run in real time.

(b) One area of future work is exploring the use of deep features such as SuperPoint [33] with the GMCP-based image retrieval framework [169] reported in Chapter 5. A more efficient feature detector method has the potential to increase the number of feature matches, which would increase the performance and allow estimating the camera pose.

(c) Another potential area of future work is the integration of sensors to estimate the user's geolocation, e.g. inertial sensors and electronic compasses. The use of a fusion approach to combine information from multiple sensors available on smartphones can help to improve the accuracy and robustness of vision-based methods for outdoor pedestrian navigation.

(d) The evaluations reported in this thesis can be extended by analysing other datasets covering different environments and scenarios. Field tests in urban and rural environments can provide valuable insights into the applicability of the system in real-world navigation scenarios.

(e) Ultimately, the development of a user-friendly interface for the localisation system as presented in Chapter 2, specifically the micronavigation instructions, can facilitate its adoption by non-expert users, requiring little or no training. The interface must provide clear and concise instructions for users to navigate using the system. Additionally, the interface can incorporate augmented reality overlays to provide additional information to users with low vision during navigation.

Appendix A

Light and shadows

This appendix provides an overview of the image formation process, presents the challenge of separating an image into reflectance and illuminance components, and examines two techniques for light normalisation.

A.1 Image formation

First of all, images do not exist without light. We can think of the images we see (and cameras register) as the result of light interacting with surfaces in the world and then reaching our eyes. The light shines onto a surface, part of the light is reflected back, which is finally captured by a sensor (Figure A.1). Therefore, the signal that reaches us changes depending on the characteristics of both the source light and the object surface. This presents a challenge when trying to understand the appearance of a scene in a consistent way.

To ensure clarity, we present some terminology (as defined by Adelson [2]):

Luminance is the amount of light that comes to the sensor from a surface.

Illuminance is the amount of light incident on a surface.

Reflectance is the proportion of incident light that is reflected from a surface.

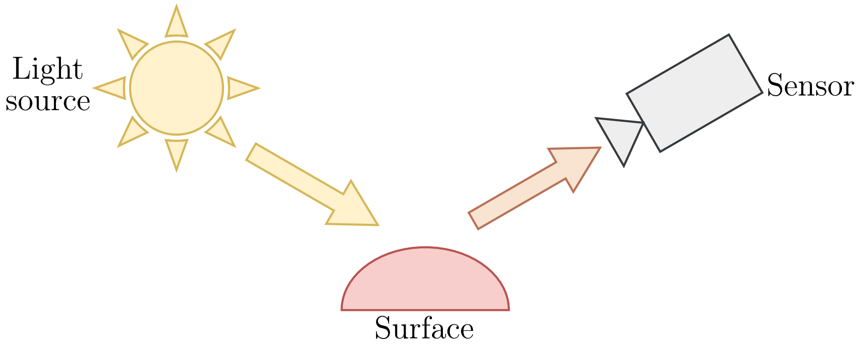


Figure A.1: The light shines onto a surface, colour changes and the new signal reflects back to the sensor.

Reflectance (or albedo) ranges from 0 to 1, equivalent to 0 % to 100 %. The ideal black is 0 %, while the ideal white is 100 %. However, in practice, typical black paint has a reflectance of approximately 5 %, and typical white paint has a reflectance of about 85 % [2]. For simplicity, we only examine matte surfaces, which can be accurately described using a single reflectance value.

Luminance, illuminance and reflectance are measurable physical quantities. Additionally, there are two related subjective variables [2]:

Lightness is the perception of a surface's reflectance based on the luminances present in a scene.

Brightness is the perceived overall level of luminance of the image itself, with no relation to the attributes of the depicted scene. It can be thought of as the perceived luminance surface of the image.

Figure A.2 provides context for understanding these terms. It shows a checker block made up of a 2×2 grid of cubes coloured either light or dark grey. The checker block is illuminated from an oblique angle, creating distinct lighting across the various faces. The luminance image can be factored into two

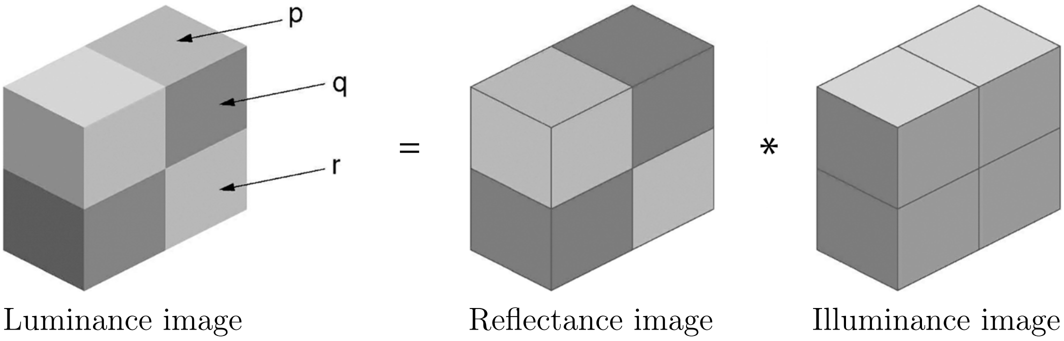


Figure A.2: The luminance image of a checker-block can be decomposed into two intrinsic images: reflectance and illuminance. Adapted from [2].

separate images: the reflectance and the illuminance. These two images are known as intrinsic images [19], which can be used to study lightness perception.

Comparing the different patches p, q and r in Figure A.2, we see that two patches may have the same reflectance, but different luminances (such as p and q); or have different reflectances and different luminances, but the same illuminance (such as q and r). Interestingly, two patches may also happen to display the same luminance, despite having distinct reflectance and illuminance (such as p and r). Although p has a lower reflectance, it is balanced by its higher illuminance. Even knowing that p and r have the same luminance, we humans perceive these two patches as being different. This counter-intuitive phenomenon is known as lightness constancy.

From a physical perspective, the lightness constancy problem can be formulated as follows. A luminance image $L(x, y)$ is the result of multiplying a reflectance image $R(x, y)$ by an illuminance image $E(x, y)$, i.e.

$$L(x, y) = R(x, y)E(x, y). \tag{A.1}$$

Looking at (A.1), it becomes clear that it is not possible to retrieve the two

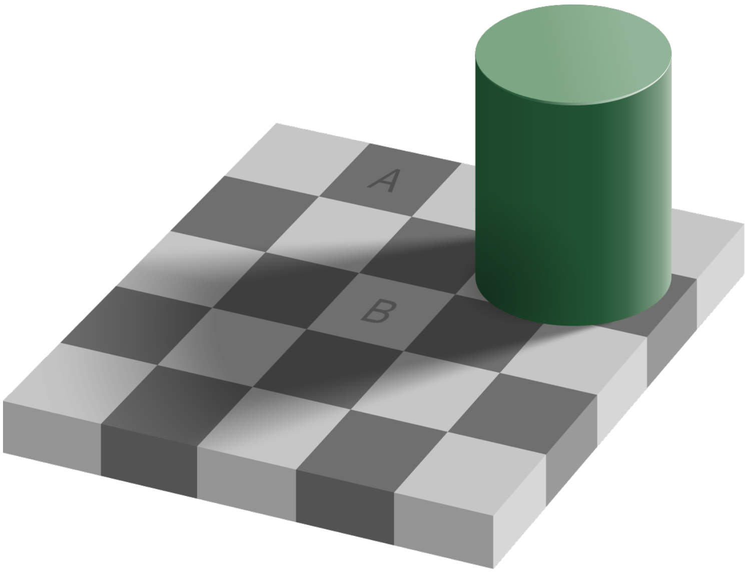


Figure A.3: Example of lightness constancy. Our visual system makes us perceive square A as darker than square B, when in fact they both have exactly the same shade of grey. Diagram originally from [3].

values $\boldsymbol{R}$ and $\boldsymbol{E}$ that were multiplied to make $\boldsymbol{L}$, as it is not possible to *unmultiply* two numbers. Given any arbitrary value for $\boldsymbol{R}(x,y)$, there are infinite possible values for $\boldsymbol{E}(x,y)$ that produce the same value $\boldsymbol{L}(x,y)$. Although this problem seems impossible to solve, our human vision system does it incredibly well.

Observe the Figure A.3. The two squares $\boldsymbol{A}$ and $\boldsymbol{B}$ have exactly the same shade of grey. However, we perceive square $\boldsymbol{B}$ as lighter than square $\boldsymbol{A}$. In fact, our human visual system can separate reflectance change from illuminance change. Thus, this must mean that reflectance and illuminance are not arbitrary. As proposed by Land and McCann in their Retinex theory [84], there are indeed statistical constraints to the possible values of reflectance and illuminance based on properties of the world.

A clear understanding of the luminance, reflectance and illuminance of a picture can help not just to identify unique local features, but also to have a better understanding of challenging scenes. This way, we explore the potential use of illuminant-invariant images as the feature space for matching street-view images for geolocation purposes. We explore in the next sections two distinct approaches: elimination of the effects of the illuminants and automatic shadow removal.

A.2 Self quotient image

The Self Quotient Image (SQI) [153] was first proposed to eliminate the effect of illumination in images for facial recognition. Compared to other methods [166], it has the advantage of having a straightforward implementation, easy application on real images and no need for training data. This method assumes that natural illumination variations are often characterised by low spatial frequencies. The illumination is normalised by dividing the image by a smoothed version of itself using local filters.

The Quotient Image (QI) is formed by dividing an image by a linear combination of three images with non-coplanar illuminants [136]. The QI depends on the albedo only, and therefore is independent of illumination. The SQI develops this method further and eliminates many assumptions, including the need for a set of aligned images.

The SQI factorises an image into two parts: an intrinsic and an extrinsic part, i.e.

$$I(x, y) = \rho(x, y)n(x, y)^T \cdot s = F(x, y) \cdot s \tag{A.2}$$

where ρ is the albedo and n represents the surface normal. $F = \rho n^T$ depends only on the albedo and surface normal and, therefore, is intrinsic. F represents the object identity. s is the illumination, therefore extrinsic.

The SQI method has two main steps: (i) estimation of the illumination and (ii) removal of the illumination effect. The first step is to estimate the extrinsic factor and generate a synthetic image $\hat{I}$ that maintains the shape and illumination of I but with a distinct albedo. The illumination normalisation is achieved by calculating the logarithmic difference between I and $\hat{I}$. The synthetic image $\hat{I}$ has a similar 3D shape and illumination as the original image I. Hence, the resulting normalised image is $\log \rho_0 - \log \rho_1$, where ρ_0 and ρ_1 are the albedo maps of I and $\hat{I}$, respectively. The normalised image $\hat{I}$ is therefore unaffected by variations in illumination.

An image I has its Self Quotient Image Q defined by

$$Q = \frac{I}{\hat{I}} = \frac{I}{F \cdot I} \tag{A.3}$$

where F is the smoothing kernel. Q is an illumination independent image for almost all regions. The regions with both no shadow and an abrupt surface normal variation are still illumination dependent.

The smoothing kernel F is defined as a weighted Gaussian filter $F = WG$, where W is the weight and G is the Gaussian kernel. A convolution region Ω is divided into two sub-regions, M_1 and M_2, based on the threshold $\tau = \text{Mean}(I_\Omega)$, i.e. the mean value of all pixels in Ω. Assuming that M_1 has more pixels than M_2, W is defined by

$$W(u, v) = \begin{cases} 0, & I(u, v) \in M_2 \\ 1, & I(u, v) \in M_1 \end{cases} \tag{A.4}$$

This filter is multi-scaled, i.e. the kernel outputs at different scales are linearly combined. Figure A.4 illustrates the formation of the smoothing kernel.

A few variations of the Self Quotient Image method have been proposed, with modifications to the filter kernel and final combination function. The

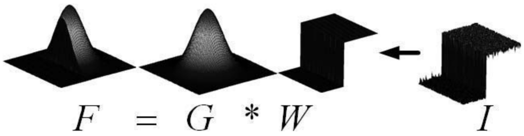


Figure A.4: Self Quotient Image weighted Gaussian filter. From [154].

Gabor Quotient Image method (GQI) [139] uses the even Gabor filter kernel G_{even} defined as:

$$G_{\text{even}}(x, y) = \cos\left(\frac{2\pi}{\lambda} x_r\right) \exp\left(-\frac{1}{2}\left(\frac{x_r^2}{\sigma_x^2} + \frac{y_r^2}{\sigma_y^2}\right)\right), \tag{A.5}$$

a linear transformation to normalise the quotient image Q to a range of $[0, 1]$:

$$Q'(x, y) = \frac{Q(x, y) - Q_{\text{min}}}{Q_{\text{max}} - Q_{\text{min}}} \tag{A.6}$$

and an exponential normalisation Q_{norm} to increase image contrast:

$$Q_{\text{norm}}(x, y) = 1 - \exp\left(-\frac{Q'(x, y)}{\text{mean}(Q'(x, y))}\right) \tag{A.7}$$

where Q_{min} and Q_{max} are the minimum and maximum values of Q, respectively. The Fast Self Quotient Image method (FSQI) [115] is another variation that uses a circularly shifted Gaussian filter kernel instead.

Implementation

We implemented the original SQI algorithm using MATLAB. It takes approximately 7.5 s to process a greyscale image with 1025 × 1082 pixels. Figure A.5 shows an example of the final result.

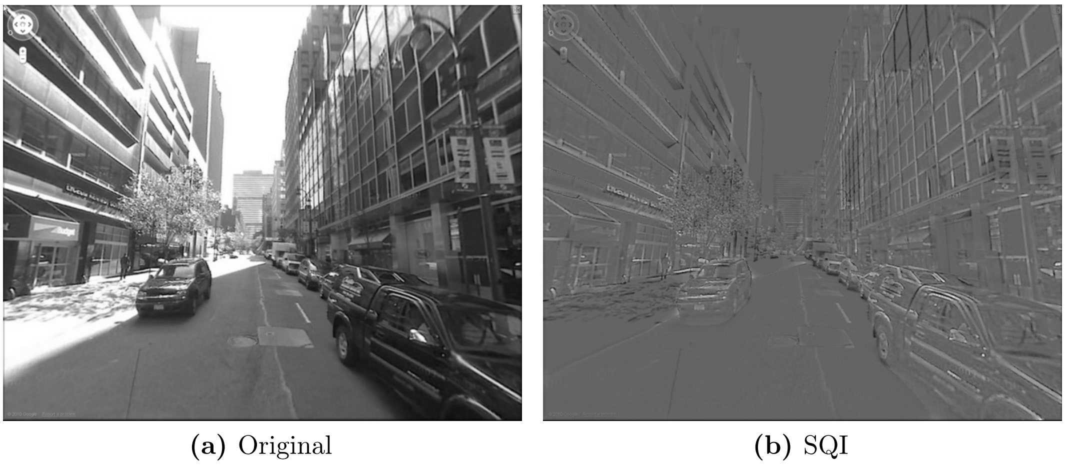

(a) Original **(b)** SQI

Figure A.5: Light normalisation using SQI applied to the reference dataset. (a) Original and (b) processed images.

A.3 Shadow removal using paired regions

The presence of shadows in images can degrade the performance of many visual tasks such as image segmentation, object recognition, tracking, and face recognition. Removing shadows is a challenging problem widely studied [42, 61, 114].

To address the problem of detecting and removing shadows from natural scene images, Guo et al. [61] employ pairwise classification to estimate the relative illumination conditions between regions based on their appearances. Using this information, a segment graph is constructed, and graph-cuts are applied to classify regions as either shadow or non-shadow. In the end, an image matting filter is applied, and a lighting model is used to re-light each pixel and produce a shadow-free image. Figure A.6 illustrates this method.

First, the image is segmented using the mean shift algorithm. A trained classifier is then used to estimate the probability that a region is in shadow. Additionally, they classify pairs of regions made of the same material as having

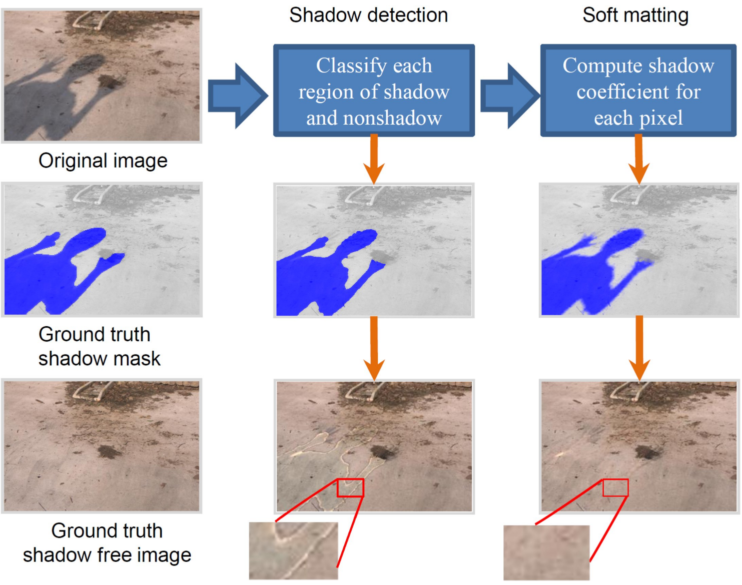

Shadow detection
Soft matting
Classify each region of shadow and nonshadow
Compute shadow coefficient for each pixel
Original image
Ground truth shadow mask
Ground truth shadow free image

Figure A.6: Method to detect and remove shadows using paired regions. From [61]

the same or different illumination. A graph is then created with the confident illumination pairs. Finally, an objective function is maximised to solve for the shadow labels. The relational graph is illustrated in Figure A.7.

The output of this process is a binary shadow mask that assigns a value $\hat{k}_i$ of 0 or 1 to each pixel i in the image, indicating whether it belongs to a shadow or not. However, using these detection results as shadow coefficients can cause strong boundary effects, as the change in illumination is often gradual and the segmentation results can be inaccurate near region boundaries. To obtain more precise shadowing coefficients $\hat{k}_i$ and achieve smooth transitions, the authors use a soft matting technique in the end.

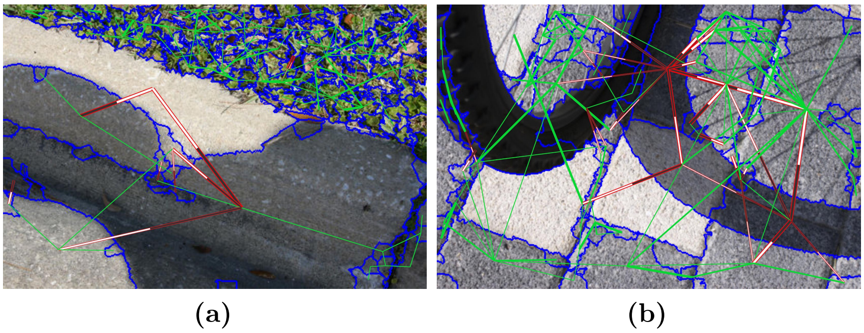


Figure A.7: Illumination relation graph plotted onto two example images. The green lines connect two areas with the same illumination. The red sticks connect areas with different illumination, in which the white end indicates non-shadow regions and the black end indicates shadows. The confidence of each pair is indicated by the line width. From [61]

Implementation

This algorithm's implementation has not been made available online and, despite our efforts to implement it based on the description given in their published papers using MATLAB, it was not possible to obtain shadow-free images that would somehow be useful to solve the problem of matching images as presented in Section 2.4.

Conclusion

In this appendix, we review the process of forming an image, covering the main concepts used in this area. We introduced the problem of decomposing an image into reflectance and illuminance and discussed implementations of illuminant normalisation algorithms applied to single images. We explored the implementation of two distinct approaches: elimination of the effects of illuminants and shadow removal.

Appendix B

Sieve algorithm

The Sieves were first proposed as morphological extensions of one-dimensional recursive median filters [18]. These filters were then extended to two dimensions using graph morphology, and since then they have been applied to a range of problems in computer vision including lip-reading [95], stereo vision [100] and image retrieval [50].

In this appendix, we provide a brief introduction to the Sieve algorithm. A more detailed mathematical description of the Sieve, its theorems and properties are available in the literature [15, 16, 49, 50, 58].

B.1 Morphological operators

The Sieve progressively removes features at each scale by merging groups of connected pixels with their neighbours. An example of a Sieve decomposition is shown in Figure B.1. At each scale, a filter removes extrema (maxima or minima) in particular areas. In the first stage, the Sieve removes flat regions of one pixel. For this example, three regions are removed. The granularity is the difference between the input and the output image at each stage. Therefore, the granules are the areas merged at a specific scale. In the second stage, the

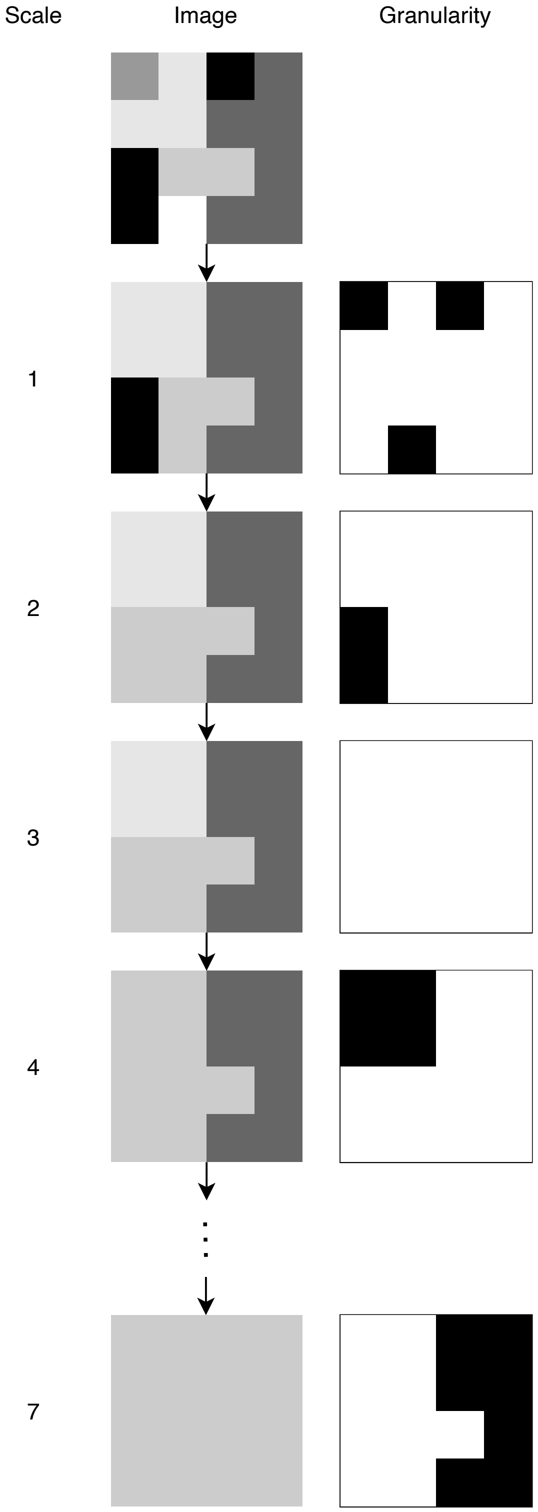


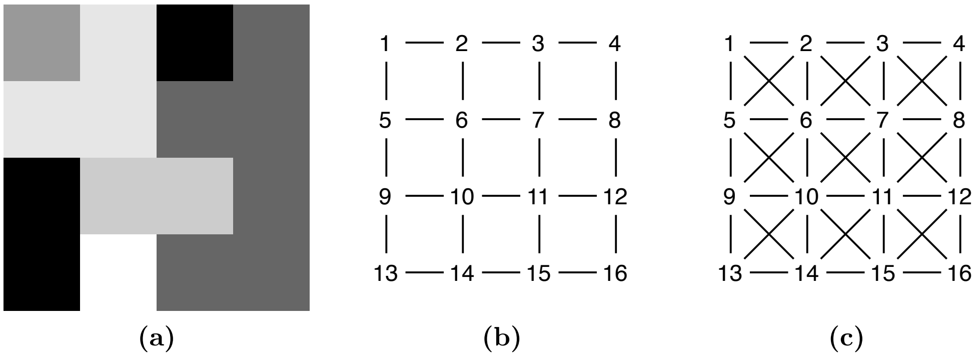
Figure B.1: Sieve decomposition process. The differences between each scale, called granules, are shown as black regions.

filter removes regions of two pixels. At this scale, only one region was removed. This process continues for bigger scales until all regions are merged into a single blob. In the example, the Sieve merges the remaining regions at scales 4 and 7, which is the whole image.

In Figure B.1, the connectivity between adjacent pixels is four-connected: each pixel has two horizontal and two vertical connections. The connectivity can also be eight-connected when diagonal links are considered (see Figure B.2). Although we just use two dimensions for images, this notation can be extended to define connected sets in an arbitrary number of dimensions.

Regarding the operation of merging pixels, four different morphological operators can be used: opening, closing, $\mathcal{M}$- and $\mathcal{N}$-filter (Table B.1). The effect of applying an opening filter, for example, would be the removal of all maxima at a specific scale. Similarly, a closing filter removes all minima smaller than a defined size. The opening and closing operators produce quite distinct results. The $\mathcal{M}$ and $\mathcal{N}$ filters differ little in practice, both remove maxima and minima at each scale.

The merging order of granules described so far, i.e. from smaller granules



(a) (b) (c)

Figure B.2: Pixel connectivity in two dimensions. (a) 4×4 pixels image in greyscale. (b) Four-connected pixels and (c) eight-connected pixels configurations. Adapted from [50].

to coarser ones, can also be called a low-pass. Two possible variations of this processing order are high- and band-passes. A high-pass Sieve merges coarser granules first, and a band-pass merges granules above and below two scales.

B.2 Sieve tree

The sequence of granules obtained at each scale of the Sieve can be organised in a hierarchical structure called a Sieve tree, where the granules form the nodes and the edges denote merging or containment. The Sieve tree structure has a node-root, and each node is a granule that has been merged into the next larger scale granule, forming a parent-child relationship. It is possible to generate more than one granule at each scale. This happens when small areas merge into distinct bigger areas, therefore forming distinct hierarchical relationships.

In Figure B.3 there is an example of decomposition into a Sieve tree. The image on the right-hand side is processed using an $\mathcal{M}$-filter. Initially, nineteen flat areas are forming the image of a cat. At the first pass, the two smallest areas corresponding to the back legs in black (minima) are merged to the adjacent flat zone with the lowest intensity level, in this case, the body of our cat. At the next step, the front legs and whiskers merged with the body.

Table B.1: Sieve morphological operators. Adapted from [58].

Operator	**Symbol**	**Extrema processing**
Opening	γ	Maxima
Closing	ψ	Minima
$\mathcal{M}$-filter	$\mathcal{M}$	Maxima-minima
$\mathcal{N}$-filter	$\mathcal{N}$	Minima-maxima

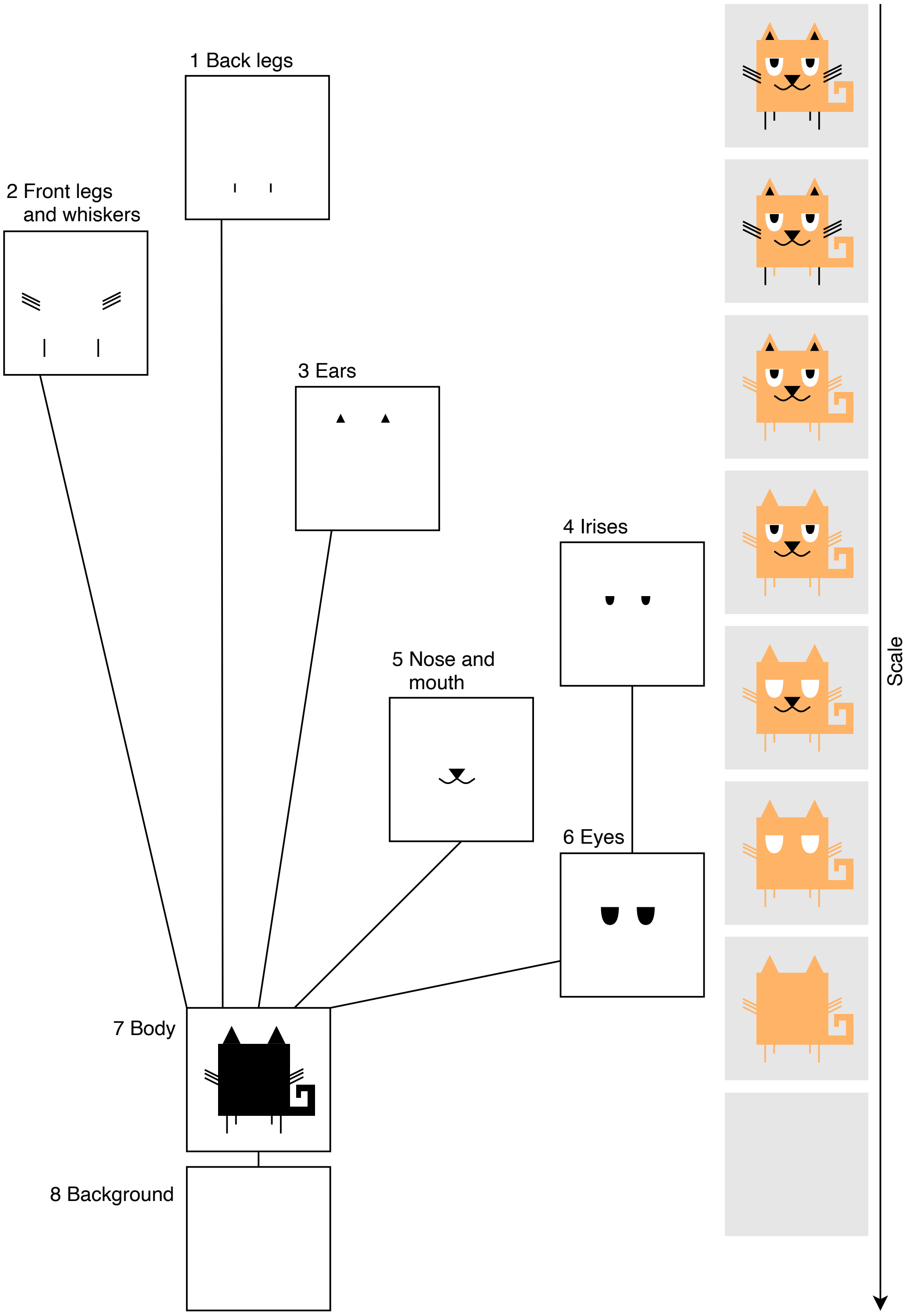


Figure B.3: Structure of a Sieve decomposition based on scale. Left: the granules, in black, form a Sieve tree. Right: the result of applying a Sieve at increasing scales on a reference image.

All these elements merged at the same pass because they have the same area. At the next scale, the ears merged with the body. Next, the irises of our cat merged with the surrounding white area corresponding to its eyes. The nose and mouth are just one connected area that merged with the body at the next step. Finally, the eyes are the last areas to merge with the body, which merges itself with the background at the final step.

The Sieve tree corresponds to the connection between the granules merged at each scale. The back legs, front legs, whiskers, ears, nose and mouth are all merged directly to the body. The irises are merged first with the eyes, which merge with the body at a greater scale. The body and the background are the latest areas to merge. In Figure B.3 we can see that the Sieve tree follows the merging order and hierarchy.

This way of merging the granules from the leaves to the root is the low-pass Sieve. Similarly, a high-pass merges the granules at the root of the tree first, and a band-pass keeps the granules between two scales and merges granules closer to both the leaves and the root of the tree.

The Sieve tree can be represented by a vector [58]

$$T = [t_1, t_2, t_3, \ldots, t_n], \tag{B.1}$$

where t_n is the parent of the nth node. The index of each node is defined by the order the granules appear, i.e. smaller granules are removed first, therefore have lower indexes. Following this notation, the Sieve tree in our example may be written as

$$T_{\text{Figure B.3}} = [7, 7, 7, 6, 7, 7, 8]. \tag{B.2}$$

B.3 Properties

It has been demonstrated that the Sieve has the following properties [15, 17, 58]:

Idempotency When the Sieve is applied to a scale, all lower scales are filtered. No smaller extreme features exist after applying the current scale filter.

Scale-space causality No new edge is introduced to the original image.

Invertibility It is possible to invert the Sieve transform and reconstruct the original image.

***n* dimensions** It works in any finite number of dimensions.

Manipulability The image can be manipulated in scale space, allowing the development of pattern recognition systems.

Calibration by scale At a particular scale, feature regions of only that scale may be measured.

Decomposition based on intensity extrema In two dimensions, the Sieve produces a connected set that decomposes an image in terms of intensity extrema of increasing area scale.

Semantically meaningful boundaries The sharp-edged region contours coincide with the semantically meaningful boundaries in the transformation domain.

Conclusion

In this appendix, we described the Sieve algorithm in terms of morphological operations and exemplified how it gradually simplifies an image by removing connected components at increasing scales. The differences between the output at each scale are called granules, which can be hierarchically organised in a Sieve tree. Some of the properties of the Sieve algorithm and the Sieve tree are considered for developing a proxy to study the effects of illuminants on the detection of local features.

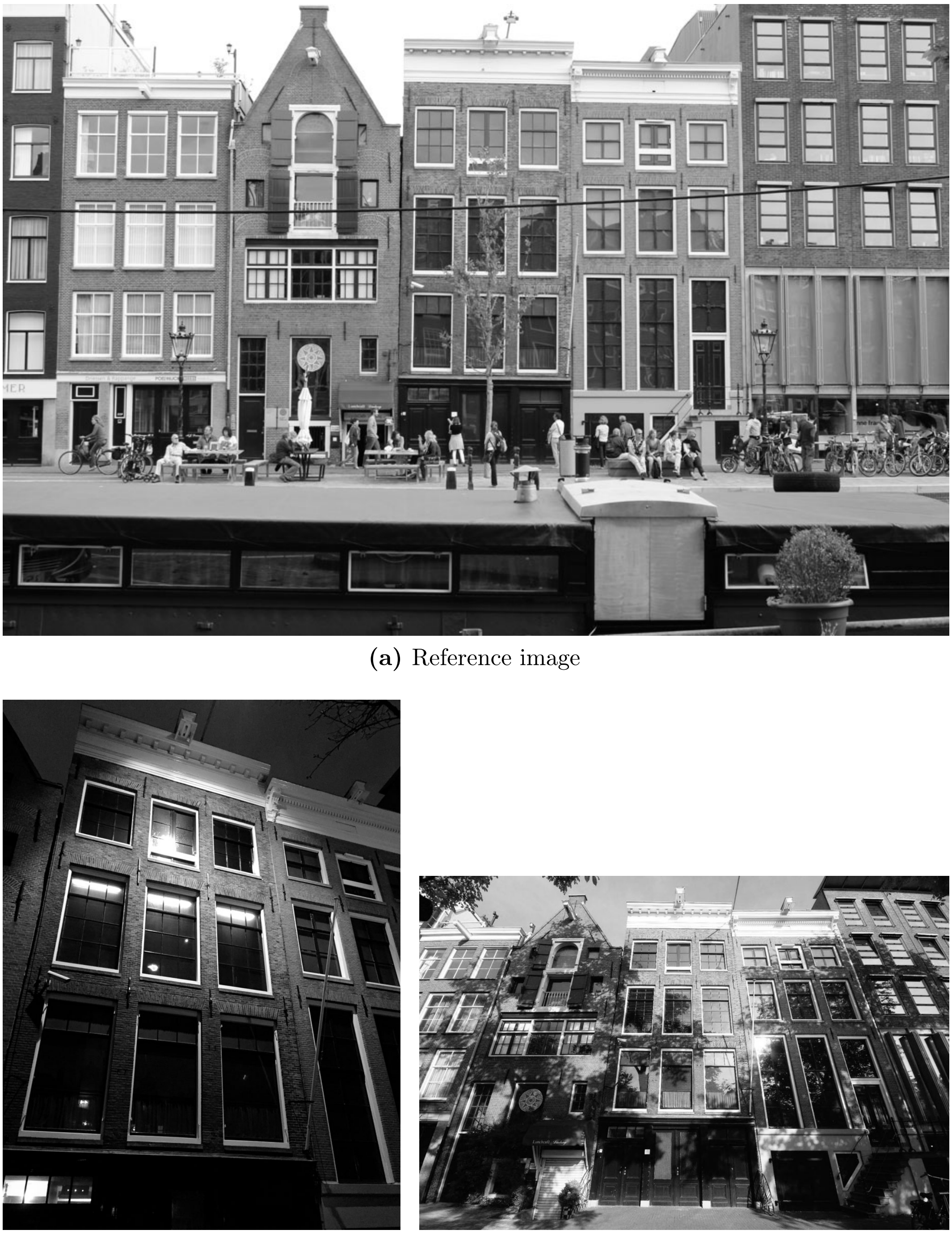
Appendix C

Examples of local feature correspondences

It is unusual to see examples of local feature correspondences in companion papers in this area. This is not a problem when there are enough local feature matches to accomplish a specific task. However, not having access to such information creates a problem in understanding the pitfalls of local feature detection methods and makes it hard to have insights when methods do not work as expected.

For the sake of completeness, in this appendix we show correspondences of local features between a reference image (Figure C.1a) and two test images (Figure C.1b and c) taken during the night and the day, respectively. Whereas Chapter 6 reported quantitative results, here we show the corresponding areas detected by MSER, SIFT-affine and SIFT algorithms on original and Self Quotient Images. The segmentation by scale provided by the sieve algorithm helps to further understand how these local features are affected by different levels of detail. These images are organised as presented in Figure C.2.

In all images, the green ellipses represent local features from the test image and pink ellipses represent features from the canonical image, projected onto the



(a) Reference image

(b) Nighttime test image

(c) Daytime test image

Figure C.1: Images used in this experimental test. (a) Reference image. (b,c) Test images took during the night and the day, respectively.

test image by applying a transformation matrix. Just the features considered correspondents are shown. The criteria to consider a correspondence was explained in detail in Section 6.3. In a nutshell, the detected areas must have an overlap of at least 60 % when normalised to a radius of 30 pixels. This normalisation becomes more apparent when comparing small features. In some cases, the overlap of non-normalised features is minimal, yet the intersection of their normalised areas is great enough to be considered as a correspondence and a potential match.

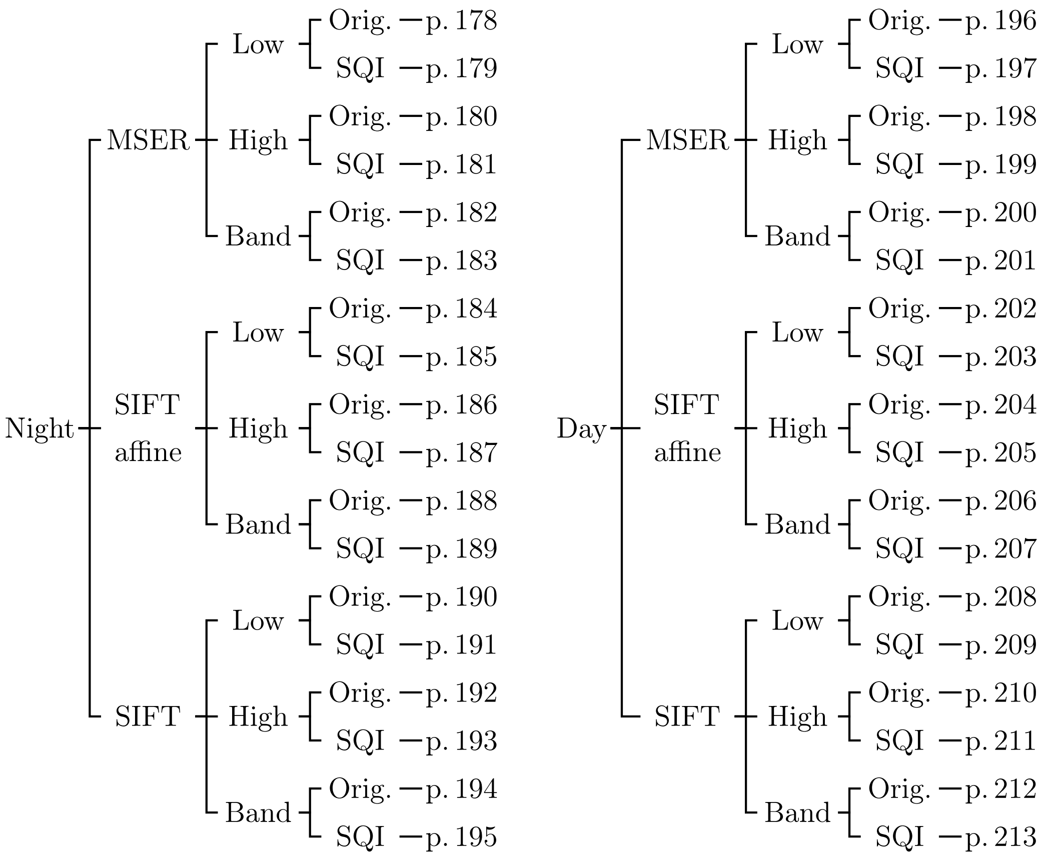


Figure C.2: Index of images presented in this appendix. Hierarchy from left to right: time, detection algorithm, sieve pass, image type, page.

Night, MSER, low-pass sieve, original image

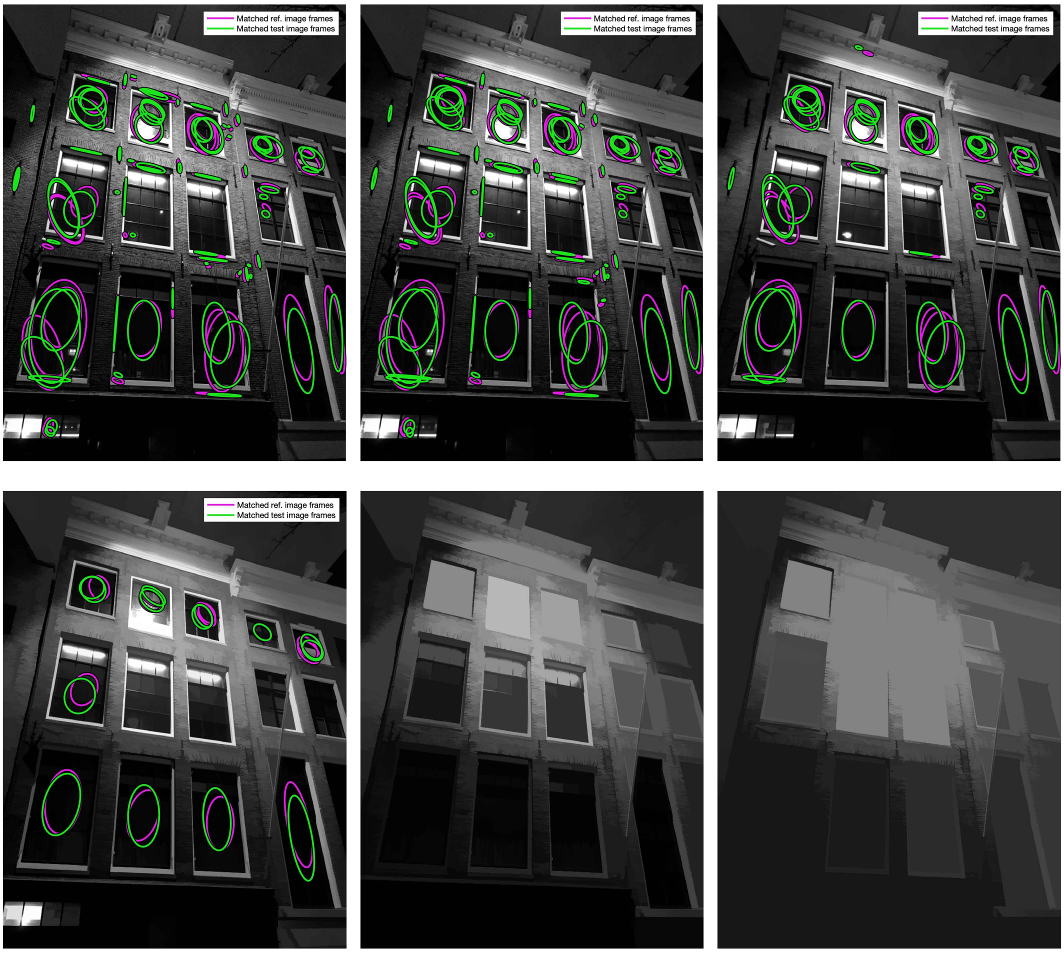

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.3: MSER correspondences on a sample image taken during the night processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, MSER, low-pass sieve, Self Quotient Image

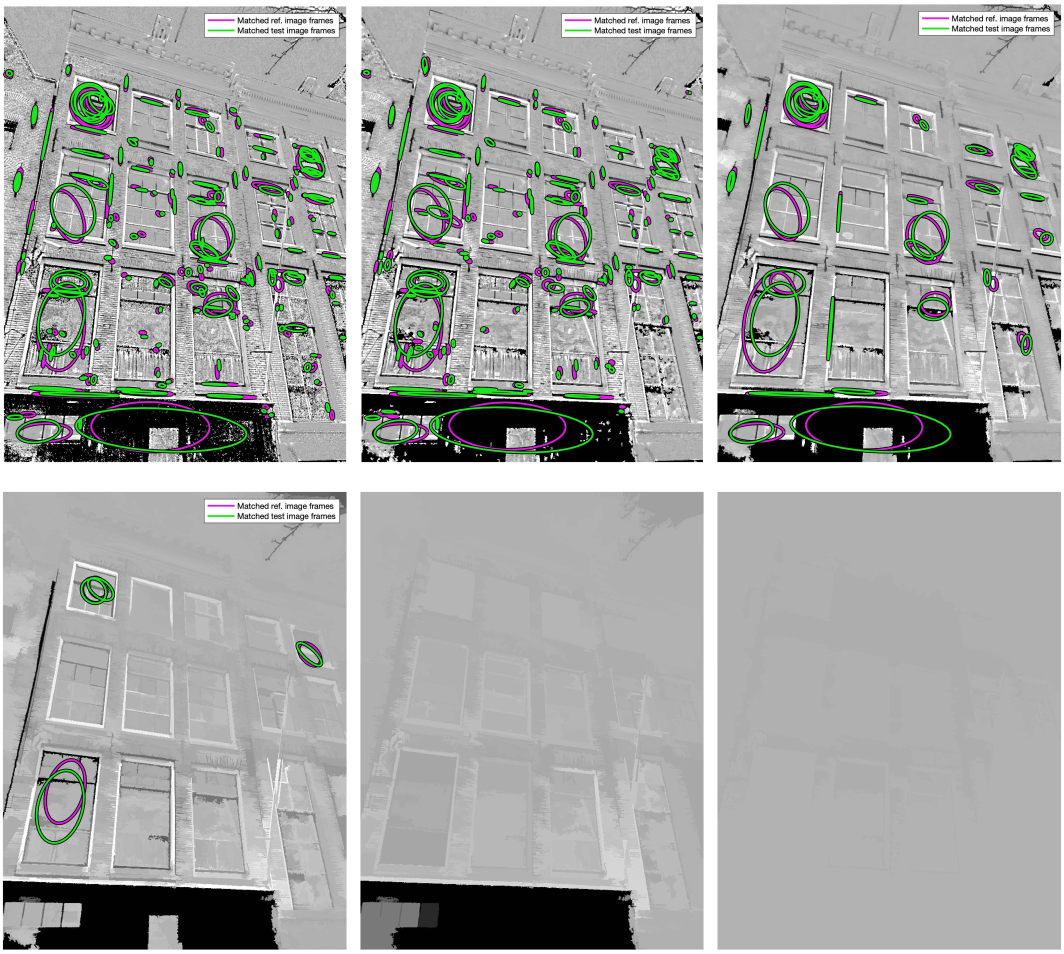

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.4: MSER correspondences on a sample Self Quotient Image taken during the night processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, MSER, high-pass sieve, original image

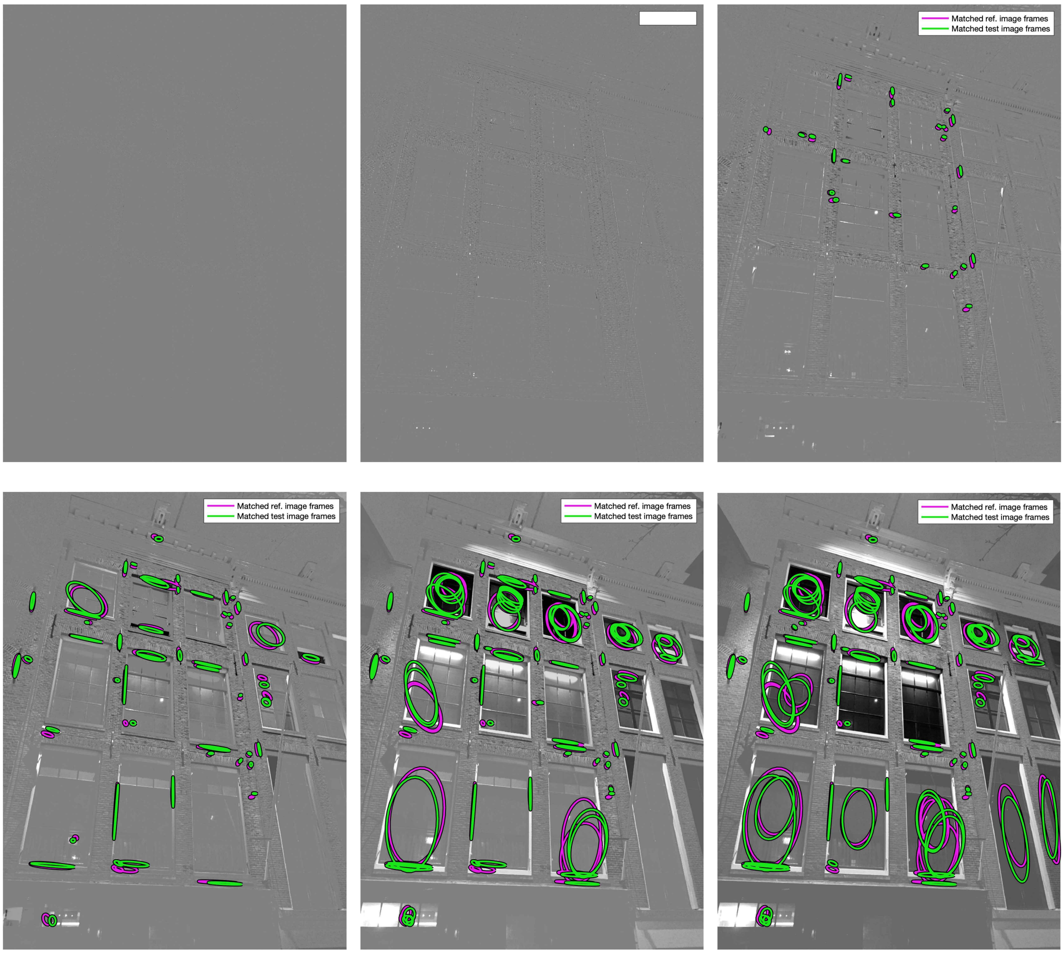


Figure C.5: MSER correspondences on a sample image taken during the night processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, MSER, high-pass sieve, Self Quotient Image

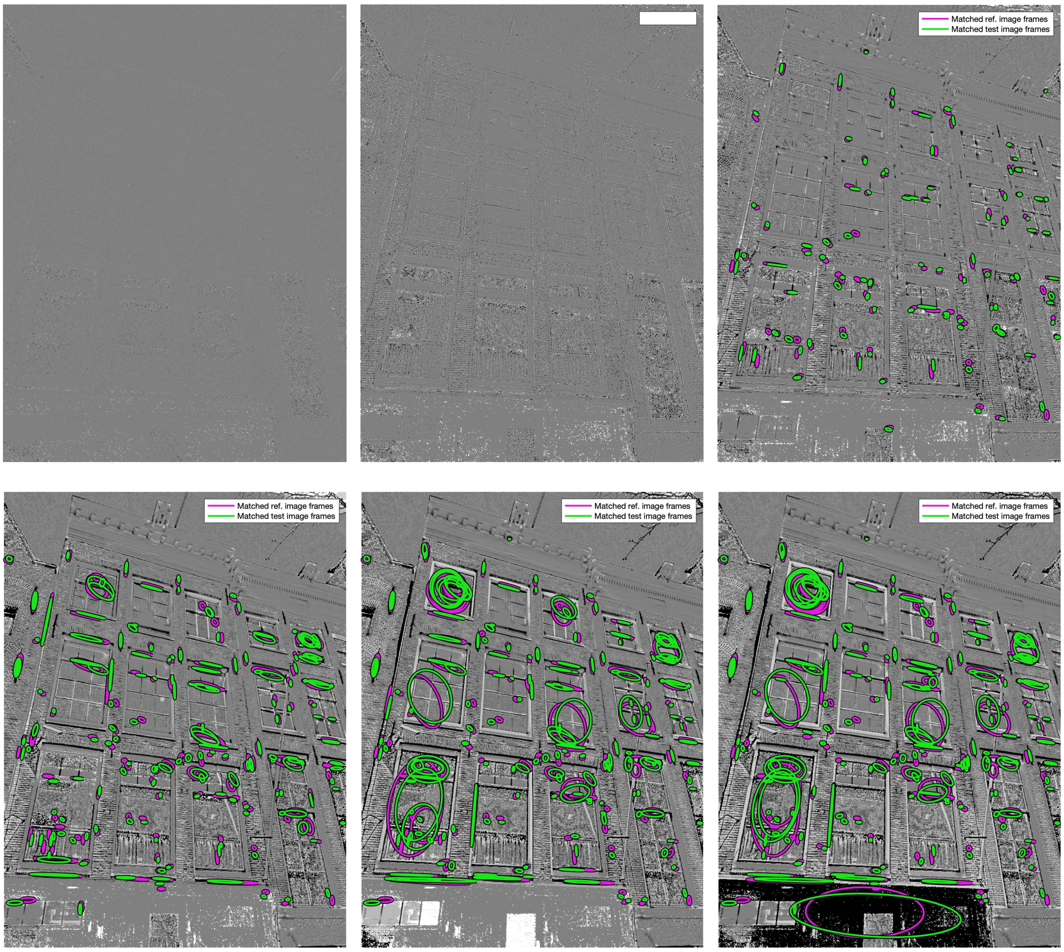


Figure C.6: MSER correspondences on a sample Self Quotient Image taken during the night processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, MSER, band-pass sieve, original image

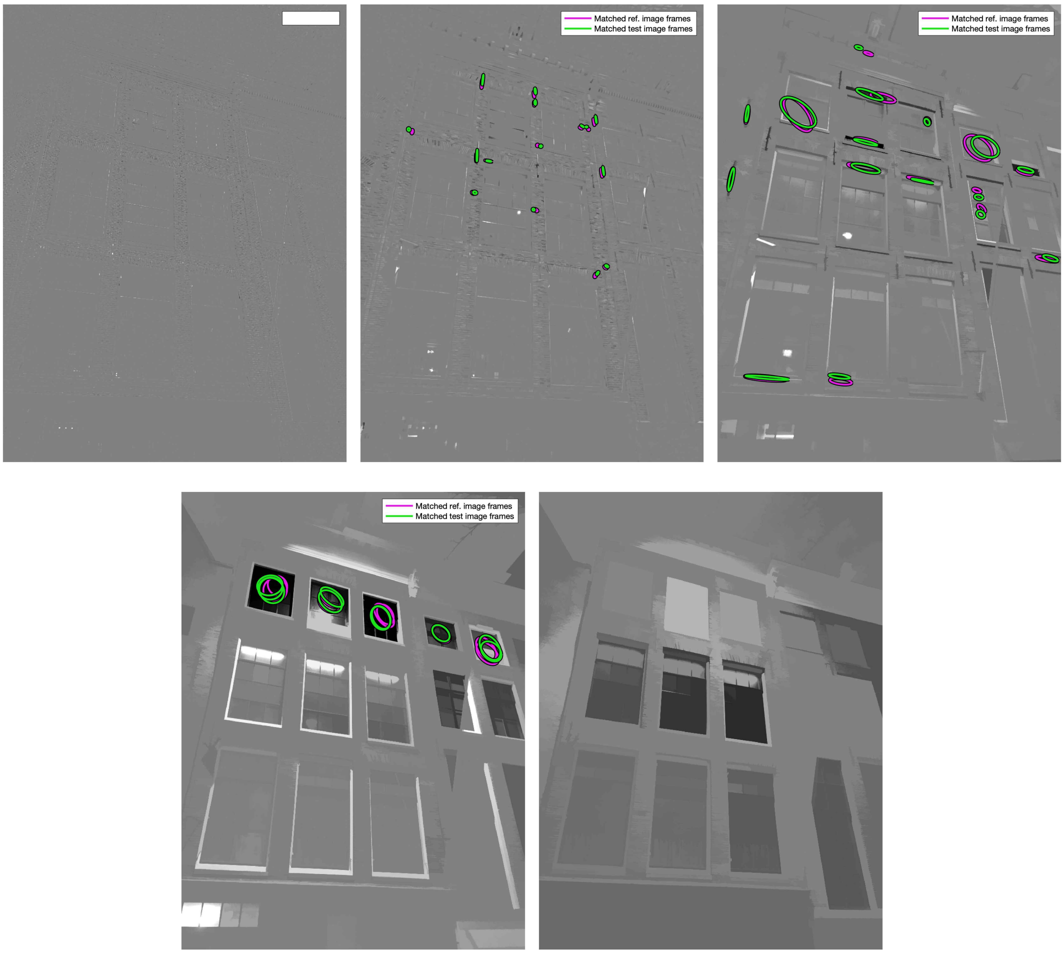

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.7: MSER correspondences on a sample image taken during the night processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Night, MSER, band-pass sieve, Self Quotient Image

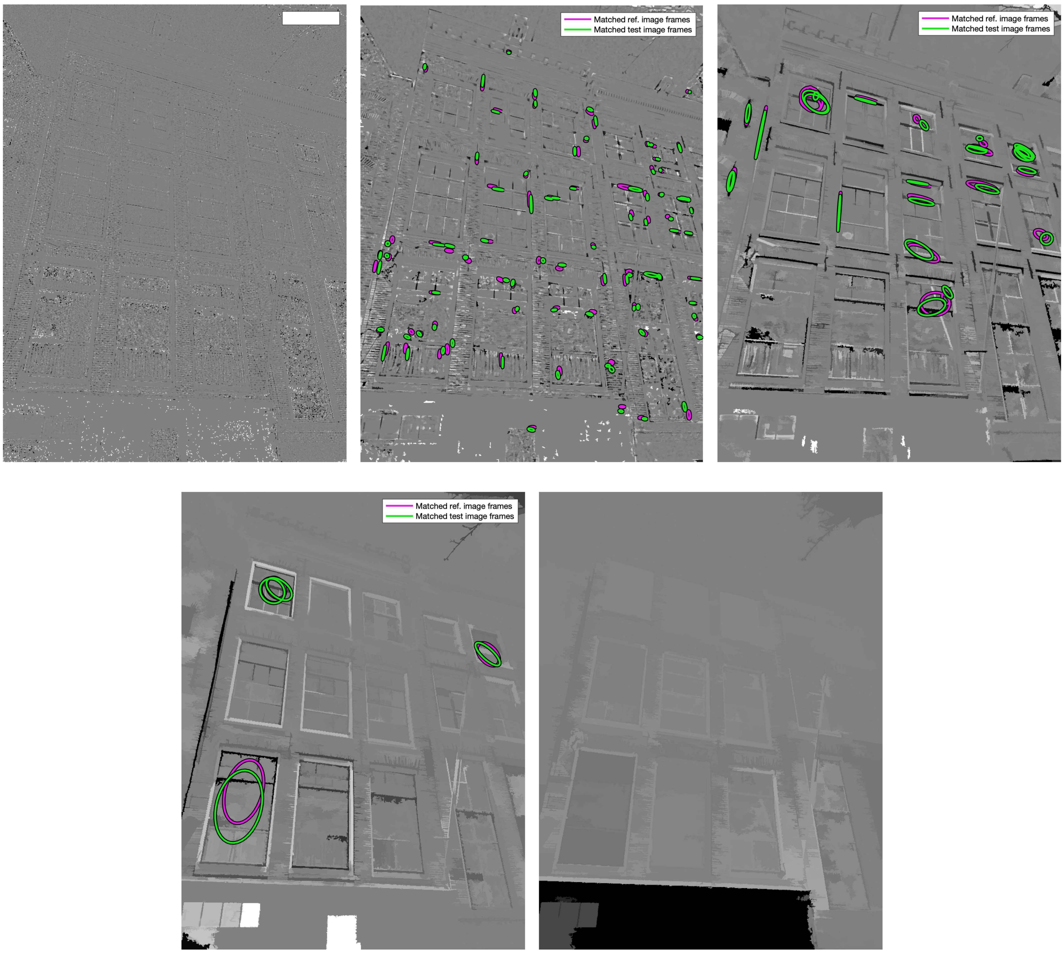


Figure C.8: MSER correspondences on a sample Self Quotient Image taken during the night processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \dots 4$ (shown left-to-right).

Night, SIFT-affine, low-pass sieve, original image

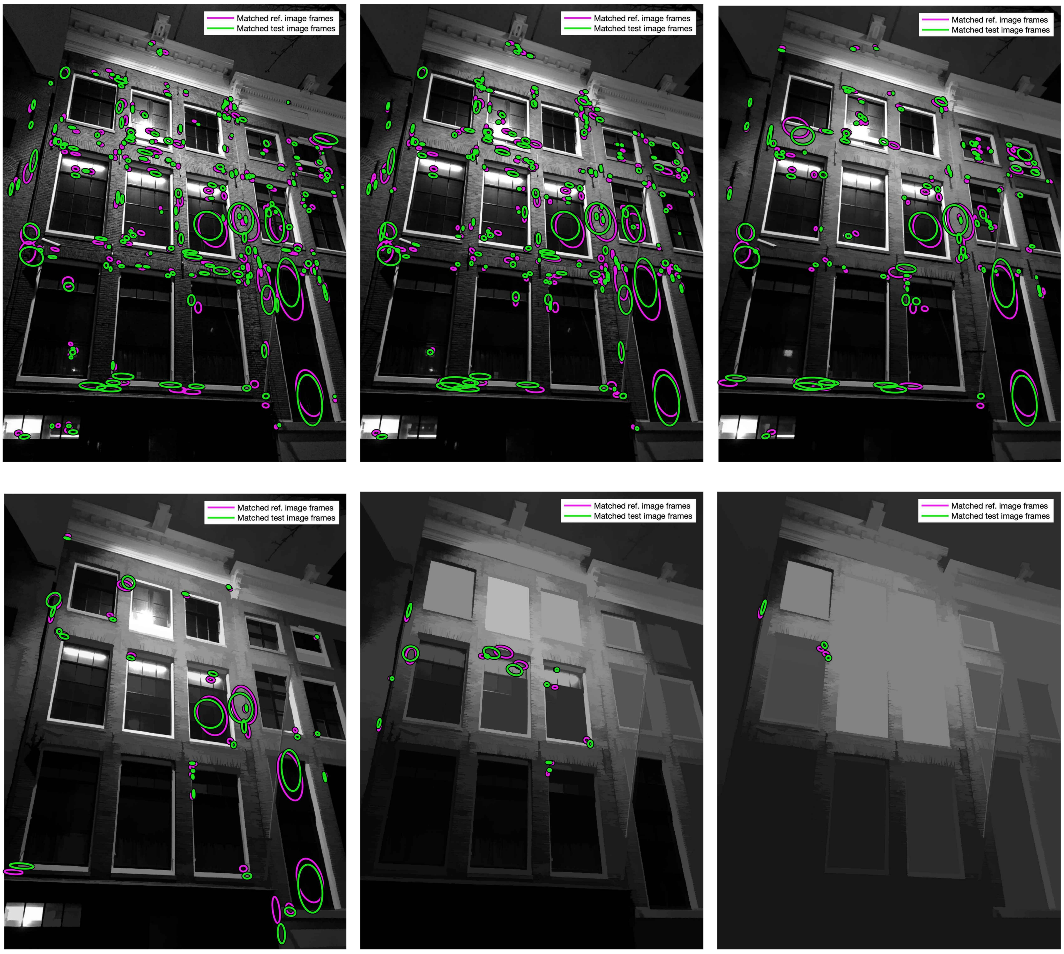

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.9: SIFT-affine correspondences on a sample image taken during the night processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \dots 5$ (shown left-to-right).

Night, SIFT-affine, low-pass sieve, Self Quotient Image

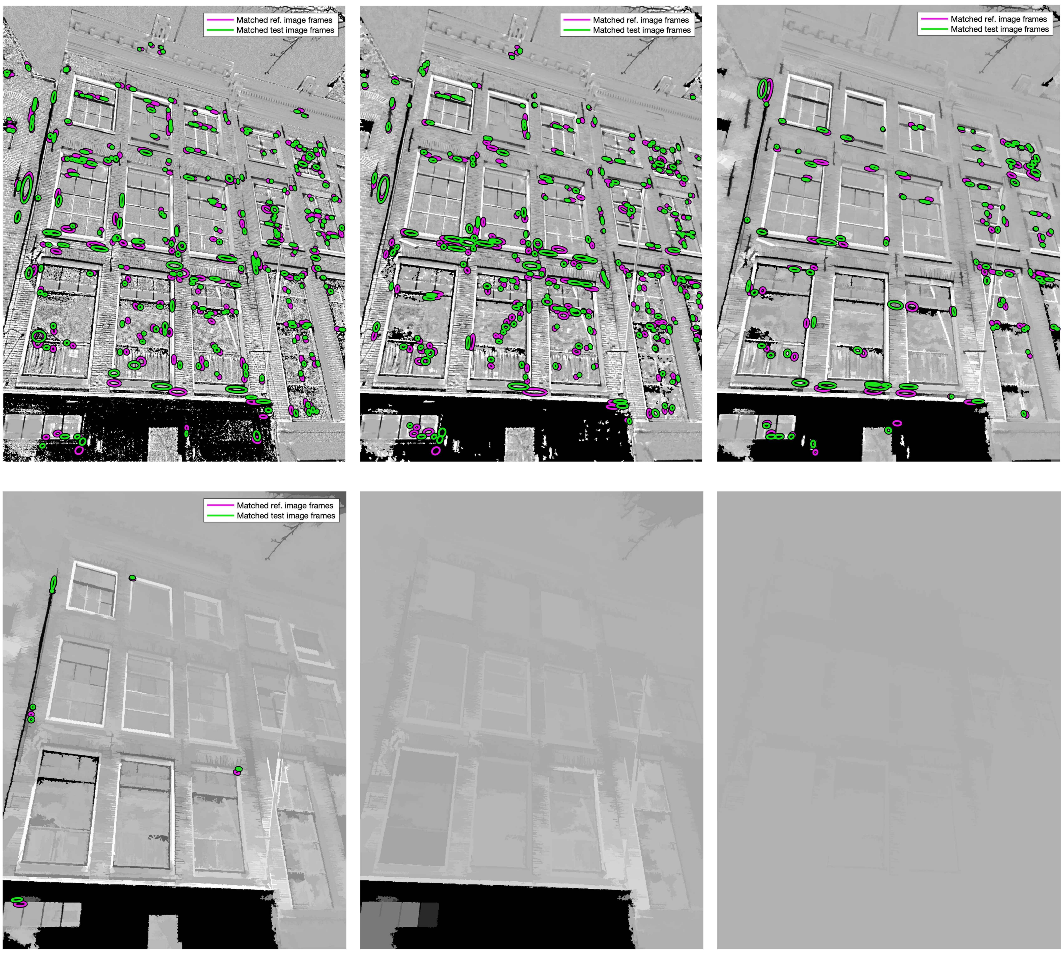


Figure C.10: SIFT-affine correspondences on a sample Self Quotient Image taken during the night processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT-affine, high-pass sieve, original image

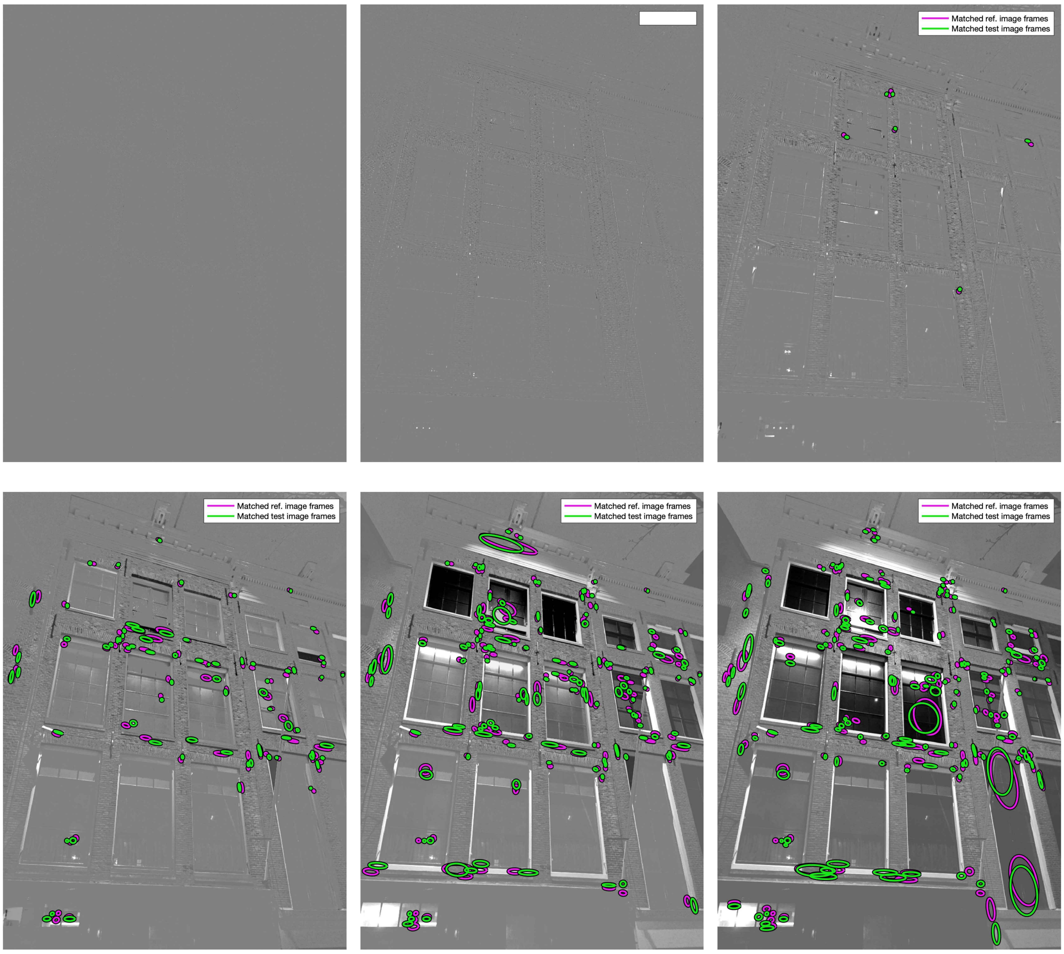


Figure C.11: SIFT-affine correspondences on a sample image taken during the night processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT-affine, high-pass sieve, Self Quotient Image

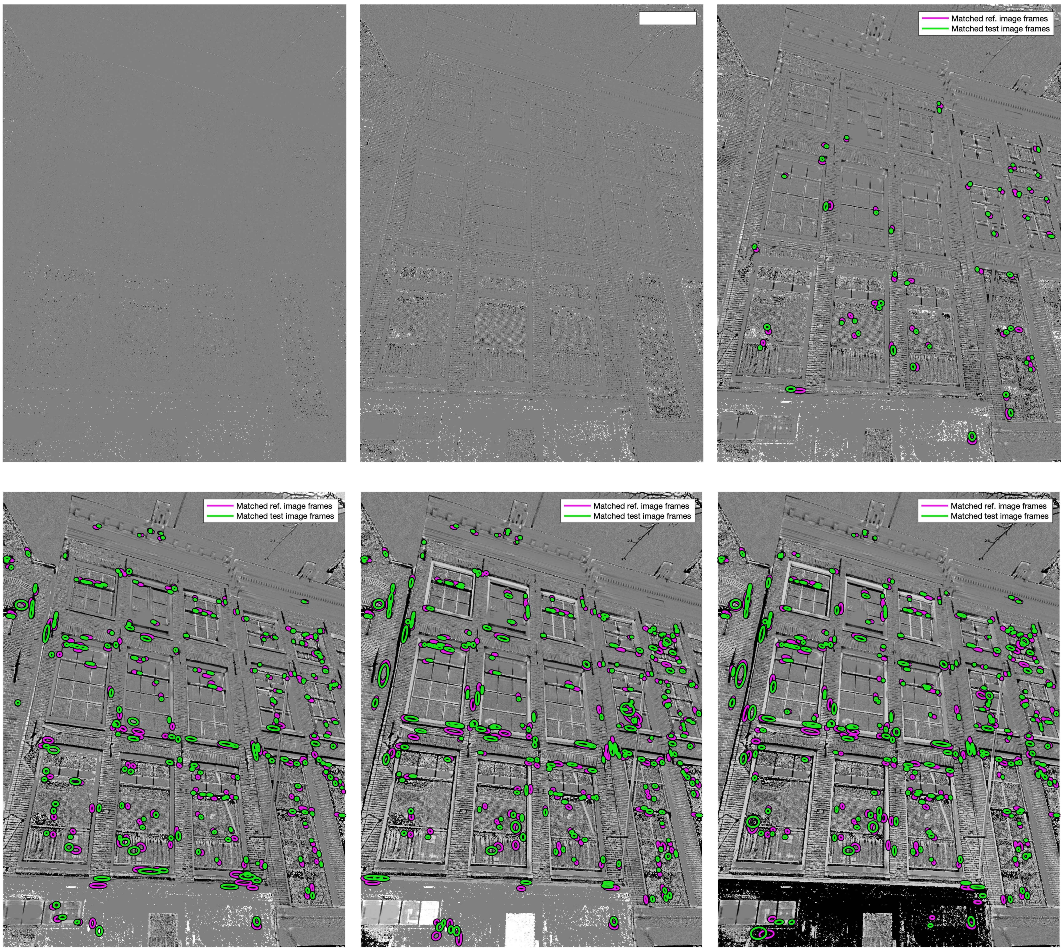


Figure C.12: SIFT-affine correspondences on a sample Self Quotient Image taken during the night processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT-affine, band-pass sieve, original image

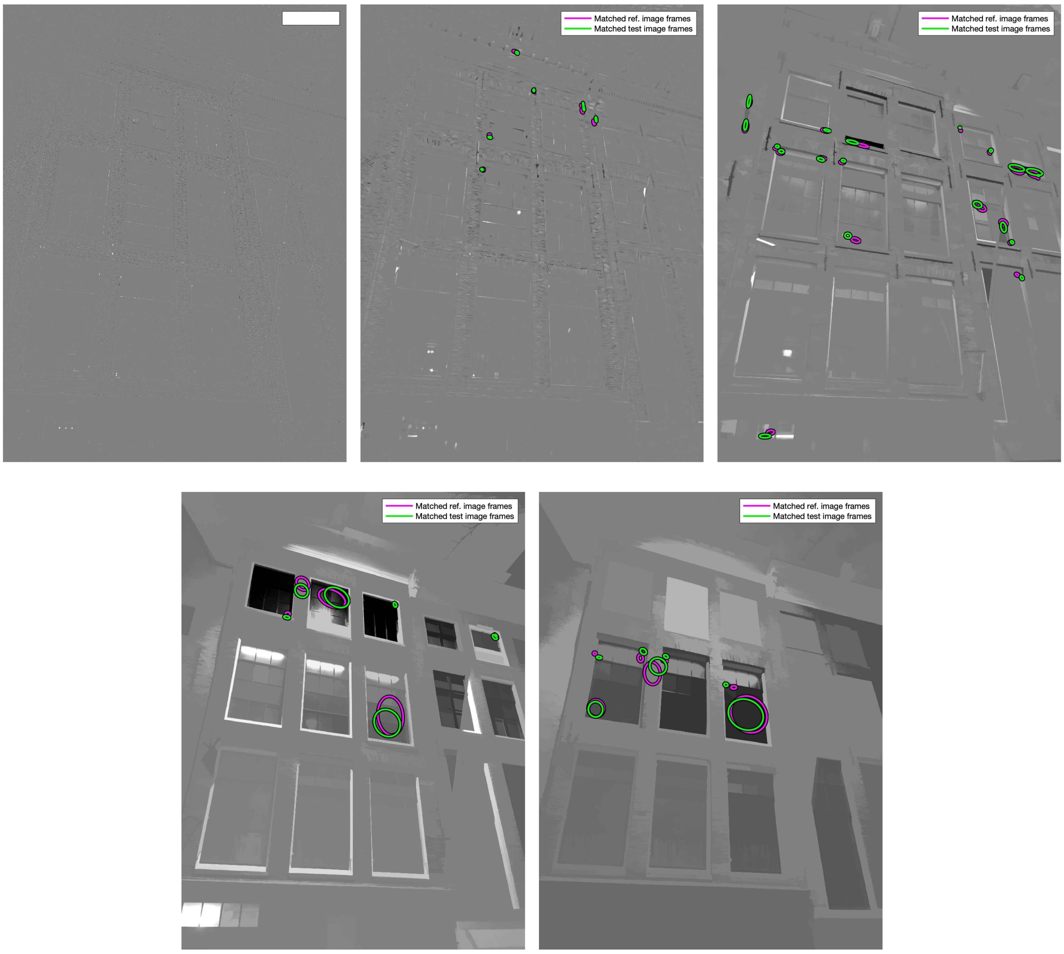


Figure C.13: SIFT-affine correspondences on a sample image taken during the night processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Night, SIFT-affine, band-pass sieve, Self Quotient Image

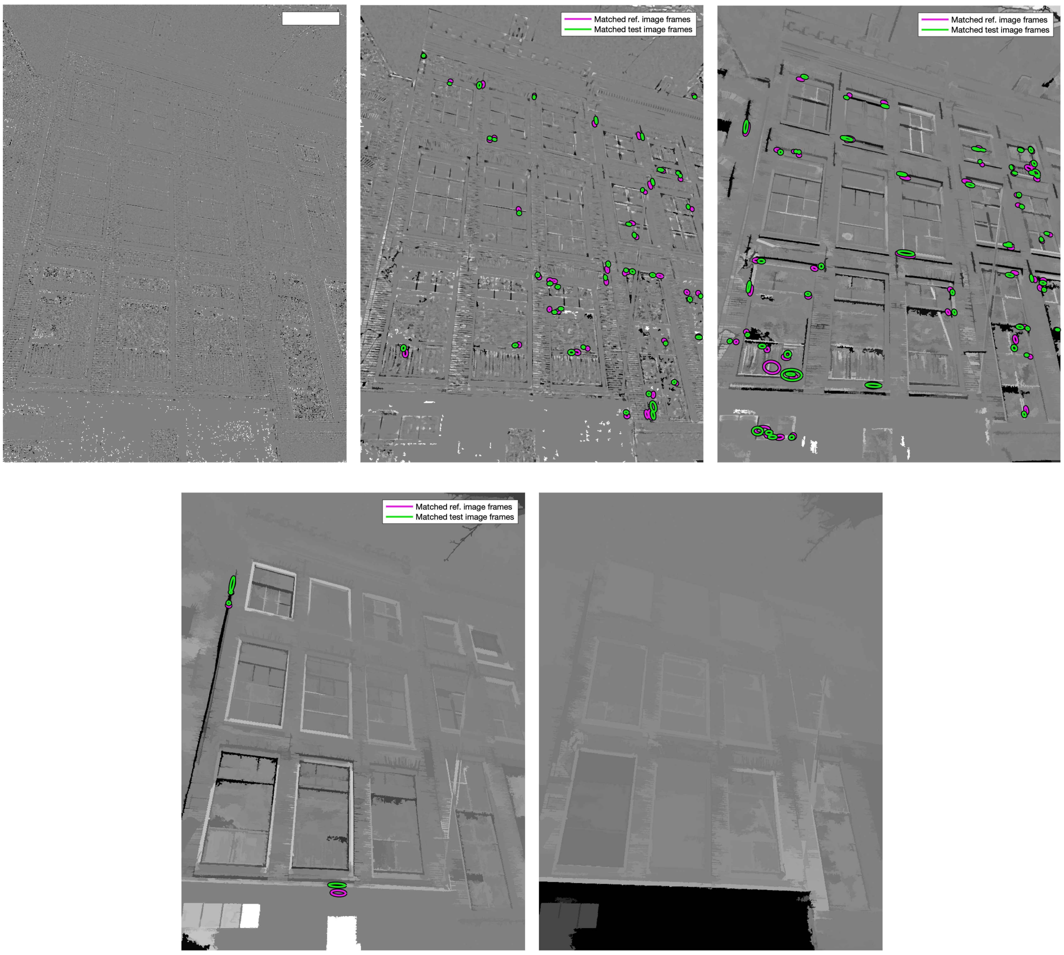

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.14: SIFT-affine correspondences on a sample Self Quotient Image taken during the night processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Night, SIFT, low-pass sieve, original image

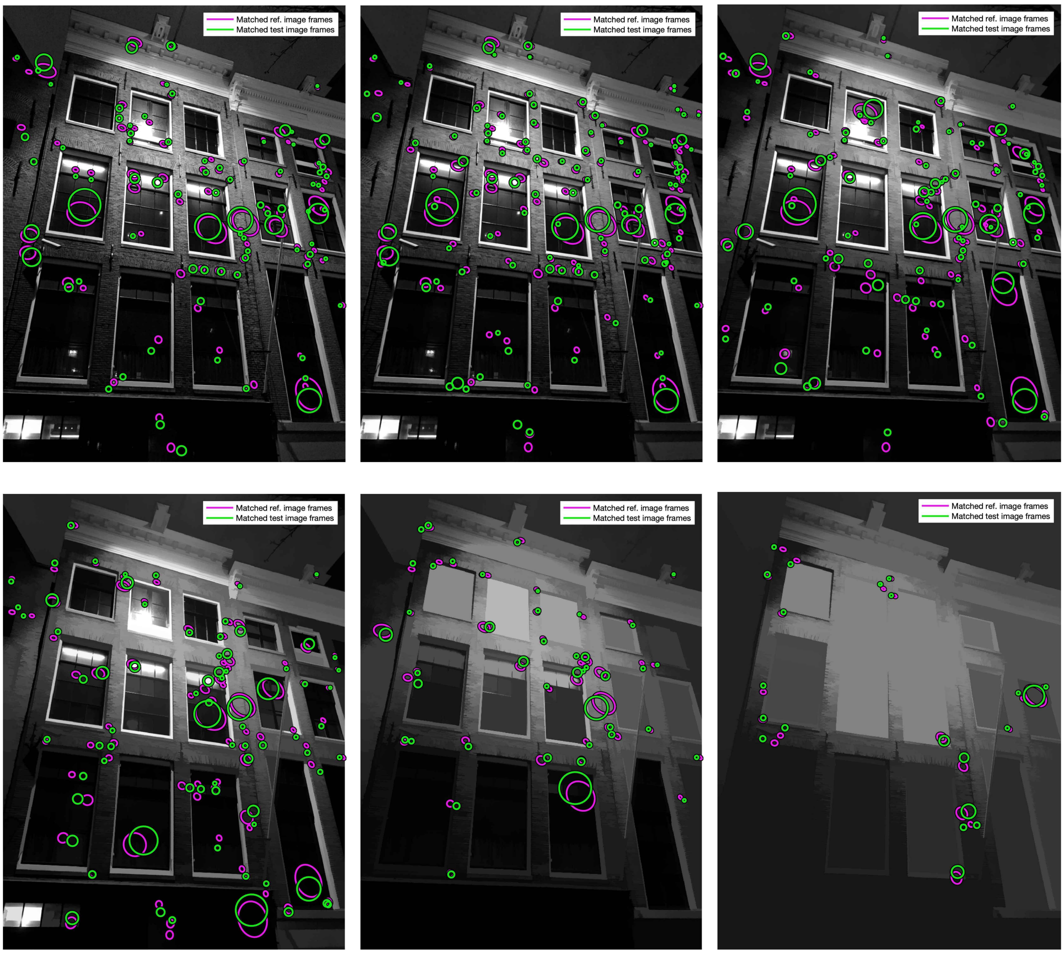

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.15: SIFT correspondences on a sample image taken during the night processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT, low-pass sieve, Self Quotient Image

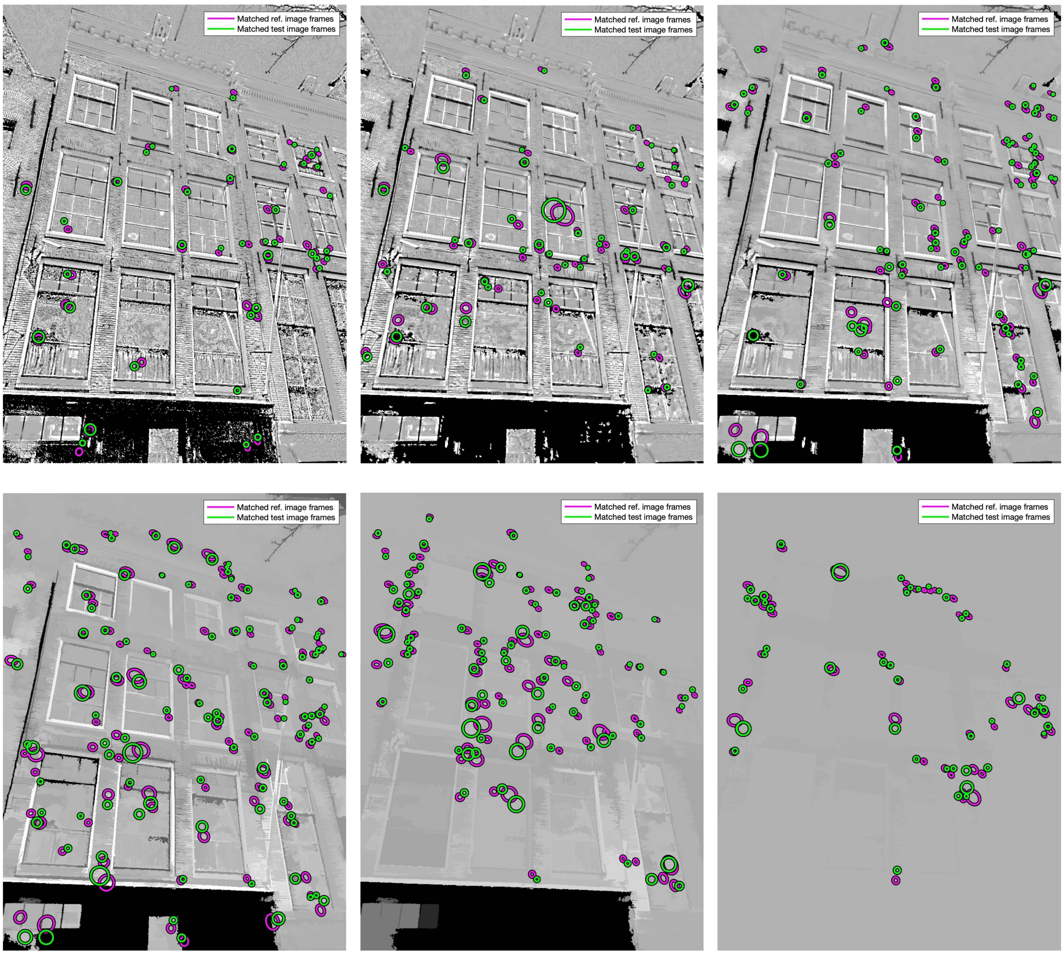

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.16: SIFT correspondences on a sample Self Quotient Image taken during the night processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT, high-pass sieve, original image

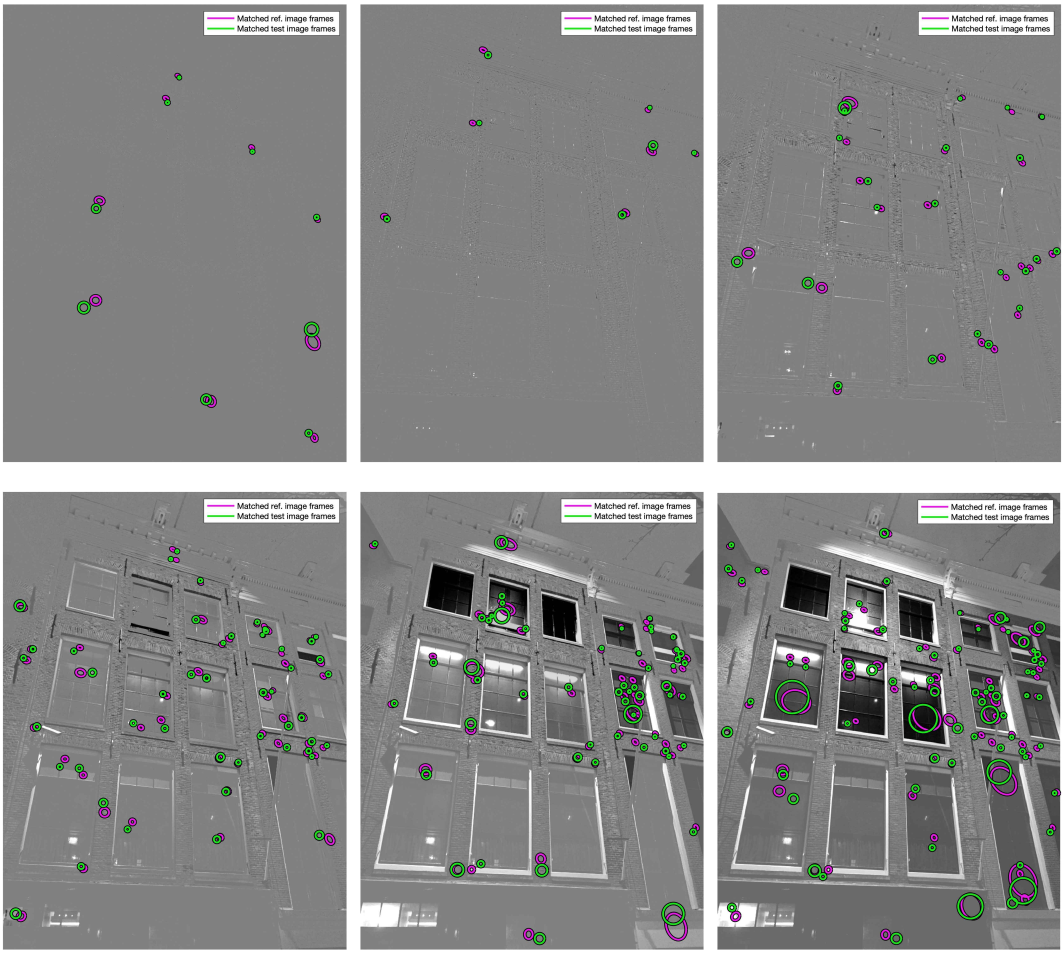


Figure C.17: SIFT correspondences on a sample image taken during the night processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT, high-pass sieve, Self Quotient Image

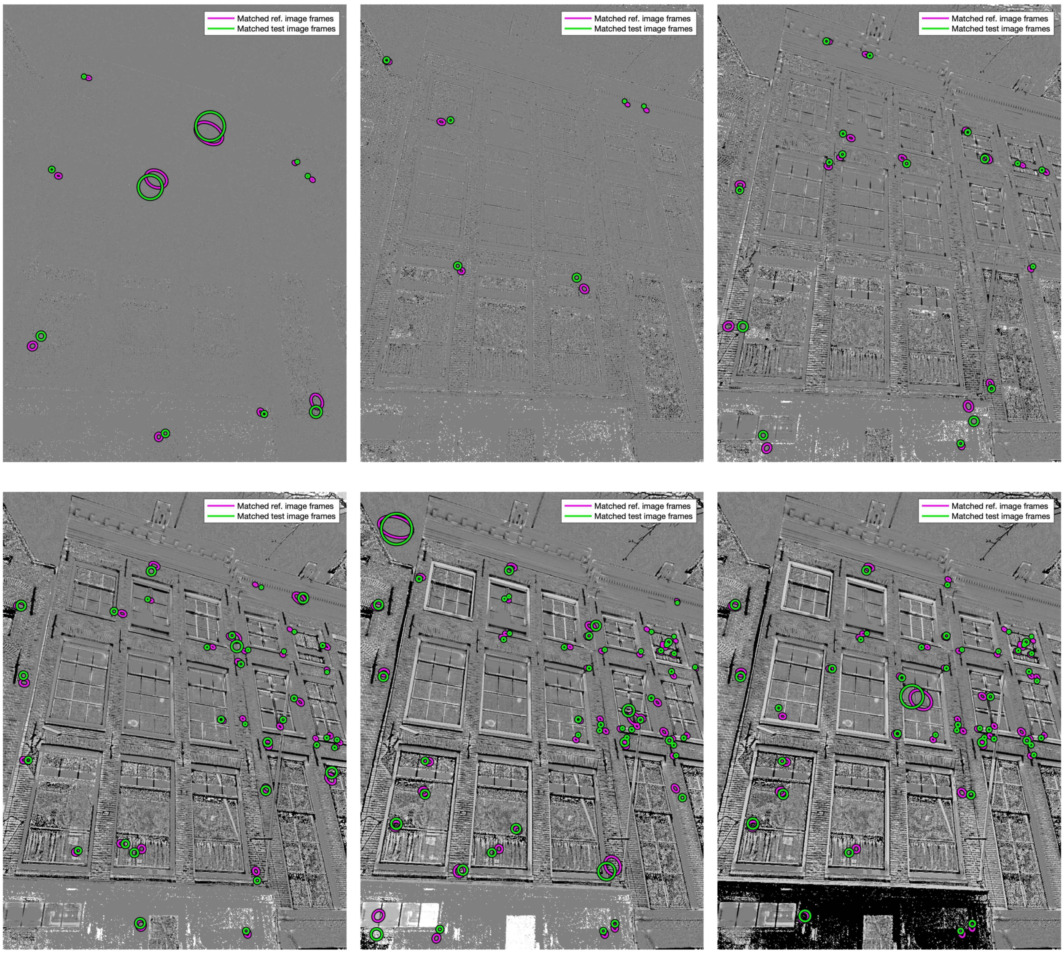


Figure C.18: SIFT correspondences on a sample Self Quotient Image taken during the night processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Night, SIFT, band-pass sieve, original image

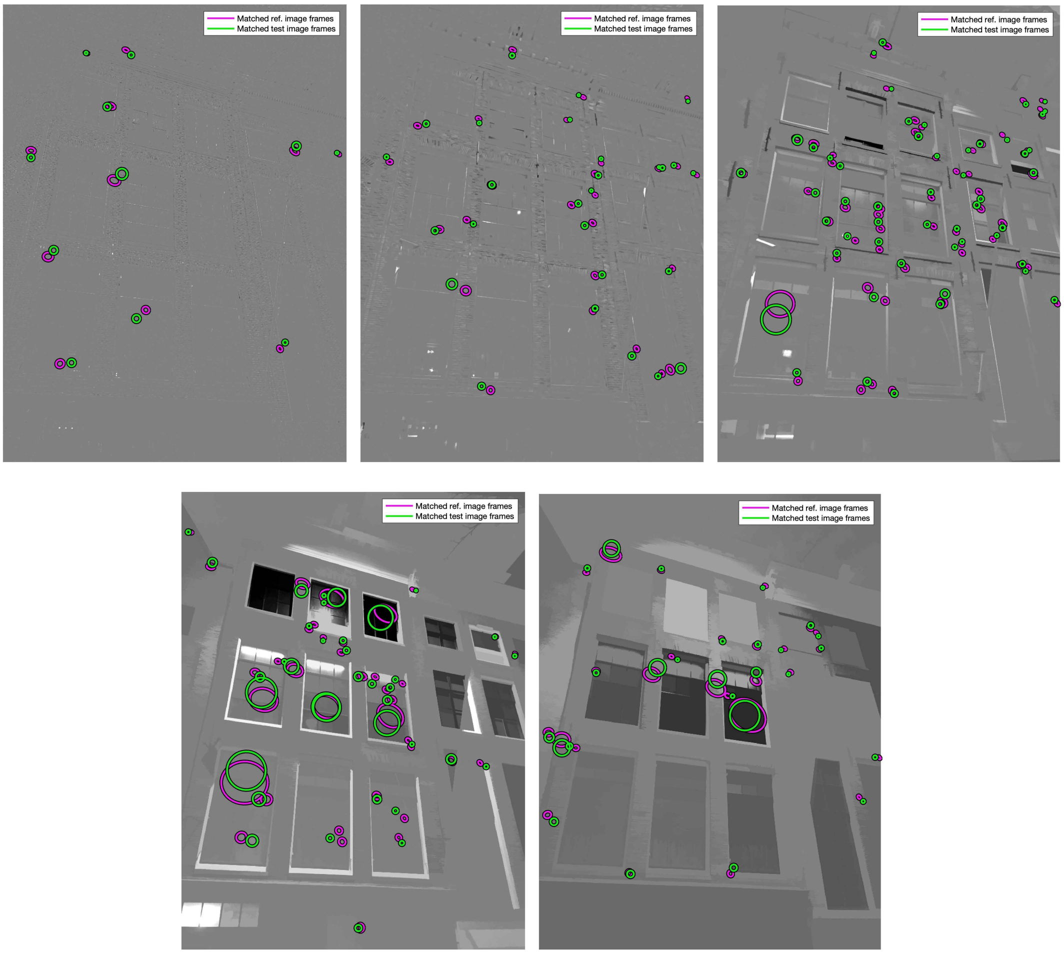

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.19: SIFT correspondences on a sample image taken during the night processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Night, SIFT, band-pass sieve, Self Quotient Image

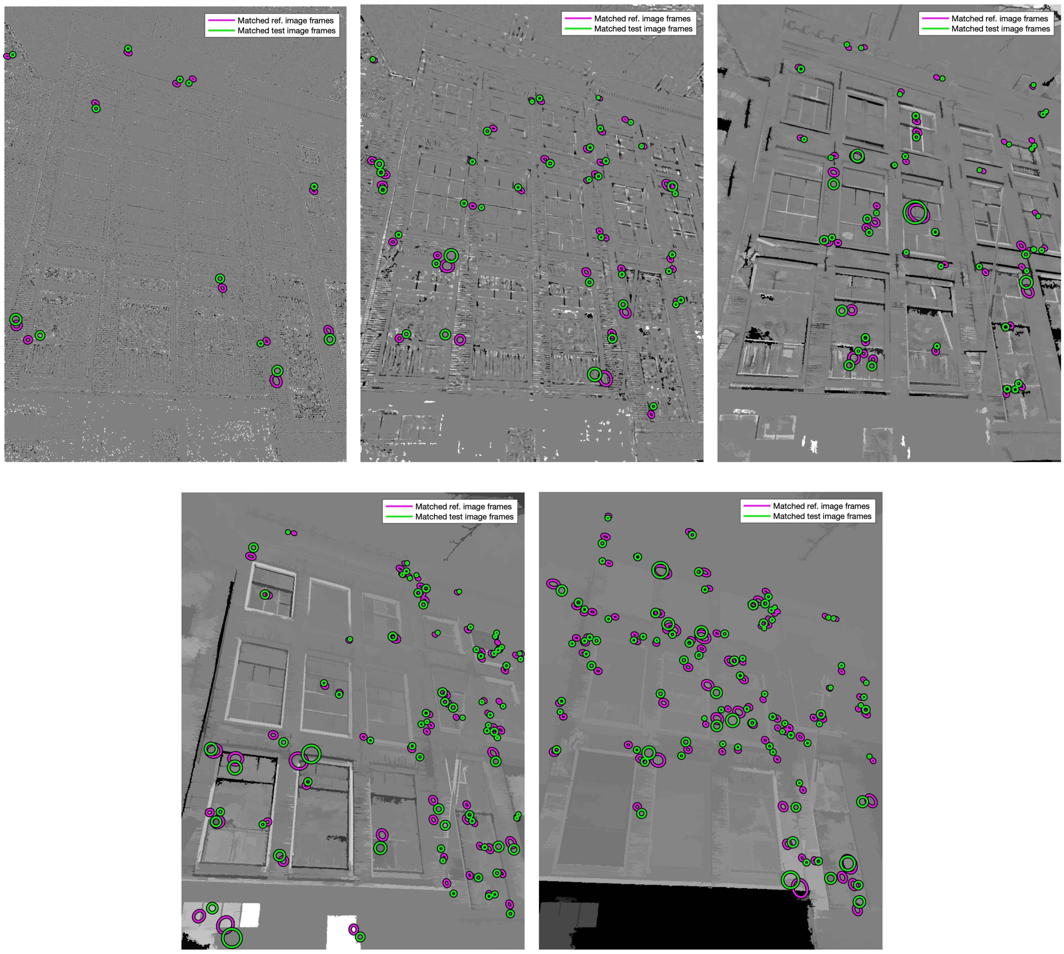

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.20: SIFT correspondences on a sample Self Quotient Image taken during the night processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \dots 4$ (shown left-to-right).

Day, MSER, low-pass sieve, original image

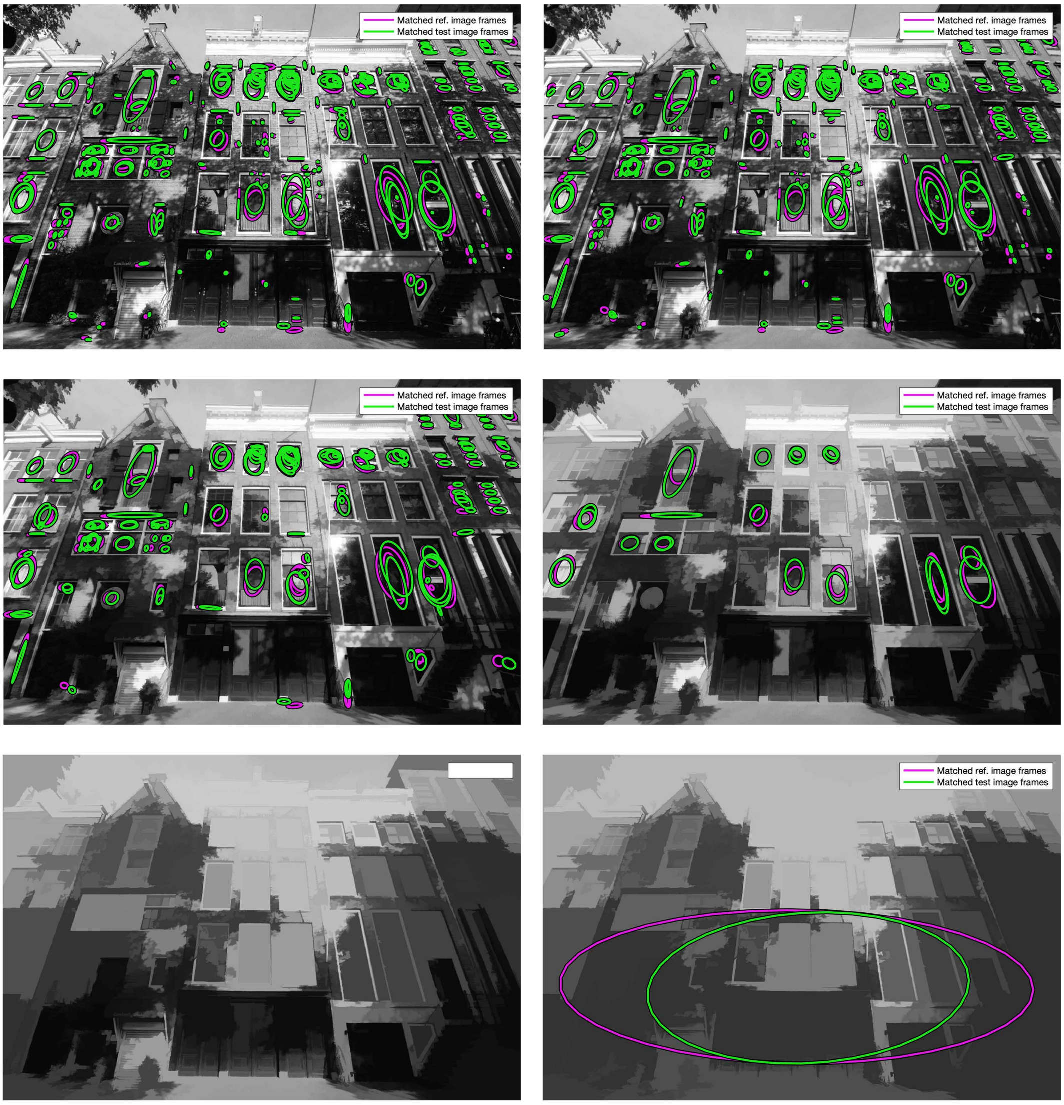


Figure C.21: MSER correspondences on a sample image taken during the day processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, MSER, low-pass sieve, Self Quotient Image

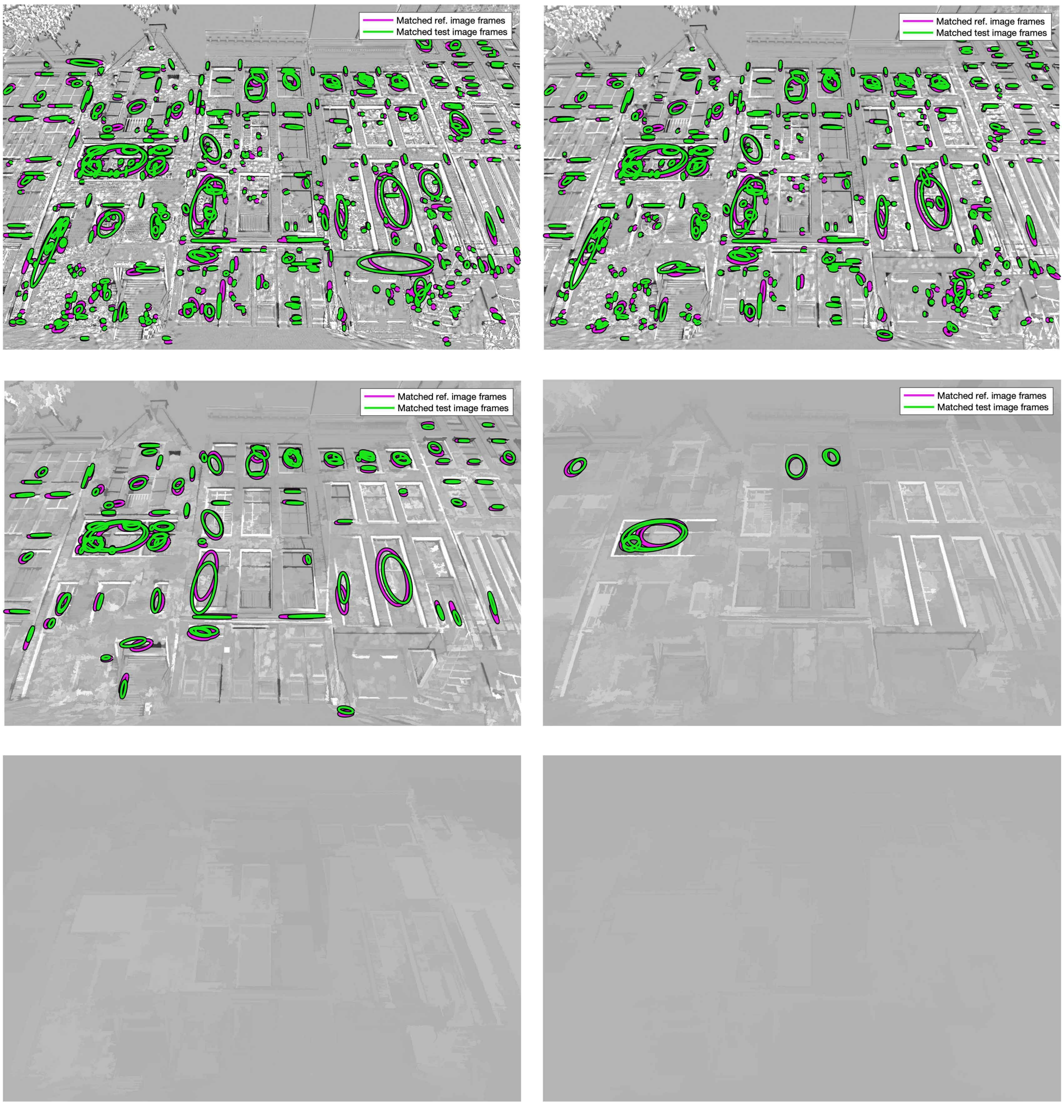

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.22: MSER correspondences on a sample Self Quotient Image taken during the day processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, MSER, high-pass sieve, original image

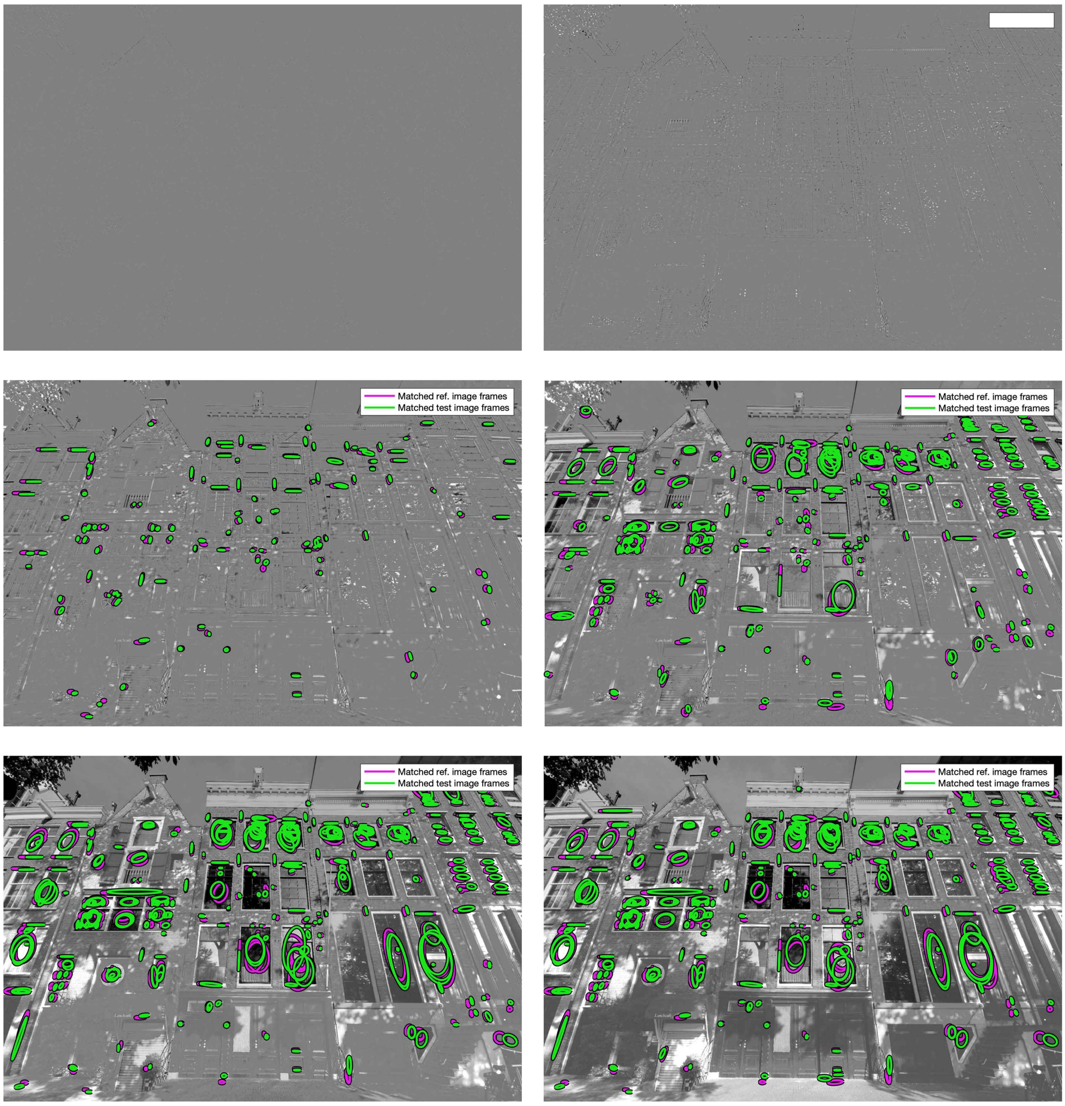


Figure C.23: MSER correspondences on a sample image taken during the day processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, MSER, high-pass sieve, Self Quotient Image

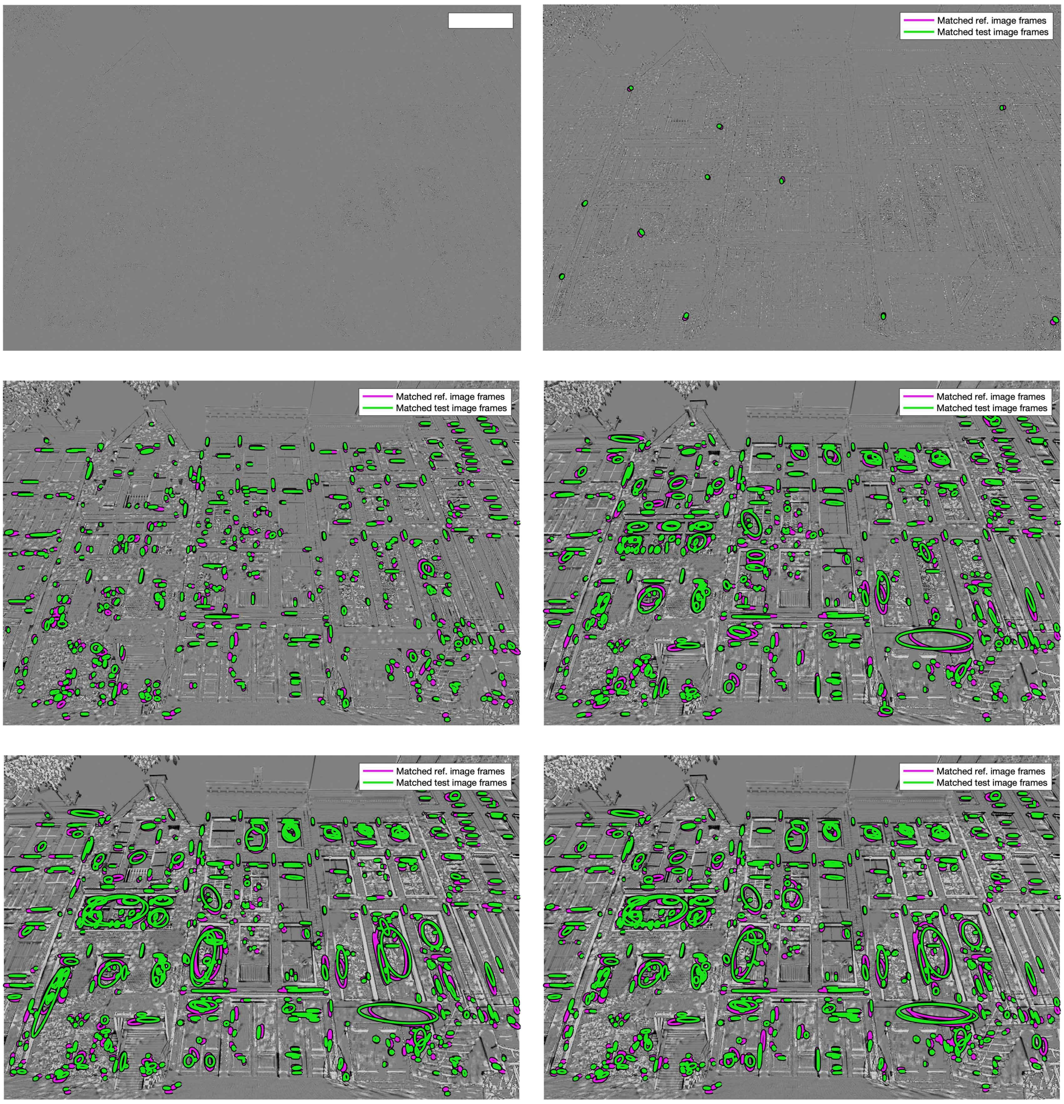


Figure C.24: MSER correspondences on a sample Self Quotient Image taken during the day processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, MSER, band-pass sieve, original image

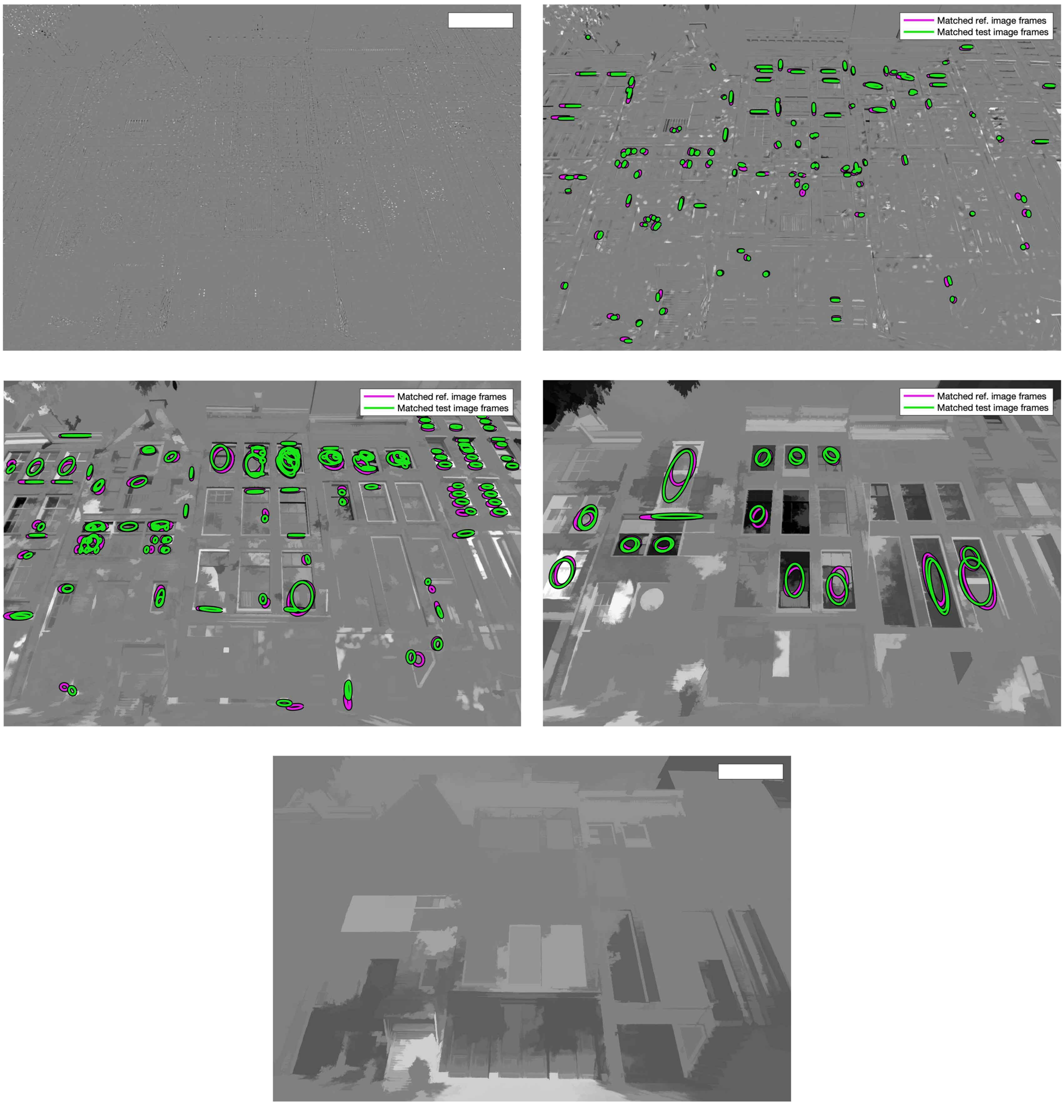

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.25: MSER correspondences on a sample image taken during the day processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Day, MSER, band-pass sieve, Self Quotient Image

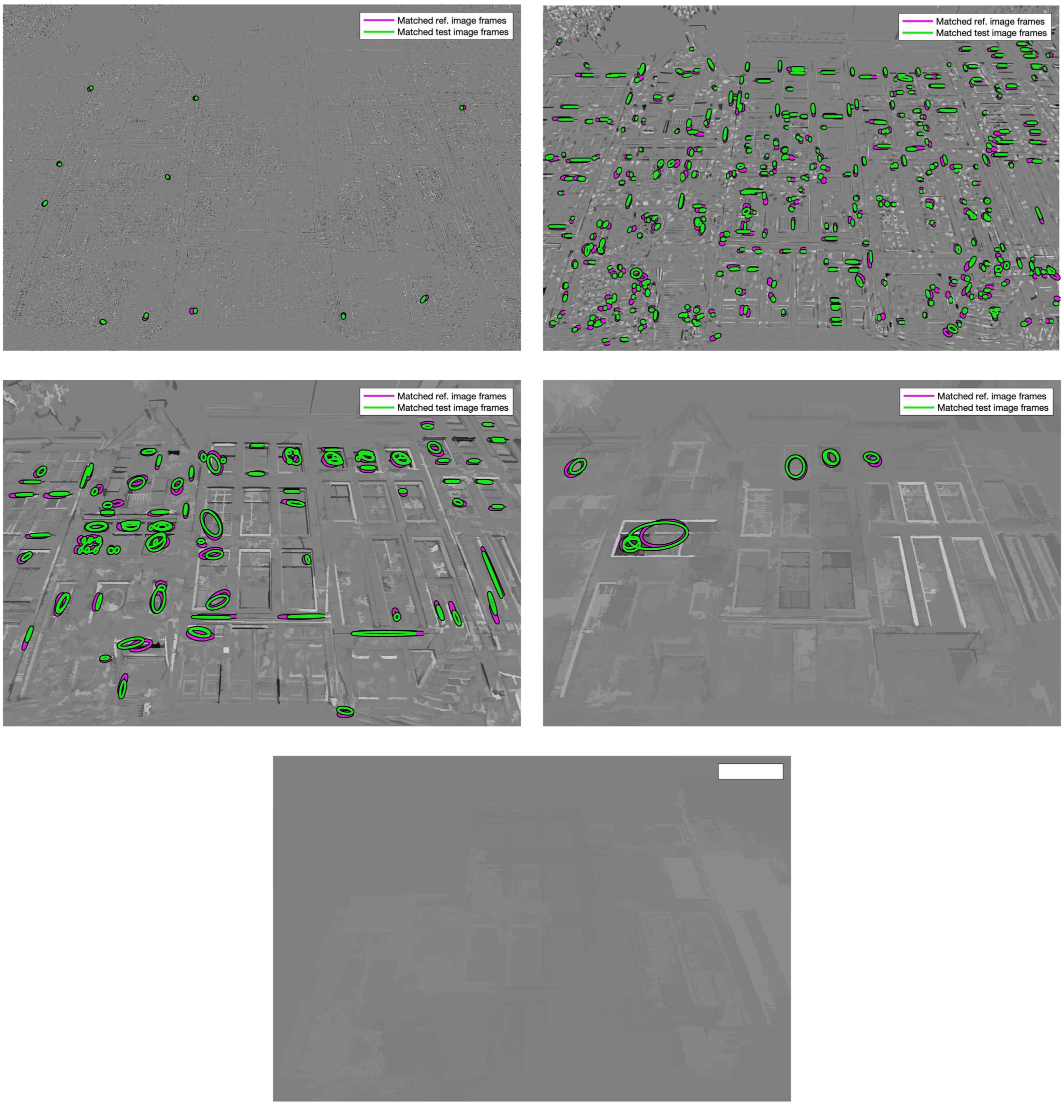


Figure C.26: MSER correspondences on a sample Self Quotient Image taken during the day processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Day, SIFT-affine, low-pass sieve, original image

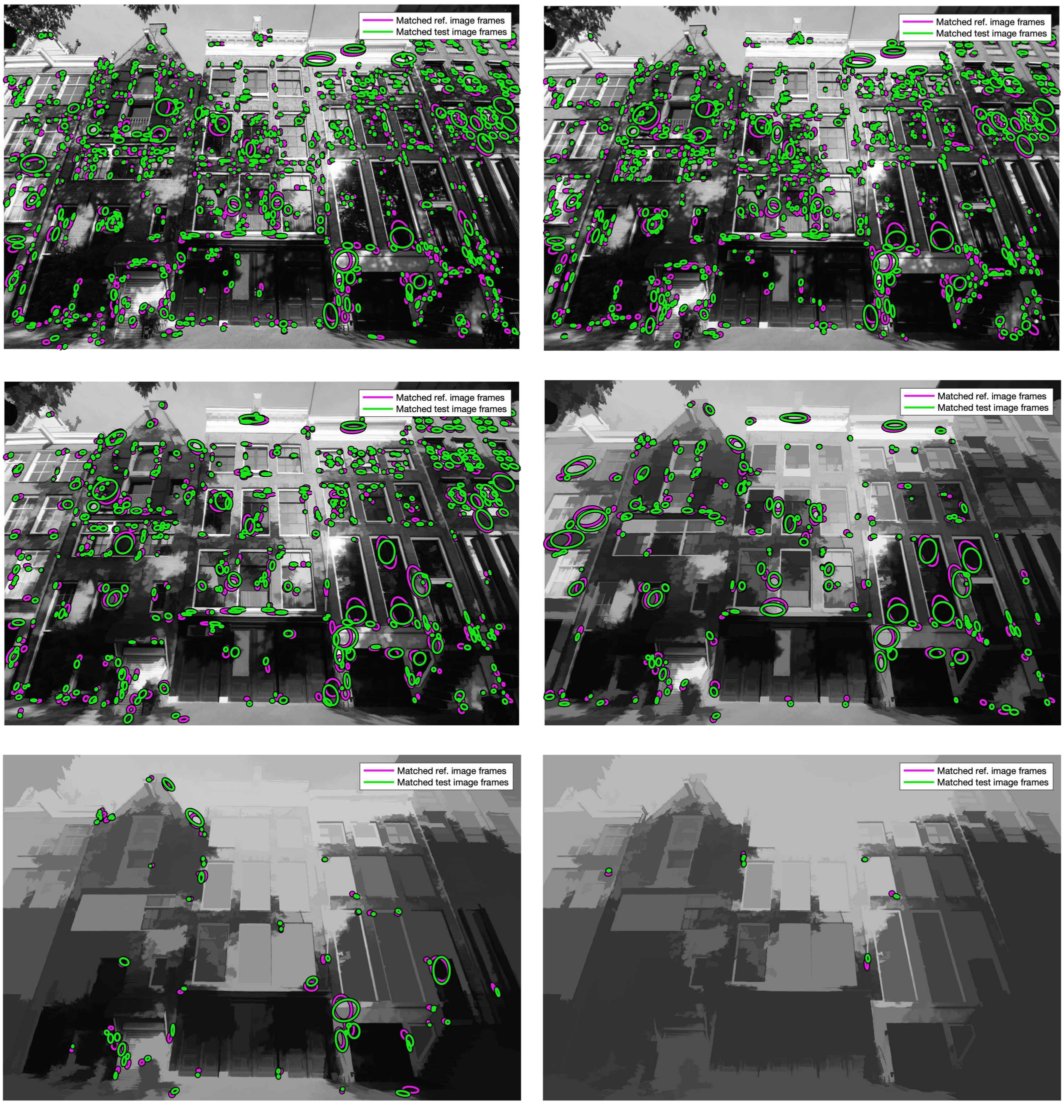


Figure C.27: SIFT-affine correspondences on a sample image taken during the day processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \dots 5$ (shown left-to-right).

Day, SIFT-affine, low-pass sieve, Self Quotient Image

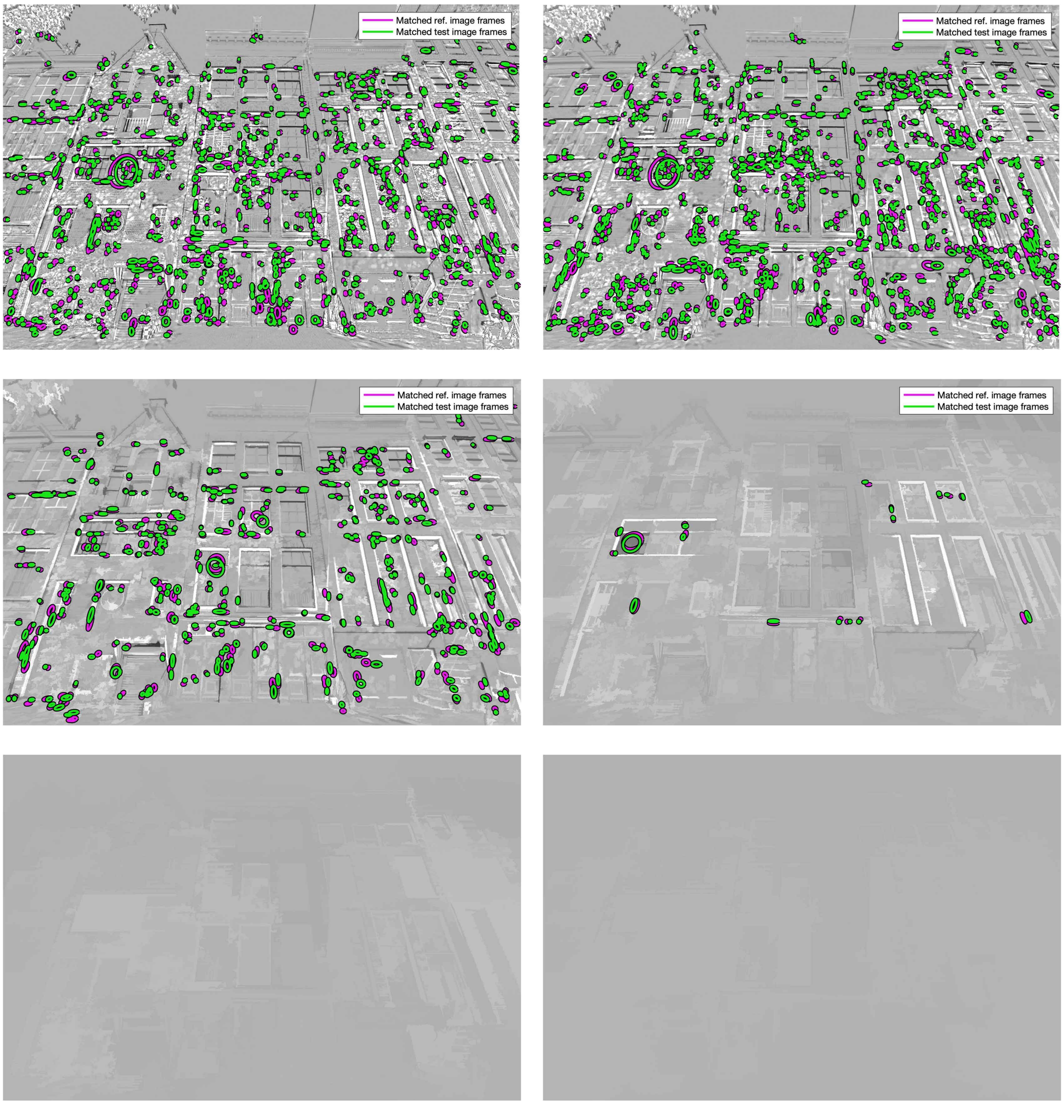


Figure C.28: SIFT-affine correspondences on a sample Self Quotient Image taken during the day processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, SIFT-affine, high-pass sieve, original image

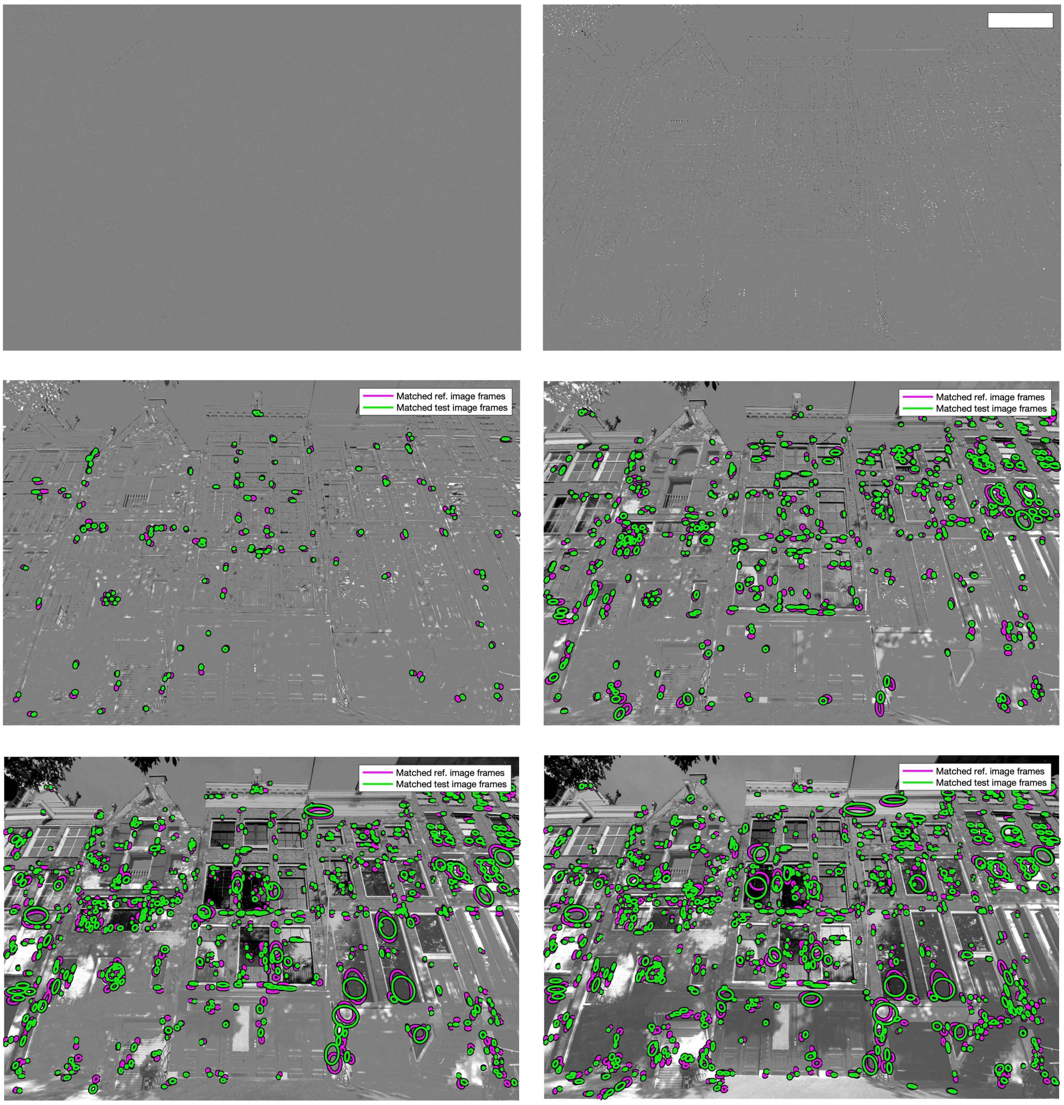

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.29: SIFT-affine correspondences on a sample image taken during the day processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \dots 5$ (shown left-to-right).

Day, SIFT-affine, high-pass sieve, Self Quotient Image

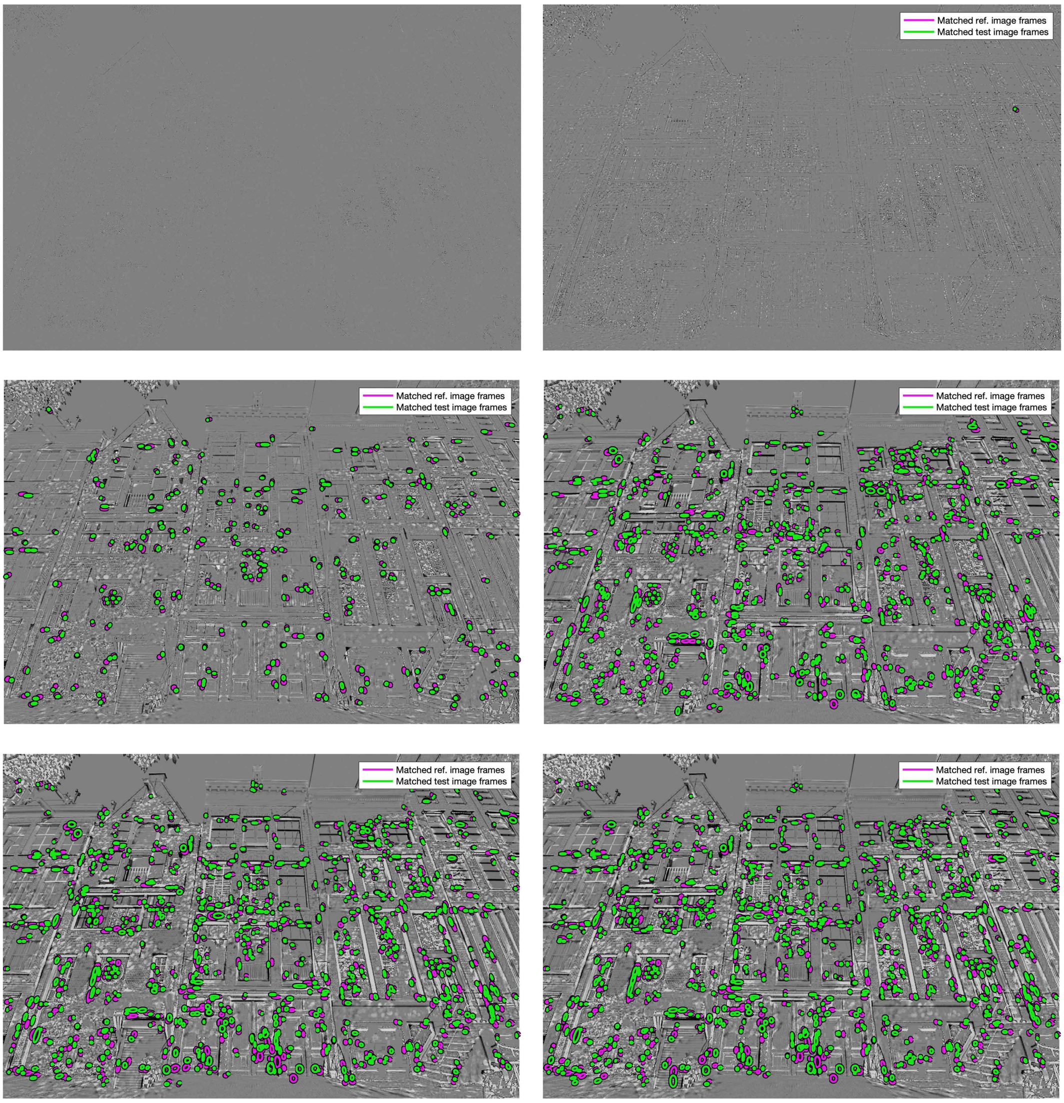


Figure C.30: SIFT-affine correspondences on a sample Self Quotient Image taken during the day processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, SIFT-affine, band-pass sieve, original image

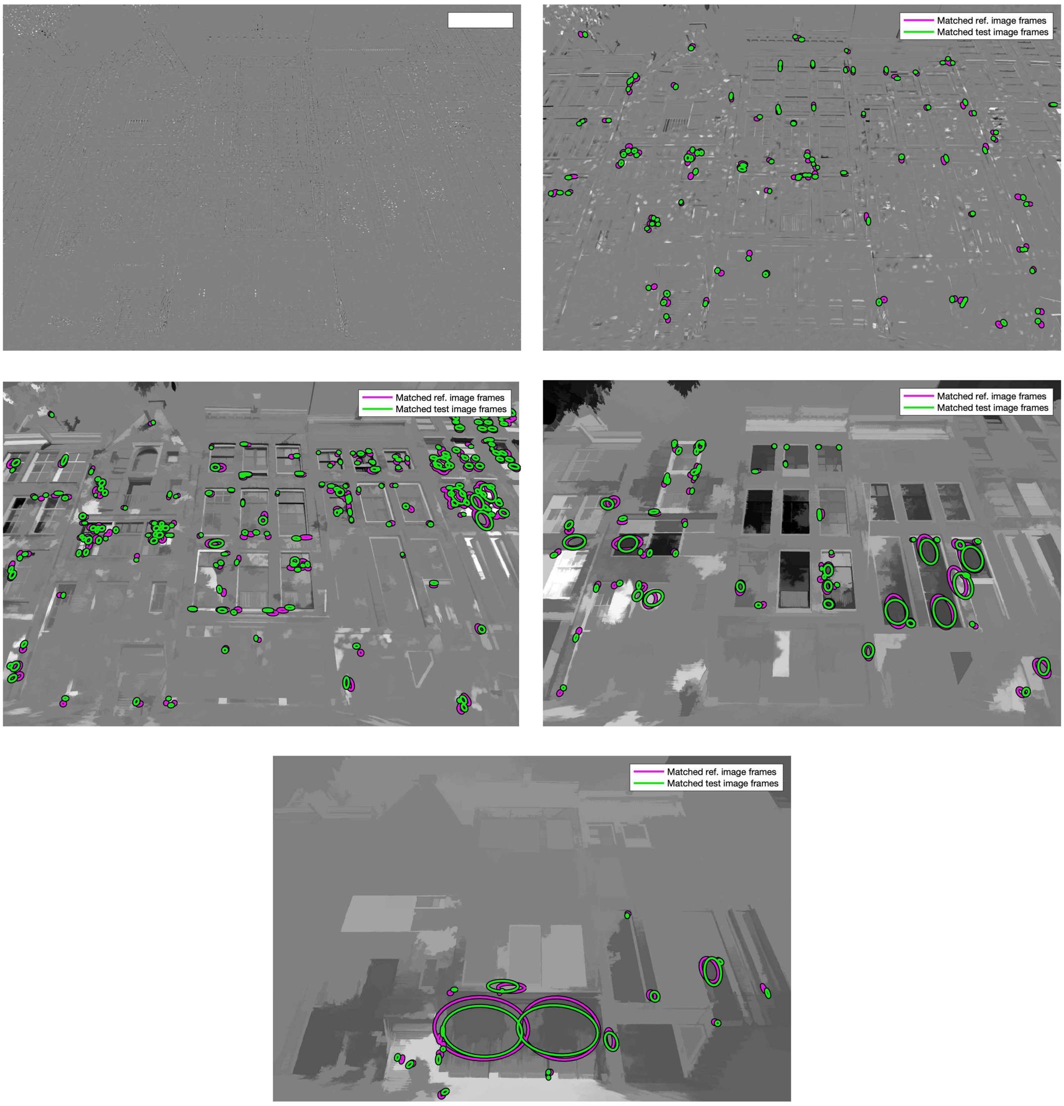


Figure C.31: SIFT-affine correspondences on a sample image taken during the day processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Day, SIFT-affine, band-pass sieve, Self Quotient Image

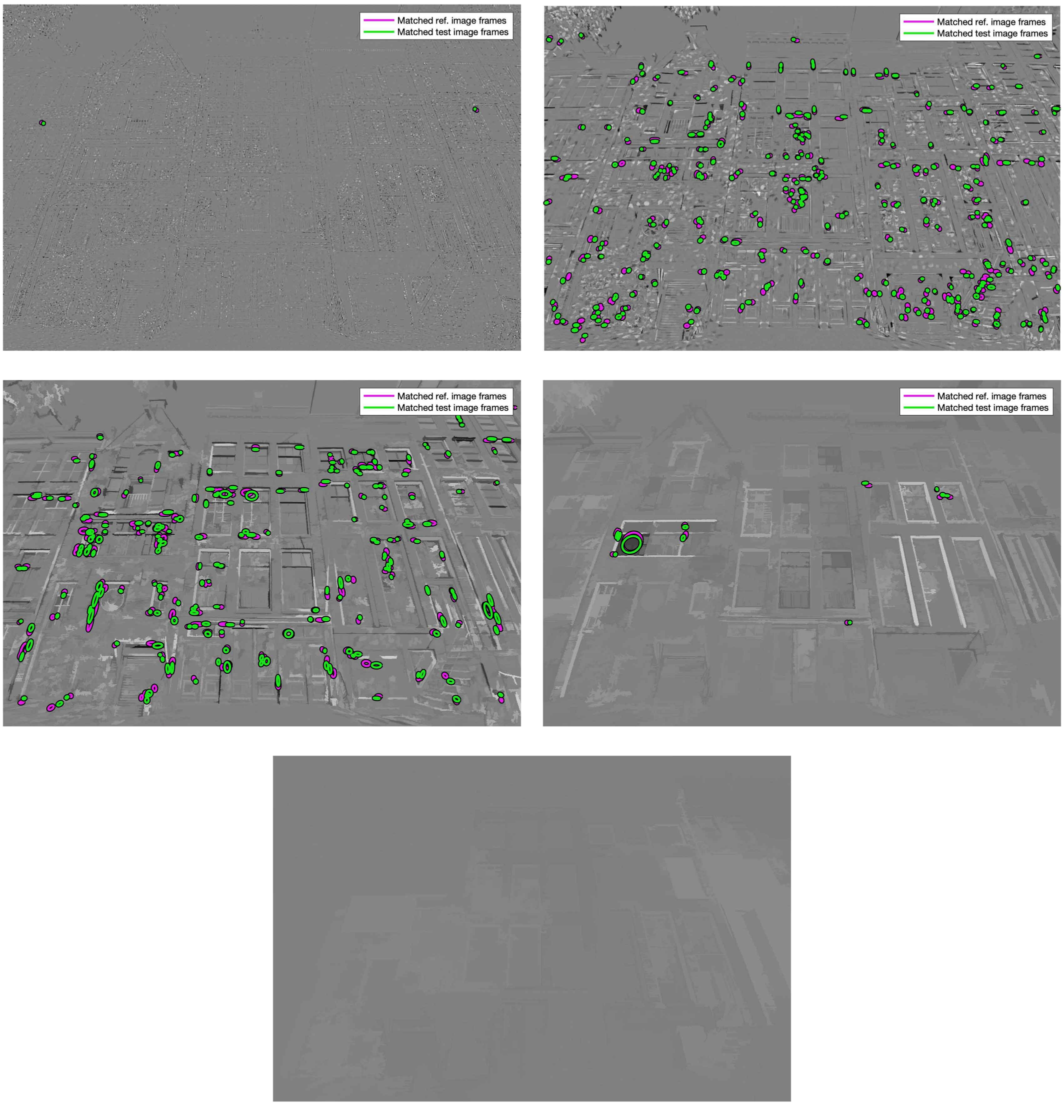

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.32: SIFT-affine correspondences on a sample Self Quotient Image taken during the day processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Day, SIFT, low-pass sieve, original image

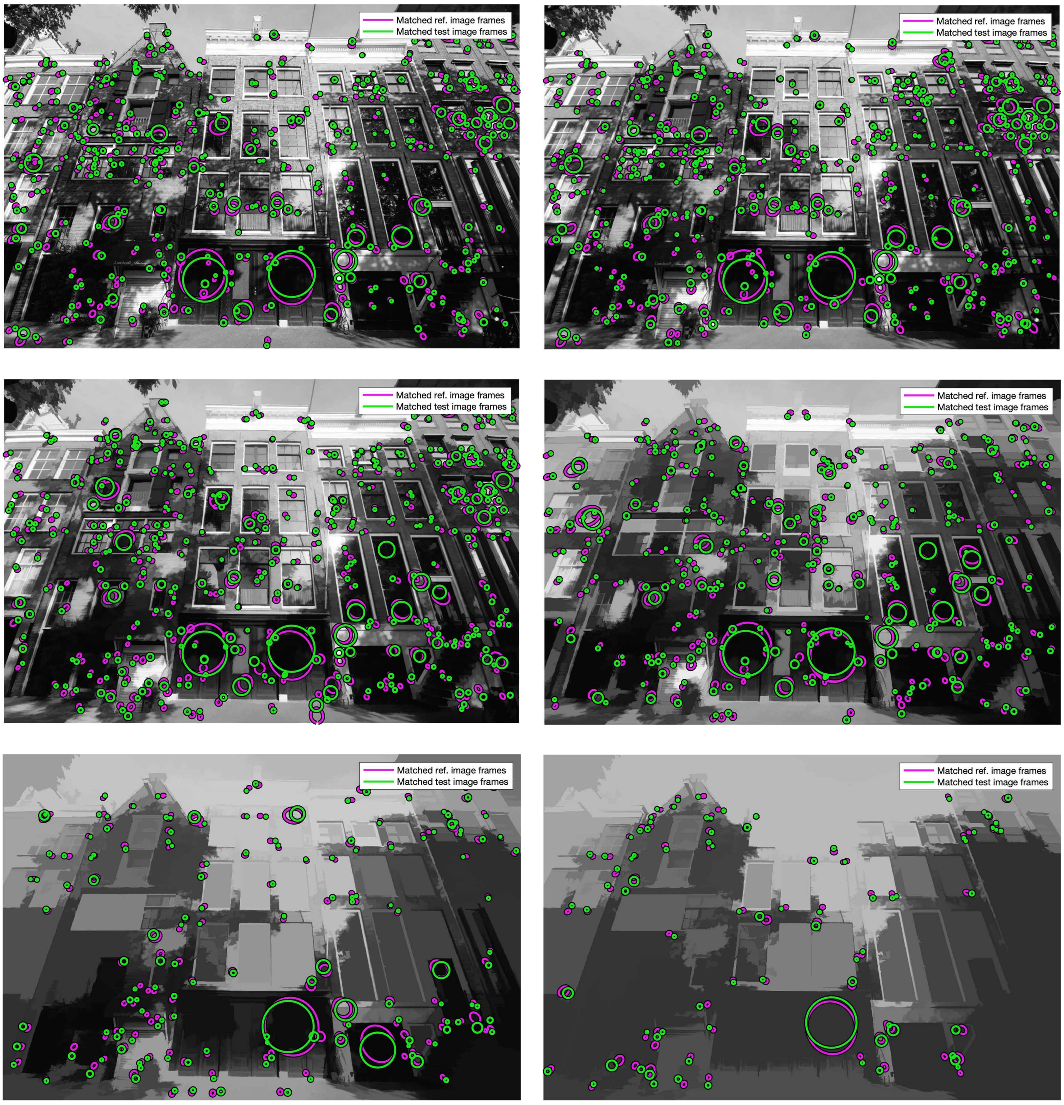

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.33: SIFT correspondences on a sample image taken during the day processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \dots 5$ (shown left-to-right).

Day, SIFT, low-pass sieve, Self Quotient Image

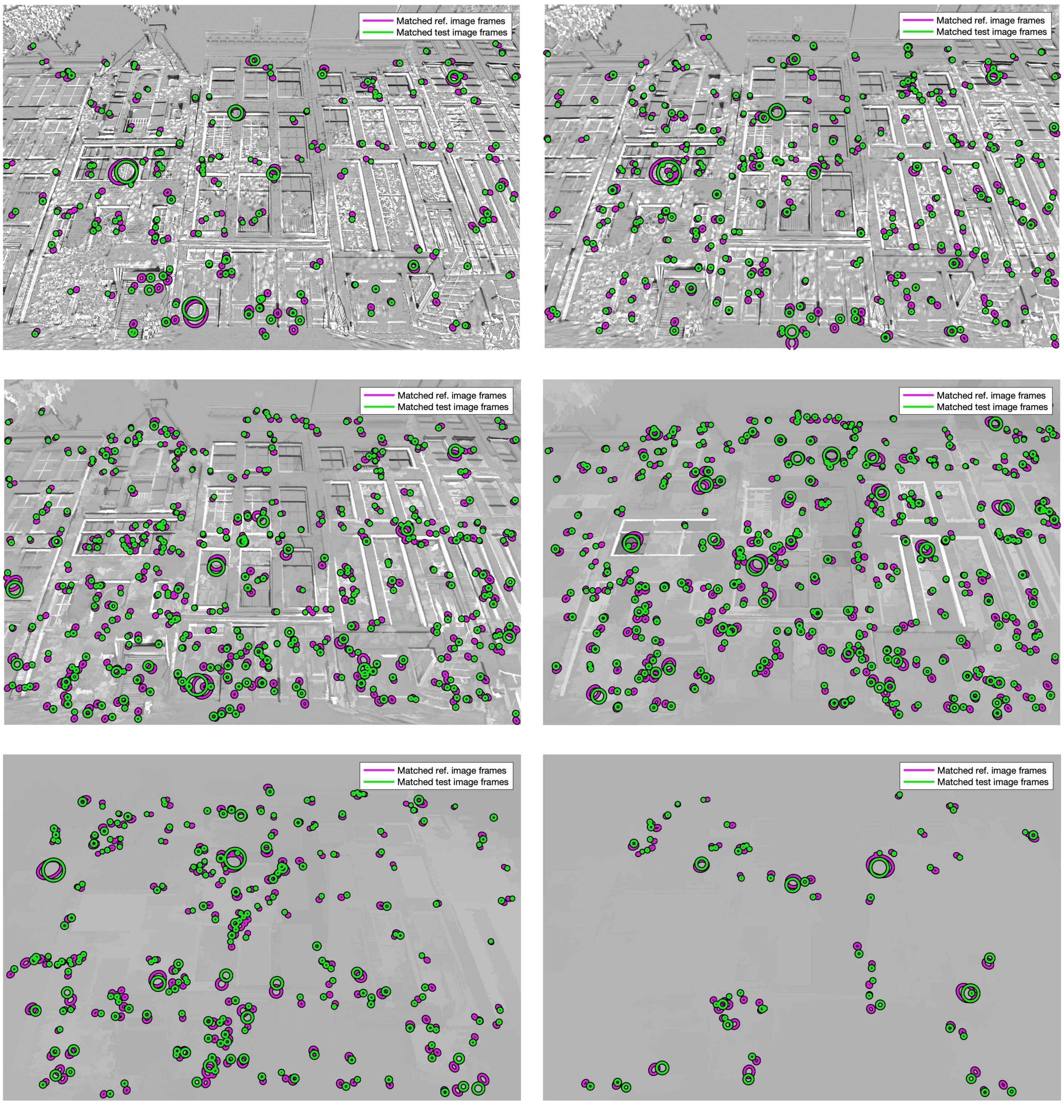


Figure C.34: SIFT correspondences on a sample Self Quotient Image taken during the day processed with a low-pass sieve. The image is sieved at scales $10^n, n = 0 \dots 5$ (shown left-to-right).

Day, SIFT, high-pass sieve, original image

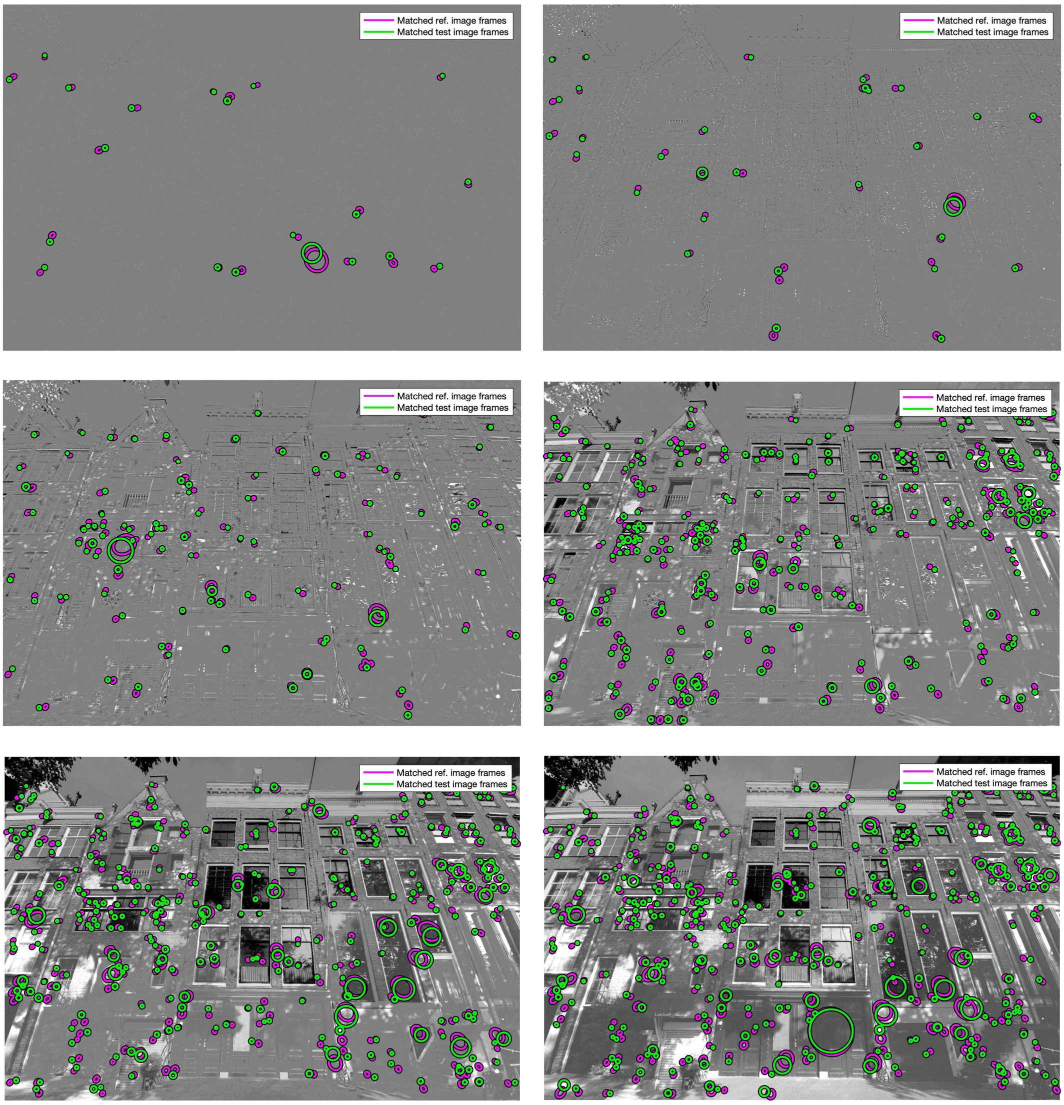


Figure C.35: SIFT correspondences on a sample image taken during the day processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, SIFT, high-pass sieve, Self Quotient Image

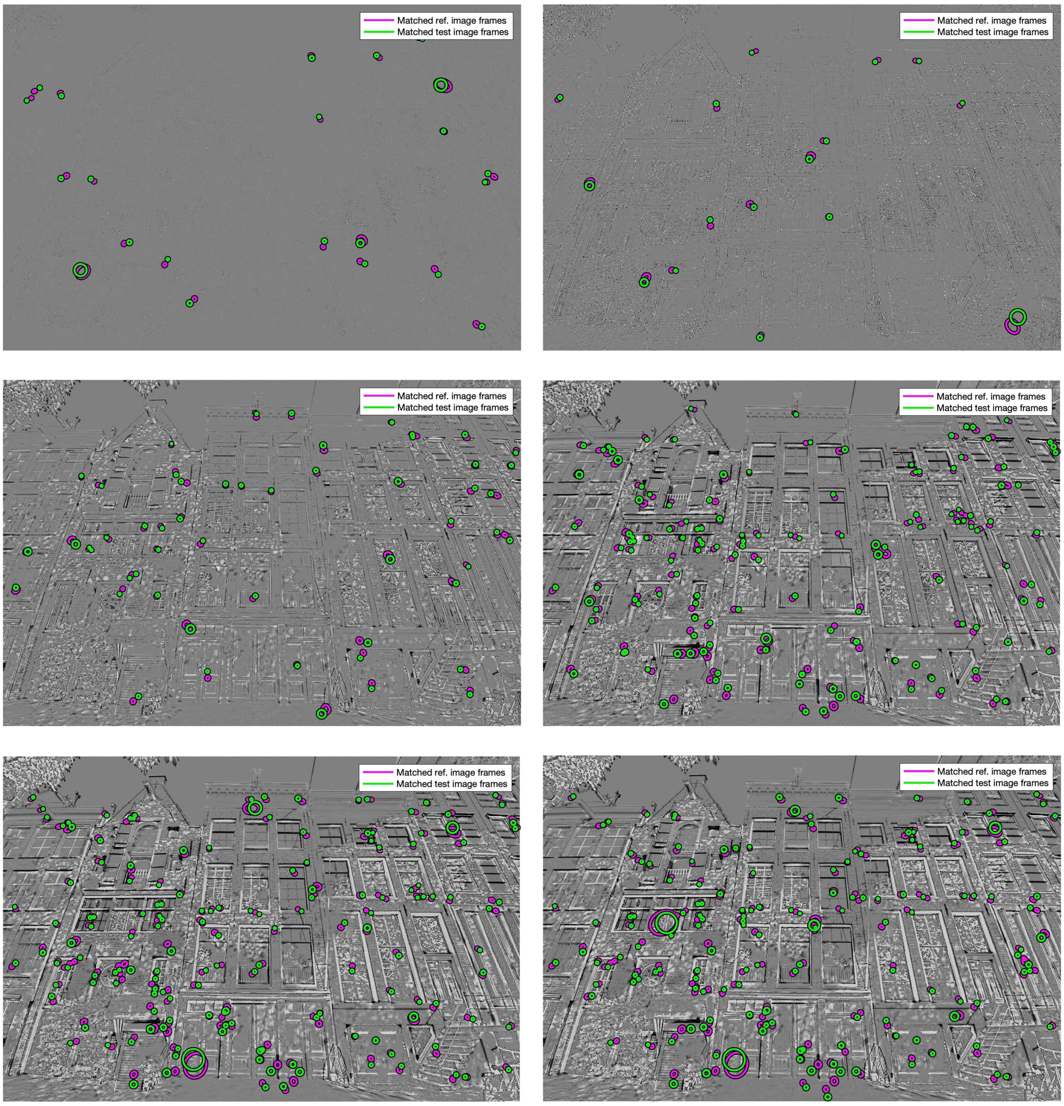

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.36: SIFT correspondences on a sample Self Quotient Image taken during the day processed with a high-pass sieve. The image is sieved at scales $10^n, n = 0 \ldots 5$ (shown left-to-right).

Day, SIFT, band-pass sieve, original image

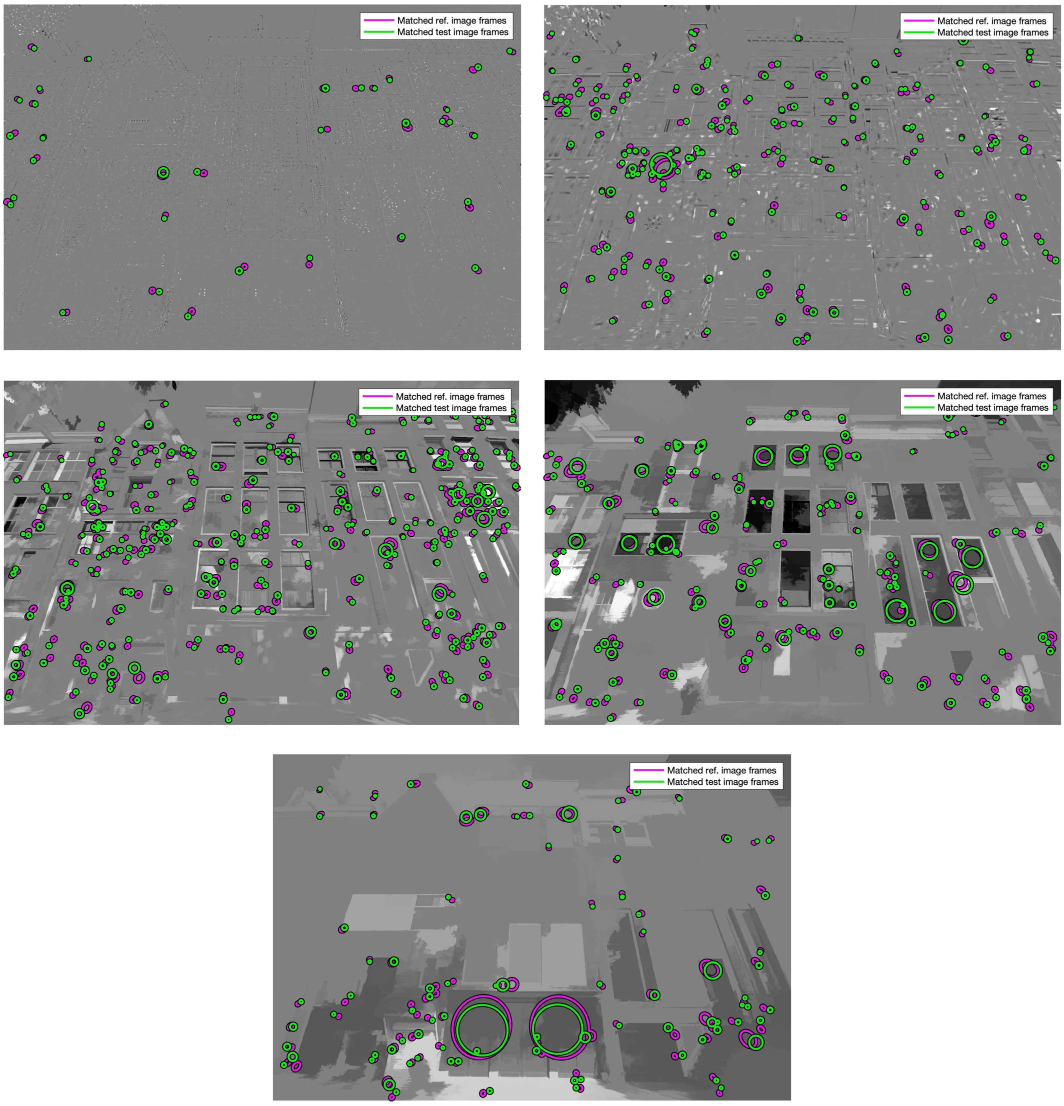

Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames
Matched ref. image frames
Matched test image frames

Figure C.37: SIFT correspondences on a sample image taken during the day processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Day, SIFT, band-pass sieve, Self Quotient Image

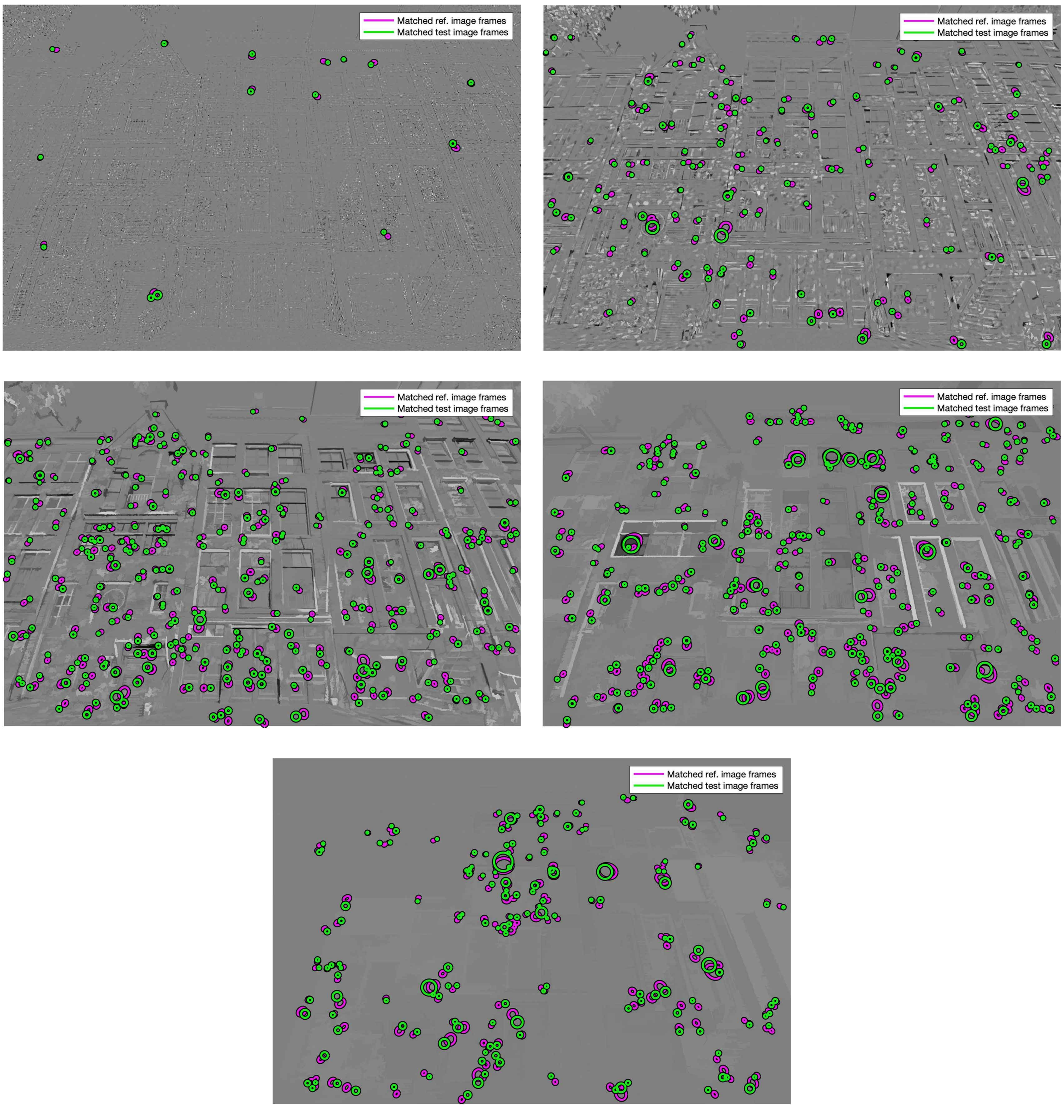


Figure C.38: SIFT correspondences on a sample Self Quotient Image taken during the day processed with a band-pass sieve. The image is sieved at bands 10^n to $10^{n+1}, n = 0 \ldots 4$ (shown left-to-right).

Bibliography

[1] Adafruit Industries LLC. ‘HC-SR04 Ultrasonic Sonar Distance Sensor.’ (2020), [Online]. Available: `https://adafruit.com/product/3942` (visited on 22/06/2020) (cit. on p. 43).

[2] E. H. Adelson, ‘Lightness Perception and Lightness Illusions,’ in *The New Cognitive Neurosciences*, M. S. Gazzaniga, Ed., 2nd ed., Cambridge, MA: MIT Press, 2000, pp. 339–351 (cit. on pp. 157–159).

[3] E. H. Adelson. ‘Checker shadow illusion,’ Wikimedia Commons. (2018), [Online]. Available: `https://commons.wikimedia.org/wiki/File:Checker_shadow_illusion.svg` (visited on 27/03/2021) (cit. on p. 160).

[4] A. Agrawal, P. Sonkar, M. Kumar and A. Kaushal, ‘GPS and GSM Based Guidance System for Blinds,’ *International Journal for Innovative Research in Science & Technology*, vol. 3, no. 12, pp. 174–178, 2017 (cit. on p. 36).

[5] D. Ahmetovic, C. Gleason, C. Ruan, K. Kitani, H. Takagi and C. Asakawa, ‘NavCog: A Navigational Cognitive Assistant for the Blind,’ in *International Conference on Human-Computer Interaction with Mobile Devices and Services*, New York, NY, US: ACM, 2016, pp. 90–99. DOI: `10.1145/2935334.2935361` (cit. on p. 36).

[6] A. Aladrén, G. López-Nicolás, L. Puig and J. J. Guerrero, ‘Navigation Assistance for the Visually Impaired Using RGB-D Sensor With Range Expansion,’ *IEEE Systems*, vol. 10, no. 3, pp. 922–932, 2016. DOI: `10.1109/JSYST.2014.2320639` (cit. on pp. 23, 36, 42, 99).

[7] P. F. Alcantarilla, O. Stasse, S. Druon, L. M. Bergasa and F. Dellaert, 'How to localize humanoids with a single camera?' *Autonomous Robots*, vol. 34, pp. 47–71, 2013. DOI: 10.1007/s10514-012-9312-1 (cit. on p. 26).

[8] J. M. Álvarez and A. M. López, 'Road Detection Based on Illuminant Invariance,' *Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 184–193, Mar. 2011. DOI: 10.1109/TITS.2010.2076349 (cit. on p. 100).

[9] American Printing House for the Blind. 'Nearby Explorer.' (2013), [Online]. Available: https://tech.aph.org/neandroid/ (visited on 12/07/2019) (cit. on pp. 37, 41).

[10] D. Anguelov et al., 'Google Street View: Capturing the World at Street Level,' *IEEE Computer*, vol. 43, no. 6, pp. 32–38, Jun. 2010. DOI: 10.1109/MC.2010.170 (cit. on pp. 77, 83, 93).

[11] Apple Inc. 'Apple rolls out all-new Maps across the United Kingdom and Republic of Ireland,' Newsroom. (2020), [Online]. Available: https://apple.com/uk/newsroom/2020/10/apple-rolls-out-all-new-maps-across-the-united-kingdom-and-republic-of-ireland/ (visited on 08/05/2021) (cit. on p. 68).

[12] Apple Inc. 'MapKit,' Apple Developer. (2021), [Online]. Available: https://developer.apple.com/documentation/mapkit (visited on 18/10/2021) (cit. on pp. 64, 69).

[13] M. Avila, K. Wolf, A. Brock and N. Henze, 'Remote Assistance for Blind Users in Daily Life,' in *Conference on Pervasive Technologies Related to Assistive Environments*, New York, NY, US: ACM, 29th Jun. 2016. DOI: 10.1145/2910674.2935839 (cit. on p. 40).

[14] J. A. Bala, S. A. Adeshina and A. M. Aibinu, 'Advances in Visual Simultaneous Localisation and Mapping Techniques for Autonomous

Vehicles: A Review,' *IEEE Sensors*, vol. 22, no. 22, p. 8943, 18th Nov. 2022. DOI: 10.3390/s22228943 (cit. on p. 25).

[15] J. A. Bangham, R. Harvey, P. D. Ling and R. V. Aldridge, 'Morphological scale-space preserving transforms in many dimensions,' *Journal of Electronic Imaging*, vol. 5, no. 3, p. 283, 1st Jul. 1996. DOI: 10.1117/12.243349 (cit. on pp. 127, 167, 173).

[16] J. Bangham, P. Chardaire, C. Pye and P. Ling, 'Multiscale nonlinear decomposition: The sieve decomposition theorem,' *Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 529–539, May 1996. DOI: 10.1109/34.494642 (cit. on pp. 127, 167).

[17] J. Bangham, P. Ling and R. Young, 'Multiscale recursive medians, scale-space, and transforms with applications to image processing,' *Transactions on Image Processing*, vol. 5, no. 6, pp. 1043–1048, Jun. 1996. DOI: 10.1109/83.503918 (cit. on p. 173).

[18] J. Bangham, 'Data-sieving hydrophobicity plots,' *Analytical Biochemistry*, vol. 174, no. 1, pp. 142–145, Oct. 1988. DOI: 10.1016/0003-2697(88)90528-3 (cit. on pp. 127, 167).

[19] H. G. Barrow and J. M. Tenenbaum, 'Recovering intrinsic scene characteristics from images,' *Computer Vision Systems*, pp. 3–26, 1978 (cit. on p. 159).

[20] BlindSquare. 'BlindSquare – Pioneering accessible navigation.' (2012), [Online]. Available: http://blindsquare.com/ (visited on 13/07/2019) (cit. on pp. 35, 41).

[21] G. Bresson, R. Aufrere and R. Chapuis, 'Making visual SLAM consistent with geo-referenced landmarks,' in *Intelligent Vehicles Symposium*, Gold Coast City, Australia: IEEE, Jun. 2013, pp. 553–558. DOI: 10.1109/IVS.2013.6629525 (cit. on p. 25).

[22] D. A. Brusnighan, M. G. Strauss, J. M. Floyd and B. C. Wheeler, ‘Orientation aid implementing the global positioning system,’ in *Northeast Bioengineering Conference*, IEEE, 1989, pp. 33–34. DOI: 10.1109/NEBC.1989.36684 (cit. on pp. 37, 41).

[23] A. Budrionis, D. Plikynas, P. Daniušis and A. Indrulionis, ‘Smartphone-based computer vision travelling aids for blind and visually impaired individuals: A systematic review,’ *Assistive Technology*, vol. 34, no. 2, pp. 178–194, 17th Apr. 2020. DOI: 10.1080/10400435.2020.1743381 (cit. on p. 68).

[24] R. Busatto and R. Harvey, ‘Outdoor Navigation Assistants for Visually Impaired Persons: Problems and Challenges,’ *Journal on Technology and Persons with Disabilities*, vol. 10, pp. 184–205, 2022 (cit. on pp. 29, 30, 62).

[25] J. Butler, *Frames of War: When Is Life Grievable?* Verso, 2009 (cit. on p. 32).

[26] E. Carrasco et al., ‘Autonomous Navigation based on Binaural Guidance for People with Visual Impairment,’ in *Assistive Technology: From Research to Practice*, ser. Assistive Technology Research Series, vol. 33, Vilamoura, Portugal: IOS Press, Sep. 2013, pp. 690–694 (cit. on p. 66).

[27] E. Carrasco et al., ‘ARGUS Autonomous Navigation System for People with Visual Impairments,’ presented at the Computers Helping People with Special Needs, ser. Lecture Notes in Computer Science, vol. 8548, Cham: Springer, 2014, pp. 100–107. DOI: 10.1007/978-3-319-08599-9_16 (cit. on pp. 39, 41).

[28] K. Chaccour and G. Badr, ‘Novel indoor navigation system for visually impaired and blind people,’ in *International Conference on Applied Research in Computer Science and Engineering*, IEEE, Oct. 2015, pp. 1–5. DOI: 10.1109/ARCSE.2015.7338143 (cit. on p. 37).

[29] S. A. Cheraghi, G. Fusco and J. M. Coughlan, 'Real-Time Sign Detection for Accessible Indoor Navigation,' *Journal on Technology & Persons with Disabilities*, vol. 9, pp. 125–139, 2021. DOI: 10211.3/219939 (cit. on p. 69).

[30] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou, 'Developing a context-aware electronic tourist guide: Some issues and experiences,' in *SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, US: ACM, 2000, pp. 17–24. DOI: 10.1145/332040.332047 (cit. on p. 41).

[31] L. G. Ciaffoni. 'Ariadne GPS.' (2011), [Online]. Available: http://ariadnegps.eu (visited on 13/07/2019) (cit. on pp. 38, 41).

[32] N. Dalal and B. Triggs, 'Histograms of Oriented Gradients for Human Detection,' in *Conference on Computer Vision and Pattern Recognition*, vol. 1, San Diego, CA, USA: IEEE, 2005, pp. 886–893. DOI: 10.1109/CVPR.2005.177 (cit. on p. 48).

[33] D. DeTone, T. Malisiewicz and A. Rabinovich, 'SuperPoint: Self-Supervised Interest Point Detection and Description,' in *Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 337–33 712. DOI: 10.1109/CVPRW.2018.00060 (cit. on pp. 48, 50, 73, 75, 155).

[34] P. Dharani, B. Lipson, D. Thomas and E. Agu, 'RFID Navigation System for the Visually Impaired,' Worcester Polytechnic Institute, 2012 (cit. on p. 39).

[35] D. M. Dockery and M. G. Krzystolik, 'The Use of Mobile Applications as Low-Vision Aids: A Pilot Study,' *Rhode Island Medical Journal*, vol. 103, no. 8, pp. 69–72, 1st Oct. 2020 (cit. on p. 40).

[36] H. Dong, J. Schafer, Y. Tao and A. Ganz, 'PERCEPT-V: Integrated Indoor Navigation System for the Visually Impaired Using Vision-Based Localization and Waypoint-Based Instructions,' *Journal on Technology*

and Persons with Disabilities, vol. 8, pp. 1–21, 2020. DOI: 10211.3/215976 (cit. on p. 70).

[37] Dreamwaves. 'WaveOut app.' (2021), [Online]. Available: https://dreamwaves.io (visited on 30/03/2022) (cit. on pp. 35, 41).

[38] R. Espinoza and M. González. 'Lazarillo.' (2016), [Online]. Available: https://lazarillo.cl/en/ (visited on 13/07/2019) (cit. on pp. 38, 41).

[39] G. I. Evenden et al., *PROJ coordinate transformation software library*, version 9.1.0, Open Source Geospatial Foundation, 2023. DOI: 10.5281/ZENODO.5884394 (cit. on p. 84).

[40] EveryWare Technologies. 'iMove.' (2013), [Online]. Available: http://everywaretechnologies.com/apps/imove (visited on 13/07/2019) (cit. on pp. 38, 41).

[41] G. D. Finlayson, 'Colour and illumination in computer vision,' *Interface Focus*, vol. 8, no. 4, 2018. DOI: 10.1098/rsfs.2018.0008 (cit. on pp. 25, 100).

[42] G. D. Finlayson, S. D. Hordley, Cheng Lu and M. S. Drew, 'On the removal of shadows from images,' *Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, Jan. 2006. DOI: 10.1109/TPAMI.2006.18 (cit. on p. 164).

[43] M. A. Fischler and R. C. Bolles, 'Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,' *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1st Jun. 1981. DOI: 10.1145/358669.358692 (cit. on pp. 75, 76, 104).

[44] Flickr Inc. 'Flickr.' (2004), [Online]. Available: https://flickr.com (visited on 01/03/2021) (cit. on p. 53).

[45] Float. 'Cydalion app.' (2016), [Online]. Available: http://cydalion.com/ (visited on 13/07/2019) (cit. on pp. 36, 42).

[46] J. Fuentes-Pacheco, J. Ruiz-Ascencio and J. M. Rendón-Mancha, 'Visual simultaneous localization and mapping: A survey,' *Artificial Intelligence Review*, vol. 43, pp. 55–81, 2015. DOI: `10.1007/s10462-012-9365-8` (cit. on p. 26).

[47] G. Fusco and J. M. Coughlan, 'Indoor localization for visually impaired travelers using computer vision on a smartphone,' in *Proceedings of the International Web for All Conference*, New York, NY, US: ACM, 20th Apr. 2020, pp. 1–11. DOI: `10.1145/3371300.3383345` (cit. on p. 37).

[48] Garmin. 'NAVIGON.' (2011), [Online]. Available: `https://navigon.com/en/mobile-app/` (visited on 13/07/2019) (cit. on pp. 38, 41).

[49] S. Gibson, R. Harvey and G. Finlayson, 'Convex Colour Sieves,' in *International Conference on Scale-Space Theories in Computer Vision*, vol. 2695, Berlin, Germany: Springer, 2003, pp. 550–563. DOI: `10.1007/3-540-44935-3_38` (cit. on pp. 127, 167).

[50] S. E. Gibson, 'Sieves for Image Retrieval,' Ph.D. dissertation, University of East Anglia, 11th Jan. 2004, 247 pp. (cit. on pp. 127, 167, 169).

[51] Google Inc. 'Street View Static API overview.' (2007), [Online]. Available: `https://developers.google.com/maps/documentation/streetview/overview` (visited on 02/03/2021) (cit. on pp. 106, 111).

[52] Google Inc. 'Redefining what a map can be with new information and AI,' The Keyword. (2021), [Online]. Available: `https://blog.google/products/maps/redefining-what-map-can-be-new-information-and-ai` (visited on 08/05/2021) (cit. on pp. 64, 68).

[53] Google Inc. 'The Directions API overview,' Google Maps Platform. (2021), [Online]. Available: `https://developers.google.com/maps/documentation/directions/overview` (visited on 18/10/2021) (cit. on p. 69).

[54] C. Granquist, S. Y. Sun, S. R. Montezuma, T. M. Tran, R. Gage and G. E. Legge, 'Evaluation and Comparison of Artificial Intelligence Vision Aids: Orcam MyEye 1 and Seeing AI,' *Journal of Visual Impairment and Blindness*, vol. 115, no. 4, pp. 277–285, 2021. DOI: 10.1177/0145482X211027492 (cit. on p. 69).

[55] N. Griffin-Shirley et al., 'A Survey on the Use of Mobile Applications for People who Are Visually Impaired,' *Journal of Visual Impairment & Blindness*, vol. 111, no. 4, pp. 307–323, 13th Jul. 2017. DOI: 10.1177/0145482X1711100402 (cit. on p. 23).

[56] R. Grosse, M. K. Johnson, E. H. Adelson and W. T. Freeman, 'Ground truth dataset and baseline evaluations for intrinsic image algorithms,' in *International Conference on Computer Vision*, Kyoto, Japan: IEEE, Sep. 2009, pp. 2335–2342. DOI: 10.1109/ICCV.2009.5459428 (cit. on p. 100).

[57] J. Gu et al., 'Recent advances in convolutional neural networks,' *Pattern Recognition*, vol. 77, pp. 354–377, May 2018. DOI: 10.1016/j.patcog.2017.10.013 (cit. on p. 48).

[58] J. A. B. Guillén, 'Applying Multiscale Morphology Methods to Image Coding and Compression,' Ph.D. dissertation, University of East Anglia, 31st Jul. 2008, 185 pp. (cit. on pp. 127, 167, 170, 172, 173).

[59] R. Gulati, 'GPS Based Voice Alert System for the Blind,' *International Journal of Scientific & Engineering Research*, vol. 2, no. 1, pp. 1–5, 2011 (cit. on p. 39).

[60] Q. Guo, W. H. Deng, O. Bebek, M. C. Cavusoglu, C. H. Mastrangelo and D. J. Young, 'Personal Inertial Navigation System Assisted by MEMS Ground Reaction Sensor Array and Interface ASIC for GPS-Denied Environment,' *Journal of Solid-State Circuits*, vol. 53, no. 11, pp. 3039–3049, Nov. 2018. DOI: 10.1109/JSSC.2018.2868263 (cit. on p. 44).

[61] R. Guo, Q. Dai and D. Hoiem, ‘Single-image shadow detection and removal using paired regions,’ in *Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2033–2040. DOI: 10.1109/CVPR.2011.5995725 (cit. on pp. 164–166).

[62] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685 (cit. on pp. 76, 103–105).

[63] K. Hata and S. Savarese, ‘Course Notes,’ CS231A Computer Vision, Stanford University, 2021 (cit. on pp. 76, 103).

[64] J. Hays and A. A. Efros, ‘IM2GPS: Estimating geographic information from a single image,’ in *Conference on Computer Vision and Pattern Recognition*, vol. 05, Jun. 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587784 (cit. on p. 53).

[65] J. Hays and A. A. Efros, ‘Large-Scale Image Geolocalization,’ in *Multimodal Location Estimation of Videos and Images*, Springer, 2015, pp. 41–62. DOI: 10.1007/978-3-319-09861-6_3 (cit. on pp. 54, 99).

[66] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, ‘Transformation of GPS Results,’ in *Global Positioning System: Theory and Practice*, 5th ed. Vienna: Springer, 2001, pp. 279–307. DOI: 10.1007/978-3-7091-6199-9_10 (cit. on pp. 78, 79).

[67] A. Hub and B. Schmitz, ‘Addition of RFID-based initialization and object recognition to the navigation system TANIA,’ in *Technology and Persons with Disabilities Conference*, California State University, 2009 (cit. on p. 39).

[68] Humanware. ‘Victor Reader Trek.’ (2017), [Online]. Available: https://store.humanware.com/hus/victor-reader-trek-talking-book-player-gps.html (visited on 12/07/2019) (cit. on pp. 37, 41).

[69] T. Ifukube, T. Sasaki and C. Peng, ‘A blind mobility aid modeled after echolocation of bats,’ *Transactions on Biomedical Engineering*, vol. 38, no. 5, pp. 461–465, May 1991. DOI: `10.1109/10.81565` (cit. on p. 36).

[70] D. Jain, ‘Path-guided indoor navigation for the visually impaired using minimal building retrofitting,’ in *SIGACCESS Conference on Computers and Accessibility*, ACM, 2014, pp. 225–232. DOI: `10.1145/2661334.2661359` (cit. on p. 37).

[71] J. S. Ju, E. Ko and E. Y. Kim, ‘EYECane,’ in *SIGACCESS Conference on Computers and Accessibility*, New York, NY, US: ACM, 2009, p. 237. DOI: `10.1145/1639642.1639693` (cit. on pp. 36, 42).

[72] L. Kaminski, R. Kowalik, Z. Lubniewski and A. Stepnowski, ‘VOICE MAPS - portable, dedicated GIS for supporting the street navigation and self-dependent movement of the blind,’ in *International Conference on Information Technology*, IEEE, 2010, pp. 153–156 (cit. on pp. 37, 41).

[73] S. Kammoun et al., ‘Navigation and space perception assistance for the visually impaired: The NAVIG project,’ *Innovation and Research in BioMedical engineering*, vol. 33, no. 2, pp. 182–189, Apr. 2012. DOI: `10.1016/j.irbm.2012.01.009` (cit. on pp. 35, 41, 42).

[74] S. Kanuganti. ‘Introduction to the Aira Explorer Program.’ (2015), [Online]. Available: `https://aira.io/explorer-learn-more` (visited on 10/07/2019) (cit. on pp. 35, 40–42).

[75] N. Kanwal, E. Bostanci, K. Currie and A. F. Clark, ‘A Navigation System for the Visually Impaired: A Fusion of Vision and Depth Sensor,’ *Applied Bionics and Biomechanics*, vol. 2015, pp. 1–16, 2015. DOI: `10.1155/2015/479857` (cit. on pp. 36, 42).

[76] B. F. G. Katz et al., ‘NAVIG: Augmented reality guidance system for the visually impaired,’ *Virtual Reality*, vol. 16, no. 4, pp. 253–269, 2012. DOI: `10.1007/s10055-012-0213-6` (cit. on pp. 35, 41, 42).

[77] I. A. Kazerouni, L. Fitzgerald, G. Dooly and D. Toal, ‘A survey of state-of-the-art on visual SLAM,’ *Expert Systems with Applications*, vol. 205, p. 117 734, Nov. 2022. DOI: 10.1016/j.eswa.2022.117734 (cit. on p. 25).

[78] S. Kirkpatrick and M. Lilburn. ‘Loadstone GPS.’ (2004), [Online]. Available: http://loadstone-gps.com (visited on 13/07/2019) (cit. on pp. 38, 41).

[79] L. Kneip, D. Scaramuzza and R. Siegwart, ‘A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation,’ in *Conference on Computer Vision and Pattern Recognition*, Jun. 2011, pp. 2969–2976. DOI: 10.1109/CVPR.2011.5995464 (cit. on pp. 50, 75).

[80] S. Koley and R. Mishra, ‘Voice Operated Outdoor Navigation System for Visually Impaired Persons,’ *International Journal of Engineering Trends and Technology*, vol. 3, no. 2, pp. 153–157, 2012 (cit. on pp. 35, 41).

[81] V. Kulyukin, C. Gharpure, J. Nicholson and S. Pavithran, ‘RFID in robot-assisted indoor navigation for the visually impaired,’ in *International Conference on Intelligent Robots and Systems*, vol. 2, IEEE, 2005, pp. 1979–1984. DOI: 10.1109/IROS.2004.1389688 (cit. on p. 36).

[82] C. K. Lakde and P. S. Prasad, ‘Review Paper on Navigation System for Visually Impaired People,’ *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 1, pp. 166–168, 30th Jan. 2015. DOI: 10.17148/IJARCCE.2015.4134 (cit. on p. 23).

[83] A. Lakehal, S. Lepreux, C. Efstratiou, C. Kolski and P. Nicolaou, ‘Investigating Smartphones and AR Glasses for Pedestrian Navigation and their Effects in Spatial Knowledge Acquisition,’ in *International Conference on Human-Computer Interaction with Mobile Devices and*

Services, New York, NY, US: ACM, 5th Oct. 2020, pp. 1–7. DOI: 10.1145/3406324.3410722 (cit. on pp. 39, 41, 45).

[84] E. H. Land and J. J. McCann, 'Lightness and Retinex Theory,' *Journal of the Optical Society of America*, vol. 61, no. 1, pp. 1–11, 1st Jan. 1971. DOI: 10.1364/JOSA.61.000001 (cit. on pp. 121, 160).

[85] B. Li, J. P. Muñoz, X. Rong, J. Xiao, Y. Tian and A. Arditi, 'ISANA: Wearable Context-Aware Indoor Assistive Navigation with Obstacle Avoidance for the Blind,' in *European Conference on Computer Vision*, G. Hua and H. Jégou, Eds., vol. 9914, Amsterdam, The Netherlands: Springer, 2016, pp. 448–462. DOI: 10.1007/978-3-319-48881-3 (cit. on pp. 35, 42).

[86] K.-C. Liu, C.-H. Wu, S.-Y. Tseng and Y.-T. Tsai, 'Voice Helper: A Mobile Assistive System for Visually Impaired Persons,' in *Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, IEEE, Oct. 2015, pp. 1400–1405. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.209 (cit. on pp. 37, 41).

[87] J. M. Loomis, J. R. Marston, R. G. Golledge and R. L. Klatzky, 'Personal Guidance System for People with Visual Impairment: A Comparison of Spatial Displays for Route Guidance.,' *Journal of Visual Impairment & Blindness*, vol. 99, no. 4, pp. 219–232, 2005 (cit. on pp. 38, 41).

[88] D. G. Lowe, 'Distinctive Image Features from Scale-Invariant Keypoints,' *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. DOI: 10.1023/B:VISI.0000029664.99615.94 (cit. on pp. 48, 103, 116).

[89] D. G. Lowe, 'Object Recognition from Local Scale-Invariant Features,' in *International Conference on Computer Vision*, vol. 2, Kerkyra, Greece: IEEE, 1999, pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410 (cit. on p. 132).

[90] Q.-T. Luong and O. D. Faugeras, ‘The fundamental matrix: Theory, algorithms, and stability analysis,’ *International Journal of Computer Vision*, vol. 17, no. 1, pp. 43–75, Jan. 1996. DOI: `10.1007/BF00127818` (cit. on p. 130).

[91] J. Ma, X. Jiang, A. Fan, J. Jiang and J. Yan, ‘Image Matching from Handcrafted to Deep Features: A Survey,’ *International Journal of Computer Vision*, vol. 129, no. 1, pp. 23–79, Jan. 2021. DOI: `10.1007/s11263-020-01359-2` (cit. on p. 48).

[92] N. Mahmud, R. K. Saha, R. B. Zafar, M. B. H. Bhuian and S. S. Sarwar, ‘Vibration and voice operated navigation system for visually impaired person,’ in *International Conference on Informatics, Electronics & Vision*, IEEE, May 2014. DOI: `10.1109/ICIEV.2014.6850740` (cit. on p. 36).

[93] B. Mandal, ‘Innovative Minds – Goggle for Blind (G4B),’ *Science Reporter*, pp. 58–59, March 2018 (cit. on p. 36).

[94] J. Matas, O. Chum, M. Urban and T. Pajdla, ‘Robust wide-baseline stereo from maximally stable extremal regions,’ *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, Sep. 2004. DOI: `10.1016/j.imavis.2004.02.006` (cit. on p. 132).

[95] I. Matthews, ‘Features for audio-visual speech recognition,’ Ph.D. dissertation, University of East Anglia, Sep. 1998, 185 pp. (cit. on pp. 127, 167).

[96] B. Mayerhofer, B. Pressl and M. Wieser, ‘ODILIA - A Mobility Concept for the Visually Impaired,’ in *International Conference on Computers for Handicapped Persons*, vol. 5105, Linz, Austria: Springer, 2008, pp. 1109–1116. DOI: `10.1007/978-3-540-70540-6_166` (cit. on pp. 35, 41).

[97] P. Mehta, P. Kant, P. Shah and A. K. Roy, ‘VI-Navi: A Novel Indoor Navigation System for Visually Impaired People,’ in *International*

Conference on Computer Systems & Technologies, ACM, 2011, pp. 365–371. DOI: 10.1145/2023607.2023669 (cit. on p. 37).

[98] K. Mikolajczyk et al., 'A Comparison of Affine Region Detectors,' *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 14th Nov. 2005. DOI: 10.1007/s11263-005-3848-x (cit. on p. 130).

[99] K. Mikolajczyk and C. Schmid, 'An Affine Invariant Interest Point Detector,' in *International Conference on Computer Vision*, Springer, 2002, pp. 128–142. DOI: 10.1007/3-540-47969-4_9 (cit. on pp. 26, 111).

[100] K. Moravec, R. Harvey and J. Bangham, 'Scale trees for stereo vision,' *Vision, Image, and Signal Processing*, vol. 147, no. 4, pp. 363–370, 2000. DOI: 10.1049/ip-vis:20000583 (cit. on pp. 127, 167).

[101] J.-M. Morel and G. Yu, 'ASIFT: A New Framework for Fully Affine Invariant Image Comparison,' *Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009. DOI: 10.1137/080732730 (cit. on p. 132).

[102] M. Muja and D. G. Lowe, 'Fast Approximate Nearest Neighbors With Automatic Algorithm Configuration,' in *Proceedings of the International Conference on Computer Vision Theory and Applications*, Lisboa, Portugal: SciTePress, 2009, pp. 331–340. DOI: 10.5220/0001787803310340 (cit. on pp. 54, 55, 101, 103).

[103] E. Na'aman, A. Shashua and Y. Wexler, 'User Wearable Visual Assistance System,' pat., 2012 (cit. on pp. 36, 42).

[104] N. Nandhini, G. Vinoth Chakkaravarthy and G. D. Priya, 'Talking Assistance about Location Finding both Indoor and Outdoor for Blind People,' *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 2, pp. 9644–9651, 2014 (cit. on pp. 36, 99).

[105] M. Nassih, I. Cherradi, Y. Maghous, B. Ouriaghli and Y. Salih-Alj, ‘Obstacles Recognition System for the Blind People Using RFID,’ in *International Conference on Next Generation Mobile Applications, Services and Technologies*, IEEE, Sep. 2012, pp. 60–63. DOI: 10.1109/NGMAST.2012.28 (cit. on p. 37).

[106] D. Nistér and H. Stewénius, ‘Linear Time Maximally Stable Extremal Regions,’ in *European Conference on Computer Vision*, D. Forsyth, P. Torr and A. Zisserman, Eds., vol. 5303, Marseille, France: Springer, 2008, pp. 183–196. DOI: 10.1007/978-3-540-88688-4_14 (cit. on p. 48).

[107] Novartis Pharmaceuticals. ‘ViaOpta Nav.’ (2014), [Online]. Available: https://viaopta-apps.com/ViaOpta-Nav.html (visited on 13/07/2019) (cit. on pp. 38, 41).

[108] R. Öktem, E. Aydın and N. E. Çağıltay, ‘An indoor navigation aid designed for visually impaired people,’ in *Conference of Industrial Electronics*, IEEE, 2008, pp. 2982–2987. DOI: 10.1109/IECON.2008.4758435 (cit. on p. 37).

[109] C. Olvera, Y. Rubio and O. Montiel, ‘Multi-objective Evaluation of Deep Learning Based Semantic Segmentation for Autonomous Driving Systems,’ in *Studies in Computational Intelligence*, vol. 862, Cham, Switzerland: Springer, 2020, pp. 299–311. DOI: 10.1007/978-3-030-35445-9_23 (cit. on p. 68).

[110] OrCam. ‘OrCam MyEye.’ (2013), [Online]. Available: https://orcam.com/en/myeye2/ (visited on 12/07/2019) (cit. on pp. 36, 42, 68).

[111] OsmAnd. ‘Offline Mobile Maps & Navigation.’ (2010), [Online]. Available: https://osmand.net (visited on 13/07/2019) (cit. on pp. 38, 41).

[112] O. Otaegui et al., ‘ARGUS: Assisting Personal Guidance System for People with Visual Impairment,’ presented at the International Technical Meeting of The Satellite Division of the Institute of Navigation, vol. 26,

Nashville, Tennessee: Institute of Navigation, Sep. 2013, pp. 2276–2283 (cit. on p. 39).

[113] K. M. Othman and A. B. Rad, 'A Doorway Detection and Direction (3Ds) System for Social Robots via a Monocular Camera,' *IEEE Sensors*, vol. 20, no. 9, 27th Apr. 2020. DOI: 10.3390/s20092477 (cit. on p. 68).

[114] V. Parubochyi and R. Shuvar, 'Performance evaluation of self-quotient image methods,' *Ukrainian Journal of Information Technologies*, vol. 2, no. 1, pp. 8–14, 2020. DOI: 10.23939/ujit2020.02.008 (cit. on p. 164).

[115] V. Parubochyi and R. Shuwar, 'Fast self-quotient image method for lighting normalization based on modified Gaussian filter kernel,' *The Imaging Science Journal*, vol. 66, no. 8, pp. 471–478, 17th Nov. 2018. DOI: 10.1080/13682199.2018.1517857 (cit. on p. 163).

[116] S. Pavey, A. Dodgson, G. Douglas and B. Clements, 'Travel, Transport, and Mobility of people who are blind and partially sighted in the UK,' RNIB, 2009 (cit. on pp. 22, 23).

[117] J. F. Peters, *Life Among the Yanomami*. Broadview Press, 1998 (cit. on p. 31).

[118] H. Petrie, V. Johnson, T. Strothotte, A. Raab, S. Fritz and R. Michel, 'MOBIC: Designing a Travel Aid for Blind and Elderly People,' *Journal of Navigation*, vol. 49, no. 1, pp. 45–52, 21st Jan. 1996. DOI: 10.1017/S0373463300013084 (cit. on pp. 38, 41).

[119] N. Piasco, D. Sidibé, C. Demonceaux and V. Gouet-Brunet, 'A survey on Visual-Based Localization: On the benefit of heterogeneous data,' *Pattern Recognition*, vol. 74, pp. 90–109, Feb. 2018. DOI: 10.1016/j.patcog.2017.09.013 (cit. on pp. 53, 72, 73).

[120] M. Pielot, B. Poppinga and S. Boll, 'PocketNavigator: Vibro-Tactile Waypoint Navigation for Everyday Mobile Devices,' *International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 423–426, 2010 (cit. on pp. 38, 41).

[121] Plato, *Theaetetus* (Oxford World's Classics), trans. by J. McDowell. Oxford, England: Oxford University Press, 2014 (cit. on p. 31).

[122] L. Ran, S. Helal and S. Moore, 'Drishti: An integrated indoor/outdoor blind navigation system and service,' in *Conference on Pervasive Computing and Communications*, IEEE, 2004, pp. 23–30. DOI: 10.1109/PERCOM.2004.1276842 (cit. on pp. 35, 41).

[123] S. Real and A. Araujo, 'Navigation Systems for the Blind and Visually Impaired: Past Work, Challenges, and Open Problems,' *IEEE Sensors*, vol. 19, no. 15, p. 3404, 2nd Aug. 2019. DOI: 10.3390/s19153404 (cit. on p. 23).

[124] T. Reinhardt. 'Using Global Localization to Improve Navigation,' Google AI Blog. (2019), [Online]. Available: https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html (visited on 26/06/2020) (cit. on pp. 43, 53).

[125] Right-Hear. 'RightHear.' (2015), [Online]. Available: https://right-hear.com/ (visited on 13/07/2019) (cit. on p. 37).

[126] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, 'MobileNetV2: Inverted Residuals and Linear Bottlenecks,' *Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 21st Mar. 2019. DOI: 10.1109/TPAMI.2017.2711011 (cit. on pp. 50, 51).

[127] P.-E. Sarlin, C. Cadena, R. Siegwart and M. Dymczyk, 'From Coarse to Fine: Robust Hierarchical Localization at Large Scale,' in *Conference on Computer Vision and Pattern Recognition*, IEEE, 2019, pp. 12 708–12 717. DOI: 10.1109/CVPR.2019.01300 (cit. on pp. 49, 73–76, 83).

[128] P.-E. Sarlin, D. DeTone, T. Malisiewicz and A. Rabinovich, 'SuperGlue: Learning Feature Matching With Graph Neural Networks,' in *Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2020, pp. 4937–4946. DOI: 10.1109/CVPR42600.2020.00499 (cit. on pp. 48, 50, 75).

[129] P.-E. Sarlin et al., *Hierarchical Localization (HLoc)*, version 1.4, 20th Jul. 2023 (cit. on p. 51).

[130] D. Sato et al., 'NavCog3 in the Wild: Large-scale Blind Indoor Navigation Assistant with Semantic Features,' *Transactions on Accessible Computing*, vol. 12, no. 3, pp. 1–30, 17th Sep. 2019. DOI: 10.1145/3340319 (cit. on p. 44).

[131] T. Sattler et al., 'Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions,' presented at the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE, 2018. DOI: 10.1109/CVPR.2018.00897 (cit. on pp. 51, 75).

[132] J. L. Schönberger and J.-M. Frahm, 'Structure-from-Motion Revisited,' in *Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 4104–4113. DOI: 10.1109/CVPR.2016.445 (cit. on pp. 72, 74, 76, 84).

[133] J. L. Schönberger, E. Zheng, J.-M. Frahm and M. Pollefeys, 'Pixelwise View Selection for Unstructured Multi-View Stereo,' in *European Conference on Computer Vision*, B. Leibe, J. Matas, N. Sebe and M. Welling, Eds., Amsterdam, Netherlands: Springer, 2016, pp. 501–518. DOI: 10.1007/978-3-319-46487-9_31 (cit. on pp. 76, 84).

[134] Sendero Group. 'BrailleNote GPS version 22.' (2002), [Online]. Available: http://senderogroup.com/products/shopgps.html (visited on 12/07/2019) (cit. on pp. 38, 41).

[135] Sendero Group. 'Seeing Eye.' (2013), [Online]. Available: http://senderogroup.com/products/shopseeingeyegps.html (visited on 13/07/2019) (cit. on pp. 37, 41).

[136] A. Shashua and T. Riklin-Raviv, 'The quotient image: Class-based re-rendering and recognition with varying illuminations,' *Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 129–139, Feb. 2001. DOI: 10.1109/34.908964 (cit. on p. 161).

[137] J. Slade and R. Edwards, ‘My Voice 2015: The views and experiences of blind and partially sighted people in the UK,’ 1.1, 2015 (cit. on p. 21).

[138] Snigle. ‘Corsair GPS.’ (2016), [Online]. Available: https://snigle.github.io/corsaire/ (visited on 13/07/2019) (cit. on pp. 38, 41).

[139] S. Srisuk and A. Petpon, ‘A Gabor Quotient Image for Face Recognition under Varying Illumination,’ in *International Symposium on Visual Computing*, vol. 5359, Springer, 2008, pp. 511–520. DOI: 10.1007/978-3-540-89646-3_50 (cit. on p. 163).

[140] Swis Federation of the Blind. ‘MyWay Classic.’ (2012), [Online]. Available: https://itunes.apple.com/us/app/uber/id368677368?mt=8 (visited on 13/07/2019) (cit. on pp. 38, 41).

[141] D. Tang. ‘Disability in Mythic and Ancient Greece.’ (2018), [Online]. Available: https://athenianinspector.wordpress.com/2018/10/21/disability-in-ancient-greece/ (visited on 04/07/2019) (cit. on pp. 30, 31).

[142] R. Tapu, B. Mocanu and T. Zaharia, ‘Real time static/dynamic obstacle detection for visually impaired persons,’ in *International Conference on Consumer Electronics*, IEEE, Jan. 2014, pp. 394–395. DOI: 10.1109/ICCE.2014.6776055 (cit. on pp. 36, 42).

[143] I. Tashev, D. Johnston and H. Gamper. ‘Microsoft Soundscape - Microsoft Research.’ (2018), [Online]. Available: https://microsoft.com/en-us/research/product/soundscape/ (visited on 10/07/2019) (cit. on pp. 39, 42).

[144] The MathWorks Inc. ‘Computer Vision Toolbox.’ (1994) (cit. on p. 111).

[145] The Stationery Office. ‘Equality Act.’ (2010), [Online]. Available: https://gov.uk/guidance/equality-act-2010-guidance (cit. on p. 21).

[146] P. Torr and A. Zisserman, 'MLESAC: A New Robust Estimator with Application to Estimating Image Geometry,' *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, Apr. 2000. DOI: 10.1006/cviu.1999.0832 (cit. on p. 104).

[147] Transition Technologies. 'Seeing Assistant Move.' (2013), [Online]. Available: http://seeingassistant.tt.com.pl/move/ (visited on 13/07/2019) (cit. on pp. 38, 41).

[148] U.S. Air Force. 'GPS Accuracy,' GPS.gov. (2014), [Online]. Available: https://gps.gov/systems/gps/performance/accuracy/ (visited on 07/05/2018) (cit. on pp. 24, 41).

[149] U.S. Air Force. 'Other Global Navigation Satellite Systems (GNSS),' GPS.gov. (2017), [Online]. Available: https://gps.gov/systems/gnss/ (visited on 18/06/2020) (cit. on p. 41).

[150] A. Vedaldi and B. Fulkerson, 'VLFeat - An open and portable library of computer vision algorithms,' in *International Conference on Multimedia*, 2010. DOI: 10.1145/1873951.1874249 (cit. on p. 111).

[151] R. Velázquez, 'Wearable Assistive Devices for the Blind,' in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment: Issues and Characterization*, vol. 75, Springer, 2010, pp. 331–349. DOI: 10.1007/978-3-642-15687-8_17 (cit. on p. 45).

[152] M. H. A. Wahab et al., 'Smart Cane: Assistive Cane for Visually-impaired People,' *International Journal of Computer Science Issues*, vol. 8, no. 4, pp. 21–27, 24th Oct. 2011 (cit. on p. 36).

[153] H. Wang, S. Z. Li, Y. Wang and J. Zhang, 'Self Quotient Image for Face Recognition,' in *International Conference on Image Processing*, vol. 2, IEEE, 2004, pp. 1397–1400. DOI: 10.1109/ICIP.2004.1419763 (cit. on pp. 100, 161).

[154] H. Wang, S. Z. Li and Y. Wang, 'Generalized quotient image,' in *Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 498–505. DOI: 10.1109/CVPR.2004.1315205 (cit. on p. 163).

[155] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarre and D. Rus, 'Enabling independent navigation for visually impaired people through a wearable vision-based feedback system,' in *International Conference on Robotics and Automation*, Marina Bay Sands, Singapore: IEEE, May 2017, pp. 6533–6540. DOI: 10.1109/ICRA.2017.7989772 (cit. on pp. 35, 42).

[156] S. Wang, X. Zhan, Y. Fu and Y. Zhai, 'Feature-based visual navigation integrity monitoring for urban autonomous platforms,' *Aerospace Systems*, vol. 3, no. 3, pp. 167–179, Sep. 2020. DOI: 10.1007/s42401-020-00057-8 (cit. on p. 79).

[157] S. Wang, J. Zheng, H.-M. Hu and B. Li, 'Naturalness Preserved Enhancement Algorithm for Non-Uniform Illumination Images,' *Transactions on Image Processing*, vol. 22, no. 9, pp. 3538–3548, Sep. 2013. DOI: 10.1109/TIP.2013.2261309 (cit. on p. 100).

[158] S. Wang, H. Pan, C. Zhang and Y. Tian, 'RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs,' *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 263–272, Feb. 2014. DOI: 10.1016/j.jvcir.2013.11.005 (cit. on pp. 35, 42, 99).

[159] Wayfindr. 'Wayfindr.' (2017), [Online]. Available: https://wayfindr.net (visited on 13/07/2019) (cit. on p. 35).

[160] T. Weyand, I. Kostrikov and J. Philbin, 'PlaNet - Photo Geolocation with Convolutional Neural Networks,' in *European Conference on Computer Vision*, Amsterdam, Netherlands: Springer, 17th Feb. 2016, pp. 37–55. DOI: 10.1007/978-3-319-46484-8_3 (cit. on pp. 51, 52, 99).

[161] H. J. Wiberg. ‘Be My Eyes.’ (2015), [Online]. Available: https://bemyeyes.com/ (visited on 10/07/2019) (cit. on pp. 35, 40, 42).

[162] M. A. Williams, A. Hurst and S. K. Kane, ‘"Pray before you step out": Describing Personal and Situational Blind Navigation Behaviors,’ in *SIGACCESS Conference on Computers and Accessibility*, New York, NY, US: ACM, 21st Oct. 2013, pp. 1–8. DOI: 10.1145/2513383.2513449 (cit. on p. 40).

[163] World Health Organization, ‘World Report on Disability,’ World Health Organization, 2011, p. 24 (cit. on pp. 21, 32).

[164] World Health Organization. ‘Blindness and vision impairment: Refractive errors.’ (2021), [Online]. Available: https://who.int/news-room/q-a-detail/blindness-and-vision-impairment-refractive-errors (visited on 17/10/2021) (cit. on p. 21).

[165] J. Worsfold and E. Chandler, ‘Wayfinding Project,’ RNIB, 2010 (cit. on p. 33).

[166] Xiaoyang Tan and B. Triggs, ‘Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions,’ *Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010. DOI: 10.1109/TIP.2010.2042645 (cit. on p. 161).

[167] Z. Yao et al., ‘3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML,’ *Open Geospatial Data, Software and Standards*, vol. 3, no. 1, p. 5, Dec. 2018. DOI: 10.1186/s40965-018-0046-7 (cit. on p. 93).

[168] S. Young. ‘Disability is no justification for murder.’ (2013), [Online]. Available: https://abc.net.au/news/2013-09-03/young-kyla-puhle-death/4930742 (visited on 06/07/2019) (cit. on p. 31).

[169] A. R. Zamir and M. Shah, 'Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs,' *Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1546–1558, Aug. 2014. DOI: 10.1109/TPAMI.2014.2299799 (cit. on pp. 53, 55, 59, 60, 99, 101, 106, 110, 114, 116, 117, 155).

[170] P. A. Zandbergen, 'Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning,' *Transactions in GIS*, vol. 13, no. s1, pp. 5–25, Jun. 2009. DOI: 10.1111/j.1467-9671.2009.01152.x (cit. on p. 154).

[171] N. Zhu, J. Marais, D. Betaille and M. Berbineau, 'GNSS Position Integrity in Urban Environments: A Review of Literature,' *Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2762–2778, Sep. 2018. DOI: 10.1109/TITS.2017.2766768 (cit. on pp. 94, 98).