Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

Augmenting Internet Maps by Leveraging Community Created and Systematically Collected Imagery

Michael Kröpfl

Thesis supervisor and first reviewer
Prof. Dr. Franz Leberl
Graz University of Technology

Second reviewer
Prof. Dr. David Nistér
Microsoft

Bothell, Washington, July 2013

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Ort

Datum

Unterschrift

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place

Date

Signature

Acknowledgement

This thesis is the result of fruitful collaboration with many people with whom I have had the honor to work, both at Vexcel as well as Microsoft. It would have been impossible without the guidance and supervision of my various mentors throughout this time. I would like to thank Traude and Franz Leberl, Christian Jorde, Michael Gruber, Martin Ponticelli, Stephen Lawler, Gur Kimchi, Eyal Ofek, Blaise Agüera y Arcas, Frederik Schaffalitzky, Georg Klein and David Nistér, for giving me the opportunity to learn and apply a variety of skills required in a professional career, as an engineer and as a researcher. Particularly I feel grateful with Eyal Ofek for motivating me to pursue my doctorate studies and for supervising several of the research projects described herein.

Many of the research contributions were only possible in the context of a team, providing the necessary tools and data upon which ideas and analysis were based. For the collaboration on the UltraCam-M system I would like to thank the development team, Hannes Hegenbarth, Gerhard Neuhold, Zanin Cosic, Stefan Piller, Stefan Bernögger and Christian Neuner, and the PCV team, Konrad Karner, Joachim Pehserl, Joachim Bauer, Lukas Zebedin, Mario Sormann, Andreas Klaus and Barbara Gruber for their guidance on the requirements and solutions for a mobile mapping system for 3D city modeling. I also feel grateful for working with the members of the Virtual Earth Research team, Yonatan Wexler, Ido Omer, Boris Epshtein and Billy Chen, the Bing Mobile AR team, Daniel Buchmueller, Michael Grabner, Vadim Cugunovs, Dan Hou, and other colleagues at Microsoft such as Kartik Muktinutalapati and Sudipta Sinha on streetside privacy protection, business geocoding improvement, location matching, tag ranking and point cloud registration.

Due to my choice of pursuing the sometimes demanding path of applied industry research, I had less interaction with members of the ICG than other students, although I had the opportunity to work with several world-class graduates. I would like to thank Horst Bischof, Dieter Schmalstieg, Gerhard Reitmayr and Tom Pock for being accommodating to my specific circumstances.

To my team-manager at Microsoft and second thesis reviewer, David Nistér I feel indebted for his continuous backing of my thesis project, his guidance and contribution to the research work and his input during the review of the thesis. I also feel humbled by the opportunity to work in a world-class team of researchers on technology that will help define the future.

However, my deepest thanks goes to my supervisor Franz Leberl for a) introducing me to the areas of photogrammetry, mapping and computer vision, b) entrusting me with the responsibility to work on numerous momentous and challenging research problems, c) giving me the opportunity to pursue a doctorate degree, d) providing the often necessary encouragement and push to keep going and e) the detailed and frank feedback during the thesis review.

I would also like to thank my parents, grandparents and my sister Karin, who have always given me the liberty, trust and support to decide and pursue my own path, even if it involved some unexpected turns such as moving abroad.

Last but not least a project like this cannot be finished without a tremendous amount of motivation. For me, this motivation is sourced first and foremost by my dear Andrea and our little Marianne, to whom I send lots of love. During the many days and nights spent on my research work, I always felt supported, inspired and loved by my family. Thank you!

Abstract

The field of internet mapping has seen substantial growth since 2005, with the entry of new global providers such as Google, Microsoft, Apple and Nokia besides numerous regional ones such as Herold and BEV in Austria or Baidu in China. The systematic collection and release of hundreds of Petabytes of geospatial image data at ground sampling distances (GSD) between 0.5 cm and 15 cm preceded the creation of multiple virtual 3D globes including millions of automatically created photorealistic building models. Further, the transition from a world of 1 billion personal computers in people's homes to a world of 1.2 billion internet-enabled smart mobile devices, localized via global positioning technology has enabled a plethora of new geospatial applications.

At the same time, driven by similar growth in a 100 billion USD cloud computing business, ubiquitous consumer electronics and social networks, various online photo collections such as Flickr, Photobucket, Panoramio, Facebook and Instagram have emerged. These have enabled the centralized collection an inventory of hundreds of Petabytes of images via crowdsourcing, by a community extending beyond 1 billion people.

Both trends lead to an interest to investigate the relation between the systematic nature of traditional mapping techniques and the people-centric capture patterns of crowdsourced data. After establishing common ground about the state of internet mapping, we engage in this particular research area by defining the criteria for geospatial imagery and describing various data types and sources, such as 60-500 cm GSD satellite imagery, 0.5-30 cm oblique and vertical aerial imagery from planes and micro aerial vehicles (MAV), as well as 0.5-2 cm streetside and indoor imagery. We aim to obtain an understanding of the fundamental properties of the two data modalities as well as hybrid forms. We then evaluate ways of leveraging the advantages of each modality and of combining them with the aim of addressing some of the key research problems in internet mapping.

First, we attend to the problem of systematically collecting tens of millions of images on public streets as a base layer for "human scale" internet mapping, as this kind of data most closely resembles the appearance and scales of 0.5-2 cm GSD community created imagery. We develop requirements and propose a solution for this problem, namely a mobile mapping system (MMS) we call UltraCam-M. Interesting problems arise in this setting. These include storage and computational loads on the order of Petabytes per month and thousands of CPU cores.

The actual use of the data poses another issue. For example, systematically collected images must anonymize private information such as people faces and car license plates. Another aspect is that all additions must be fully automatic, both during data capture and processing, as the sheer data volumes of potentially millions of images per day impede significant manual intervention. To address these issues, we propose a novel workflow for automatically detecting and blurring more than 95% of all clearly visible faces and license plates while achieving a low per-pixel false positive rate of 2.5%. Both the capture system as well as the privacy protection method matured into practical applications within Bing Maps.

Returning to the goal of connecting systematically collected imagery with crowdsourced data, we investigate state of the art image retrieval and location recognition techniques. We develop a novel geospatial image retrieval workflow for automatically matching roughly geo-positioned photos to a city scale set of millions of streetside panorama images as well as other geocoded image data which can be added dynamically. The workflow includes an optimized set of local image features, including an interest point detector based on the determinant of hessian (DOH) function, as well as a patch based polar feature descriptor. We further use a rotationally scoped "bag-of-features" based image retrieval method allowing dynamic insertion of new documents, in addition to a novel 1-point similarity and homography RANSCAC method for geometric verification. In combination this leads to high recall rates above 70% and low query times below 3 seconds. After confirming that the DOH image features compare favorably with common alternatives for location based image matching, we evaluate the proposed workflow for various applications relevant for internet mapping, and investigate its applicability for real-time use such as for Augmented Reality.

In a first application we attempt matching dissimilar user-photos from community photo collections (CPC) such as from Flickr to Bing Maps streetside imagery, with the goal of obtaining pixel-accurate geo-positioning and to embed them in the actual mapping service. Photos of the same place, taken by millions of people over various years can hence be connected together for easy finding and online exploration. This leads to increased freshness and image variety, and highlights popular regions on the map. We show that the proposed system can achieve 73% recall and 0.5% false positive rate despite differences in illumination, pose, scene contents and occlusions with server side query times typically below 3 seconds. In combination these results compare favorably with existing approaches. A similar methodology is used in a second application to improve the geocoding accuracy of business listings by matching storefront images to streetside panorama images by an average of 43 m, with comparable recall and false positive rates. This improvement to within a determined error tolerance of 10 m is required for the visibility of the businesses from the respective streetside panorama images in online maps.

As mapping applications on mobile devices still adhere to internet bandwidth and compute restrictions, this report further examines whether 10-fold reductions in the transmitted data volumes, required for to sub-second upload times, affect the quality of results, in order to find an optimal quality/performance tradeoff.

Finally, as the above image retrieval approach mostly applies to urban areas with dense streetside coverage, we also want to investigate whether crowdsourced imagery can instead be registered to aerial or satellite maps with scale differences spanning several orders of magnitude. Based on a reference method, we explore the feasibility of automatically registering shape from motion (SFM) point clouds such as obtained from Microsoft's Photosynth. By employing some improvements such as rotational scoping and fast Fourier convolution, we achieve a 20-fold speedup for registering individual point clouds, while we can double the success rate of the reference method.

Kurzfassung

Das Gebiet der Internetkartographie ist durch den Einstieg neuer globaler Anbieter wie Google, Microsoft, Apple und Nokia neben zahlreichen regionalen Anbietern wie Herold und BEV in Österreich und Baidu in China seit 2005 beträchtlich gewachsen. Die systematische Erfassung und Veröffentlichung hunderter Petabyte von Geo-Bilddaten mit Bodenauflösungen zwischen 0.5 cm und 15 cm war nötig zum Entstehen mehrerer virtueller 3D Weltmodelle, inklusive Millionen automatisch generierter photorealistischer Gebäudemodelle. Der Übergang von einer Welt von einer Milliarde PCs zu einer Welt von 1.2 Milliarden Smartphones mit mobilem Internet und globaler Positionierung ermöglichte die Entstehung einer Vielzahl neuer Geoanwendungen.

Gleichzeitig entstanden, getrieben durch Wachstum im Cloud-Computing sowie der Allgegenwart von Unterhaltungselektronik und sozialen Netzwerken, mehrere internetbasierte Bildsammlungen wie z.B. Flickr, Photobucket, Panoramio, Facebook und Instagram. Diese ermöglichen die zentralisierte Erfassung eines Bestands hunderter Petabyte von Bildern via Crowdsourcing durch mittlerweile mehr als eine Milliarde Menschen, bei einem Wachstum von täglich mehr als 300 Millionen neuen Bildern.

Beide Trends führen zu einem Interesse an der Erforschung des Zusammenhangs zwischen der systematischen Natur traditioneller Kartenerfassungsmethoden, und den popularitätsgetriebenen Aufzeichnungsmustern gemeinschaftlicher Datenaufzeichnung. Nach einem Überblick über Geschichte und derzeitigen Stand der Internetkartografie definieren wir Kriterien für Geo-Bilddaten, und beschreiben diverse Datentypen wie z.B. 60-500 cm Satellitenbilder, 0.5-30 cm vertikal und schräg von Flugzeugen oder Mikro-Flug-Vehikel (MAV) aufgenommene Luftbilder, sowie 0.5-2 cm Straßen- und Innenaufnahmen. Weiters analysieren die fundamentalen Eigenschaften der beiden Modalitäten sowie von Mischformen. Danach erkunden wir Wege, um die Vorteile verschiedener Modalitäten durch deren Verknüpfung zur Lösung einer Reihe von Forschungsfragen in der Internetkartografie zu nutzen.

Zuerst beschäftigen wir uns mit dem Problem der systematischen Erfassung von mehr als 10 Millionen Straßenaufnahmen als Grundbaustein der Internetkartographie im „menschlichen Maßstab", da diese Daten am ehesten dem Erscheinungsbild sowie den 0.5-2 cm Auflösungen des Crowdsourcing entsprechen. Wir ergründen Anforderungen und präsentieren einen Lösungsansatz für dieses Problem, in der Form eines mobilen Datenerfassungssystems namens UltraCam-M. Daraus ergeben sich interessante Problemstellungen, wie z.B. Speicher- und Prozessierungsanforderungen von Petabyte pro Monat und tausenden Prozessorkernen.

Aus der eigentlichen Verwendung der Daten ergeben sich weitere Probleme. Zum Beispiel müssen systematisch aufgezeichnete Daten private Details wie menschliche Gesichter und Autokennzeichen automatisch anonymisieren. Automatisierung bei Aufzeichnung und Prozessierung ist nötig, da die schieren Datenmengen von täglich bis zu Millionen von Bildern nennenswerte manuelle Eingriffe unmöglich machen. Um dieses Problem zu behandeln, präsentieren wir einen neuartigen Workflow, zur automatischen Erkennung und Unkenntlichmachung von mehr als 95%

der deutlich sichtbaren Gesichter und Kennzeichentafeln, unter Erreichung einer 2.5 prozentigen Fehlerquote (False Positve Rate) pro Pixel. Sowohl das Aufzeichnungsssystem als auch der Workflow fanden praktische Anwendung in Bing Maps.

Wieder zurück zum Thema der Verknüpfung systematisch und gemeinschaftlich aufgezeichneter Daten erkunden wir den derzeitigen Wissensstand im Bereich der bildbasierten Suche sowie der visuellen Ortserkennung. Wir entwickeln einen neuartigen Workflow zum automatischen Bildmatching grob (~100 m) geopositionierter Fotos mit einem stadtumfassenden Bestand von Millionen von Straßenpanoramas und sonstigen Fotos, sowie der dynamischen Hinzufügung neuer Bilder. Wir kombinieren lokale Bildfeatures, bestehend aus einem Detektor basierend auf der Determinante der hessischen Matrix (DOH) sowie einem regionsbasierten polaren Deskriptor, mit rotationsbegrenztem und dynamisch erweiterbarem Bildranking mittels visueller Wörter und einem neuartigen Ähnlichkeits- und Homographie- RANSAC Algorithmus zur geometrischen Verifizierung. Dies führt zu hohen Matchraten über 70% und geringen Suchzeiten unter 3 Sekunden. Nach einem Vergleich der DOH Features mit üblichen Alternativen, welcher überlegene Resultate ergibt, evaluieren wir den Workflow anhand mehrerer Anwendungen in der Internetkartographie, sowie für Echtzeit Augmented Reality Anwendungen.

In einer ersten Anwendung versuchen wir das Matching unähnlicher Benutzerfotos aus Online-Bildsammlungen wie z.B. Flickr, mit Straßenaufnahmen von Bing Maps, mit dem Ziel pixelgenauer Geolokalisierung und der Einbindung in den Kartendienst selbst. Fotos derselben Orte von Millionen von Leuten über Jahre hinweg können somit miteinander verknüpft und gemeinsam online durchsucht und erforscht werden. Dies führt zu erhöhter Aktualität und Bildvielfalt, und betont populäre Kartenregionen. Das beschriebene System erreicht 73% Trefferquote und 0.5% Fehlerquote, trotz Unterschieden in Beleuchtung, Pose, Szeneninhalt und Verdeckungen, bei einer serverseitigen Suchzeit unter 3 Sekunden. In Kombination übertreffen diese Resultate existierende Methoden. Eine zweite Anwendung, zur Genauigkeitsverbesserung der Geopositionierung von Firmeneinträgen um durchschnittlich 43 m durch das Matching von Fassadenbildern mit Straßenpanoramas, ergibt vergleichbare Treffer- und Fehlerquoten. Diese Verbesserung auf einen Positionsfehler unter 10 m ist für die Sichtbarkeit von Geschäften innerhalb der Straßenansicht von Kartendiensten erforderlich.

Da Kartenapplikationen auf Mobilgeräten weiterhin Internetbandbreiten- und Rechenkapazitätsbeschränkungen unterliegen, suchen wir einen optimalen Kompromiss zwischen den Hochladezeiten und der übertragenen Datenmengen für das obige System, mit dem Ergebnis einer 10-fachen Datenreduktion für Sub-Sekunden Suchzeiten bei nahezu gleichbleibender Qualität.

Abschließend möchten wir erkunden, ob gemeinschaftlich erfasste Bilder alternativ auch zu Luftbild- oder Sattelitenbildkarten registriert werden können, welche flächendeckend zur Verfügung stehen, jedoch Skalierungsunterschiede von mehreren Größenordnungen ergeben. Basierend auf einer Referenzmethode erkunden wir die automatisierte Registrierung von „Shape from Motion" (SFM) Punktwolken, wie z.B. von Microsoft Photosynth. Durch Verbesserungen, wie die Beschränkung auf plausible Rotationen und Filterung mittels Fast Fourier Transformation (FFT), erreichen wir eine 20-fache Beschleunigung bei der Registrierung einzelner Punktwolken bei Verdopplung der Erfolgsquote im Referenzvergleich.

Contents

3 Categories of Geospatial Image Data 37

List of Figures

List of Tables

1 Introduction

Internet mapping and location inspire a vast research effort. Various types of images get converted into a 3D world model at an Exabyte data volume [1]. This report focusses on two different data modalities, systematically collected geospatial imagery, and crowdsourced imagery from community photo collections (CPC) [2].

Systematic collections by professional mapping entities follow clearly defined and evenly spaced capture patterns throughout cities, states or continents [3]. Crowdsourcing via CPC on the other hand, involves communities involving Millions of amateur photographers. Images are often clustered around a sparse set of hotspot locations or landmarks, dictated by popularity amongst the crowd [4]. By combining the imagery from both modalities in a map, we seek to obtain systematic and dense coverage of large areas for completeness, with highlights around popular locations for relevance. To evaluate the potential of this approach, we need to understand the data volumes associated with the different sources. A subset of major sources is listed in Table 1-1.

An example for large scale data collection for internet mapping is the "Global Ortho" project by Microsoft's Bing Maps, with the goal of obtaining continuous aerial orthophotography of 10 million square kilometers in North America and Europe at 30 cm ground sampling distance (GSD) [5]. This 0.3 Petabyte ortho-image has been generated automatically from an estimated 3 Petabyte of aerial photographs (assuming 10 times redundancy to enable automated processing [1]). Besides vertical aerial photography, oblique imagery are used to provide more visually appealing depictions, particularly of urban buildings. Microsoft has captured a total of 0.5 Petabyte of oblique aerial imagery at 20 cm GSD over an area of 1.45 million km^2.

Provider	Product	Modality	GSD	Coverage	Data Volume
DigitalGlobe	Global Satellite Basemap	Systematic	60 cm	500 million km^2	3.7 Petabytes
Microsoft	Global Aerial Orthophoto	Systematic	30 cm	10 million km^2	0.3 Petabyte
Microsoft	Bird's Eye Oblique	Systematic	20 cm	1.45 million km^2	0.5 Petabyte
Google	Street View Panoramas	Systematic	0.5-20 cm	8 million km	20 Petabytes
Flickr	Photo Sharing	Crowdsourced	0.5-20 cm	global, sparse	4 Petabytes
Facebook	Photo Sharing	Crowdsourced	0.5-20 cm	global, sparse	150 Petabytes

Table 1-1 Overview of Sample Major Data Sources and Volumes

Another example is DigitalGlobe's use of *Earth Observation Satellites* such as GeoEye-1 [6] for the acquisition of all 500 million square kilometers of the Earth's surface at 60 cm GSD resulting in a 3.7 Petabyte global basemap [7]. An even larger dataset of 20 Petabyte including approximately

400 million street-level imagery has been collected by Google Maps, covering 8 million street kilometers worldwide at typically 0.5-20 cm GSD, depending on the distance from the camera [8].

Though systematic collection for internet mapping has resulted in unprecedented geospatial data assets, community data collections represent a potentially superior alternative, particularly for indoor locations such as museums or shopping malls [1]. Flickr [9], for example currently holds a collection of 6 billion community photos, comprising 4 Petabytes of data [10]. An estimate we did based on a sampling showed that 23% or 0.9 Petabyte are "geospatially relevant", meaning that they contain image contents representative for the capture location. Though the data volume of Flickr is smaller than that of Google Street View, it still comprises a relevant asset which has been used frequently for research related to community data exploitation [11, 12, 13, 14, 15, 16, 17]. A significantly larger asset of 220 billion photos comprising 150 Petabyte of data is hosted by social networking provider Facebook [18]. Even if only a fraction of this data (e.g. 20%) is geospatially relevant, it exceeds the data volumes occupied by Google street view imagery. From this analysis we conclude that community sources are indeed relevant in the context of internet mapping, and that their common use with systematically captured sources is worth exploring.

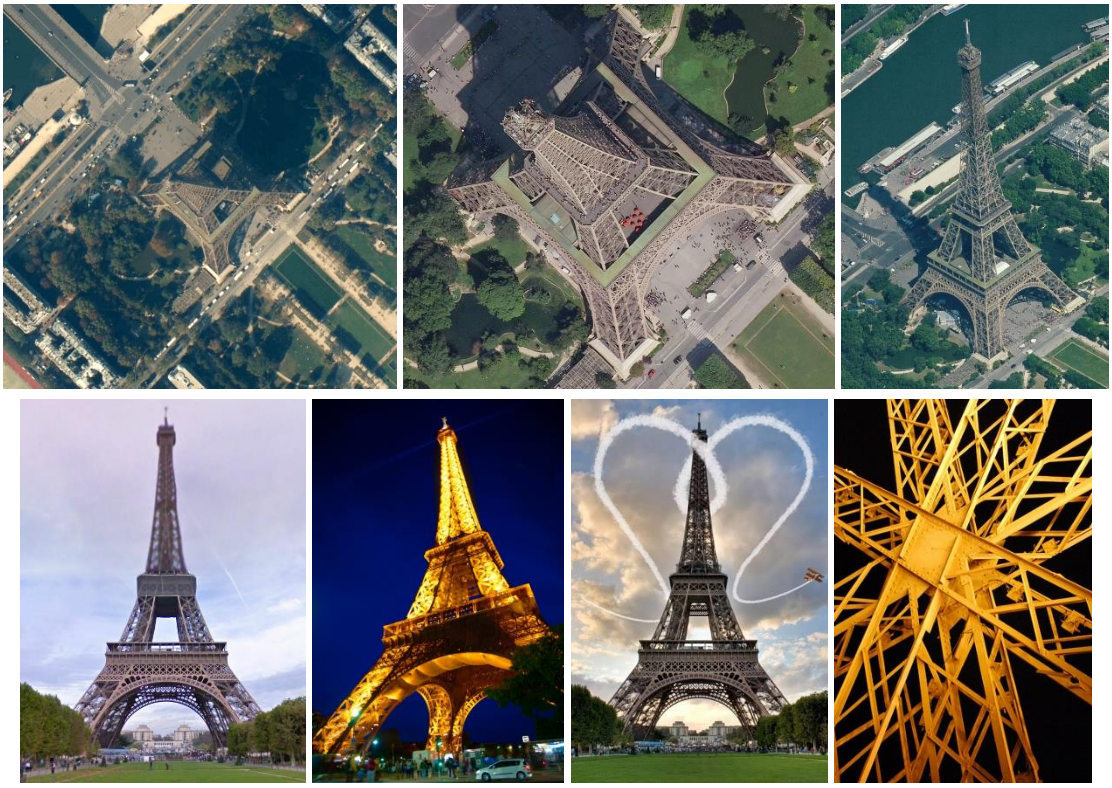


Figure 1.1 Multiple Images of a Famous Landmark (Eiffel Tower, Paris) from Top Left to Bottom Right: DigitalGlobe Global Basemap @60 cm GSD; Microsoft Global Orthophoto @20 cm GSD; Microsoft Bird's Eye Oblique @30 cm GSD; Google Street View @20 cm GSD; Flickr @20 cm GSD [19]; Facebook @5 cm GSD; Flickr @0.5 cm GSD [20];

The different qualities and resolutions of these systematically collected and crowdsourced image types are visualized using the example of a famous landmark in Figure 1.1, ranging from 60 cm GSD satellite imagery (top right) to 0.5 cm user photography from Flickr (bottom left). Note that user photography also contain interesting details such as events or images taken at night.

A particular problem we care about is the augmentation of systematically collected imagery, such as street-level panoramas with imagery from other sources, such as from CPC. This idea is similar to the examples created manually by the Museum of London [21], illustrated in Figure 1.2. Here, a pair of historic photographs from 1962 and 1953 in the foreground has been superimposed onto present-day imagery of the same locations in London, UK, as the background.

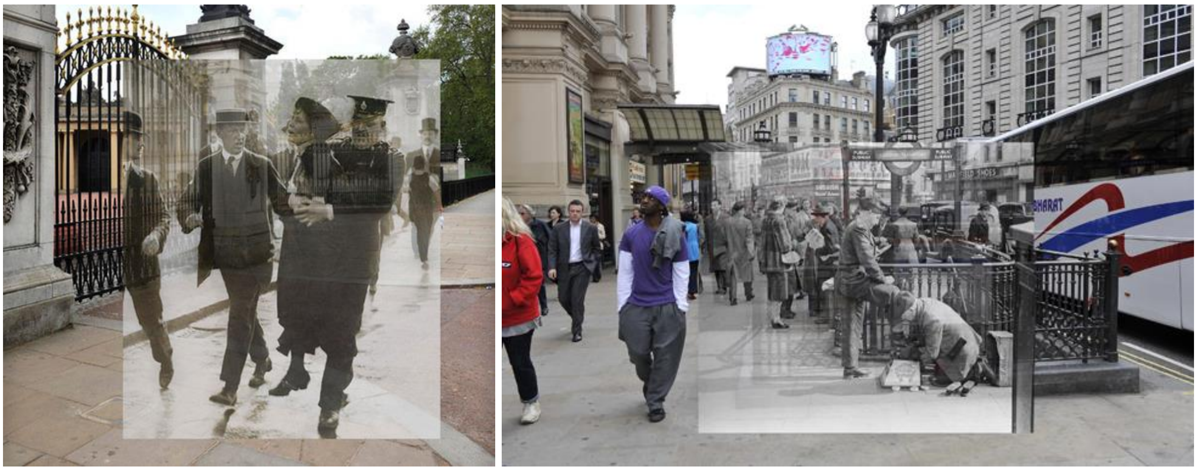


Figure 1.2 Historic Photographs Superimposed onto Present-Day Imagery of the Same Locations

"Emmeline Pankhurst being arrested while trying to present a petition to the King: 1914" © Museum of London (Left) [22];
"A soldier gets a shoe shine outside Piccadilly underground station: 1953" © Henry Grant Collection/Museum of London (Right) [23];

One may envision replacing the historic images by present-day CPC photos in the foreground, superimposed onto street-level imagery in the background. In both cases the background provides context to the narrower field-of-view foreground images, while the foreground highlights some interesting (e.g. historic) details of the scene rather than a systematically obtained street-level view. Generating such image compositions automatically from arbitrary foreground and background images by means of image matching [24], is one of the problems examined in this report. This is a challenging problem due to the large data volumes of up to billions of images illustrated above, as well as the large variety between images caused by differences in lighting, pose, quality, scale and occlusions [25]. Nevertheless, we hope to achieve high success rates above 70% and low false positive rates below 1% to ensure a satisfactory user experience.

Another problem we care about is the automatic detection and obfuscation of private image contents such as people faces and car license plates at close to 100% rates, which is obligatory for systematically captured data in many geographies worldwide [26, 27]. For example the faces visible in the right background image in Figure 1.2 could not be shown in a street-level panorama image on an internet mapping site without prior blurring [28]. This problem is similarly challenging due to the large numbers of hundreds of millions of street-level images, and the variety in the appearance of people and license plates in images.

1.1 Research Questions

The advancements in internet mapping in general and the increasing role of imagery in this domain have resulted in a multitude of new opportunities but also new challenges with respect to the use and management of such data. Hence we look at key challenges in internet mapping and the use of images.

In order to approach the key challenges, we need to have clarity of *"What are maps?"* and *"What is internet mapping?"* This requires one to look at the history of mapping in the context of the World Wide Web, and the basic data types, data sources and applications. Based on existing work on the challenges related to internet mapping [29] as well as our own observations, we quickly can identify multiple unresolved research questions. A side-effect is the *"classification of geospatial image data types"*, based both on the capture mechanisms as well as on their utility for online mapping. However we first need to understand what we consider an "image" and what makes images "geospatial". Is a 3D laser-scan of a statue considered an image, and is it geospatial? How about a photograph of a chair or a painting? Can images of products in a shop be geospatial?

We are also very interested in *"systematically captured human scale imagery"*. This is a source for many related research questions such as how data capture can be done efficiently, how the data volumes can be managed, and what solutions exist for specific problems such as image privacy protection and geolocation. But we are also driven to use *"community created images"*. They differ from systematic data in coverage, accuracy, relevance, metadata and quality. As such it is of interest if and how they can be "*used to supplement systematically collected data"*, and whether interesting geospatial information can be automatically retrieved from such sources.

Different types of geospatial imagery exhibit a variety characteristics and associated metadata. We therefore wish to *"connect these multiple sources at no cost"*, to propagate information across these sources with the aim of improving the mapping experience for the users. In this context it is particularly compelling to use *"image matching techniques"* to connect multiple images despite *"substantial image dissimilarities caused by changes in scale, illumination, scene content, pose, etc."* We looked at existing image matching techniques for reliable matching of dissimilar images, and we find new and improved matching methods. In the process we are able to point to new types of applications for internet maps and LBS.

Based on this general area of research, we aim to answer the following concrete questions:

I. *How did mapping and internet mapping evolve and what is their current state?*
   a. *What are maps in general, and what is internet mapping in particular?*
   b. *What common data types are being used?*
   c. *What are the available data sources?*
   d. *What are the main research areas to advance internet mapping?*

II. *What characterizes systematically collected and community created geospatial images and what is their relevance for internet mapping?*
   a. *What are the criteria for geospatial imagery?*
   b. *How can geospatial images be classified based on the collection methodology?*
   c. *What distinguishes systematically collected from community created images?*
   d. *What different qualities of geospatial imagery exist?*
   e. *What are the data volumes associated with different data types and sources?*
   f. *How do geospatial imagery and their use relate to non-geographic images?*

III. *As human scale imagery are most similar to amateur photos, how have they emerged in internet mapping services and what challenges did providers have to face?*
   a. *What are the major requirements for human scale data capture?*
   b. *How does a particular system design fulfill these requirements?*
   c. *What issues arise in terms of people's privacy with human scale maps?*
   d. *Can these issues be resolved using an automated workflow based on the available data?*

IV. *Assuming that image registration will support information propagation between a variety of sources, how well can automatic registration methods deal with significantly dissimilar geospatial images?*
   a. *What challenges exist with regards to matching dissimilar image from different sources?*
   b. *How can community created images be registered reliably to a trellis of systematically captured images under similar scales?*
   c. *What modifications to existing general image search and registration methods are required for geospatial imagery?*
   d. *How can we measure the performance of such methods in the geospatial realm?*
   e. *Is image registration even feasible under larger scale differences such as between human scale and overhead imagery?*

V. *What practical problems in internet mapping can be addressed by integrating community created and systematically collected imagery and what are the limitations of such attempts?*
   a. *How feasible is it to augment systematic geo-imagery with other sources?*
   b. *How can we increase the frequency of map updates beyond the rate at which systematic data are captured?*
   c. *Can we improve the quality of the map itself by conflating different image sources?*
   d. *Is it feasible to contribute real-time image data to internet maps?*
   e. *What tradeoffs need to be made to achieve real-time responses?*

VI. *What new knowledge about the world can we learn from community created geospatial imagery and their metadata, and then feed into internet maps?*
   a. *Can we visualize information patterns in crowdsourced image data?*
   b. *How can we automatically retrieve such information?*
   c. Is there value in augmenting internet maps with this kind of data?

1.2 Approach to Respond to Research Questions

The history of maps parallels the history of man. Of course we want to be clear about the most recent innovations as reflected in internet mapping (see Section 2.2) but we want to see this as an evolution of several 1000 years of previous achievements in mapping (2.1). Classical geospatial data types are vectors and rasters (images) organized in schemes described in Section 2.4. For internet mapping these sources of mapping data become increasingly varied (2.5). Finally we report on some of the open research problems in this field, as well as the key contributions of the thesis addressing these problems (2.6).

Image data used in geospatial applications have become very diverse. In Chapter 3 we review key characteristics of geospatial image data (3.1) and develop a classification of the data sources based on the acquisition methodology and associated characteristics. We distinguish between systematically collected aerial and space images (3.2 and 3.3), systematically collected human scale images (3.4), crowdsourced images (3.5) and semi-systematically collected images (3.6) and further indicate parallels to non-geographic images (3.7).

A specific type and novel system for collecting terrestrial image data captured at street level is described and evaluated in Chapter 4. This chapter presents the background for the development of a mobile mapping system for streetside data capture (4.1, 4.2) as well as an automatic workflow for protecting private information in the recorded data. The key requirements for streetside image capture at a large scale (4.3), result in a proposed system design (4.4) we call UltraCam-M. Privacy protection is an important requirement for such data. We propose an innovative algorithm, which gets evaluated on a set of approximately 2000 manually labeled images (4.5).

The diversity of image sources for internet mapping led us to research the problem of automatically matching dissimilar geospatial images in Chapter 5. We first introduce the area of location search (5.1), followed by an overview of related work (5.2). We then describe the location search problem we aim to solve (5.3), which involves several improvements (5.3.2) compared to our prior work (5.3.1) to enable various applications related to internet mapping, real-time mobile search and augmented reality (AR). An overview of a proposed extensible real-time image retrieval workflow for geospatial images is provided (5.4), followed by detained description of the feature extraction method (5.5), the image ranking approach (5.6), and the pairwise post-verification (5.7). The image retrieval method is based on prior work by Nistér and Stevénius [30], extended by dynamic addition/removal of images as well as orientation constraints. Additionally we use a novel orientation-constrained post-verification method in combination with local image features based on the hessian interest point detector and a polar patch descriptor in order to achieve optimal retrieval performance in a tolerable time.

Chapter 6 evaluates the image retrieval system proposed in Chapter 5 by means of experiments. We first describe a method and dataset for evaluating and optimizing local features for image retrieval and post-verification, and present evaluation results of the proposed system (6.1). Hence we describe two different applications of image retrieval in the field of internet mapping. One application uses the proposed system in order to precisely geo-position crowdsourced human scale imagery and to show it in the context of systematically captured data (6.1.6). The second application uses the same system with the aim to considerably improve the point of interest (POI) geocoding of businesses for navigation purposes (6.2.3). In both cases we use large sample datasets of tens of thousands of query images in order to numerically evaluate the system performance for the specific application.

For client applications on mobile devices, using a cloud based image retrieval backend, impoverished mobile internet speeds impose restrictions on the data volumes to be transferred.

Therefore we evaluate the impact of reduced image fidelity on the quality of the image retrieval results in order to find an acceptable tradeoff.

Although the Nistér-Stevénius type system is tolerant to a large degree of image variation, it is still limited to images which are captured at roughly the same (human-) scale. Therefore we further propose a system for automatic registration of a set of human scale images to aerial images with substantially different scales (Chapter 7). This method which is based on prior work By Kaminski et al. [14] first applies structure from motion (SFM) [15] on the set, and aligns the resulting point cloud to the aerial view.

1.3 Contributions and Innovations

The area of internet mapping poses several challenges related to the cost and logistics involved in the systematic acquisition, processing and presentation of imagery. This thesis proposes several ways of addressing these issues, both by supporting systematic image collection, as well as by augmenting it with imagery from other sources such as community photo collections.

Mobile Mapping System for the Internet. Systematic capture of terrestrial imagery throughout large areas, such as on public streets within a city, requires automation in the capture process, the processing and the publishing [31]. We propose a novel mobile mapping system [32] we call UltraCam-M which supports the automatic acquisition of panoramic imagery as well as the scene geometry by integrating a cluster of image sensors, depth sensors and navigational components. These data sources are controlled centrally and automatically during the capture process, providing a continuous stream of time-synchronized and geopositioned data for later processing. The propose system found practical application in Bing Maps for capturing streetside imagery in hundreds of cities worldwide, and led to several patent applications for the overall system design [33] as well as sub-components thereof [34, 35, 36, 37].

Privacy Protection of Streetside Imagery. Data captured in urban areas by a mobile mapping system exposes private contents such as people faces and license plates which need to be anonymized for publication on the web [27]. Therefore we propose a novel workflow for the automatic detection and obfuscation of people and license plates in imagery, using a combination of weak classifiers [38] identifying faces and license plates, skin and vegetation regions and planar surfaces, followed by adaptive image blurring taking into account the scale of the private objects estimated from depth data. This workflow also has been used as part of Bing Maps to privacy protect published streetside imagery, and led to several patent applications [39, 26].

Extensible Geospatial Image Index for Real-Time Location Recognition. There exists a desire to connect different sources of geospatial imagery, such as crowdsourced data from CPC and systematically collected street-level images by means of image retrieval and matching [40]. Real-time applications such as for augmented reality [41] additionally require the dynamic addition of new images to a search index, and a query of the index within seconds. We propose a novel workflow for this purpose, supporting real-time queries of an index containing millions of geospatial image documents. This workflow uses optimized image features using minima and maxima of the determinant of hessian (DOH) function [42] for detection and a 3D polar histogram of gradients (HOG) [43] within patch regions for description. Further we employ a dynamic version of a bag of features (BOF) based image retrieval method based on [30], which optionally supports rotational scoping based on orientation priors. Finally a novel post-verification method using the local image features to solve the correspondence problem by means of a rotationally scoped KD-Tree [44], in combination with a 1-point similarity and homography RANSAC [45] algorithm with subsequent optimization and thresholding is used to achieve acceptable recall/precision performance. An early variant of this workflow [25] has been used for matching CPC images from Flickr to Bing Maps streetside imagery in an application called Bing Maps Streetside Photos community tech preview (CTP) [46]. Various patent applications have been filed for the work on matching and embedding imagery with streetside panoramas [47, 48, 49].

Performance Evaluation of Local Image Features. Local feature detectors and descriptors are a key component of many image matching approaches, required to solve the correspondence problem between points in multiple images. A variety of such algorithms exist, and for a particular application an optimal choice is desired. The comparison by Mikolajczyk et al. [50] evaluates the relative performance of multiple interest point detectors for wide baseline stereo matching. However, it does not evaluate the different algorithms on scenes relevant for the problem of location recognition in complex 3D scenes at city-scale with changes in illumination, scale, pose, camera type and occlusions. Thus we propose a more relevant evaluation dataset and framework, to evaluate different local image features, or parameterizations thereof, for both the image retrieval and pairwise matching problems in location recognition. Similar to the recognition benchmark dataset by Stevénius and Nistér [51], our data is organized in quadruples of images of the same scenes, which have been specifically captured for location recognition. As quality metrics, we measure the average number of correct matches in the top 4 ranking results for image ranking, and the area under the ROC curve [52] (ROC integral) for pairwise matching. However, as we care about the applicability of a method on computationally impoverished platforms, we always relate the quality metrics with the corresponding feature extraction times.

Rotation Constrained Registration of Point Clouds to Overhead Maps. The above image matching method can deal with some amount of dissimilarity in lighting, scale, pose and occlusions. However, to deal with even larger scale differences spanning multiple orders of magnitude, we propose an improved workflow for registering point clouds obtained via structure from motion (SFM) [15] to overhead images based on prior algorithm by Kaminski et al. [14]. The proposed approach uses a modified edge cost taking into account the directions of edges in overhead images in addition to their locations, in combination with Fast Fourier convolution to achieve 15-fold speed up of the alignment with more than twice the success rate of the reference method.

1.4 Publications and Patents

Aerial Cameras

I. M. Kroepfl, “*Pulse Pattern Generator for Digital CCD Camera - Implementation by Using a CPLD*”, Diploma Thesis, Graz: Campus 02, 2003.

II. M. Kroepfl, M. Gruber and E. Kruck, "*Geometric Calibration of the Digital Large Format Aerial Camera UltraCamD*" in Proceedings of the Conference of the International Society for Photogrammetry and Remote Sensing (ISPRS) Commission 1, WG 1/2, Istanbul, 2004.

III. M. Gruber and M. Kroepfl, "*UltraCamX Calibration Report*" 18 1 2007. [Online]. Available: http://www.keystoneaerialsurveys.com/pdf/UCXCalReport_30914061_V10.pdf.

Mobile Mapping

IV. M. Kroepfl, M. Gruber, M. J. Ponticelli, S. L. Lawler, J. Bauer, F. Leberl, K. Karner, Z. Cosic, H. Hegenbarth, G. Kimchi and J. C. Curlander, "*Data Capture System*". US Patent 20100182396, 2010.

V. M. Kroepfl, M. J. Ponticelli, H. Hegenbarth, G. Kimchi and J. C. Curlander, "*Determining Exposure Time in a Digital Camera*". US Patent 20100182444, 2010.

VI. M. Kroepfl, J. Bauer, G. Neuhold, S. Bernoegger, G. Kimchi and J. C. Curlander, "*Determining Trigger Rate for a Digital Camera*". US Patent 8284250, 2012 (Granted).

VII. M. Kroepfl, J. Pehserl, J. Bauer, S. L. Lawler, G. Kimchi and J. Curlander, "*Determining velocity using multiple sensors*". US Patent 8244431, 2012 (Granted).

VIII. M. Kroepfl, G. Neuhold, S. Bernoegger, M. J. Ponticelli, J. Pehserl, G. Kimchi and J. C. Curlander, "*Synchronization of multiple data sources to a common time base*". US Patent 7974314, 2011 (Granted).

Privacy Protection

IX. Omer, M. Kroepfl, E. Ofek, K. Muktinutalapati and M. Tabb, "*Identifying Plane Outliers In Scenes Using Re-Projection For Privacy Protection*" IP.com Prior Art Database Disclosure, 2009.

X. E. Ofek, M. Kroepfl, I. Omer, M. Tabb and K. Muktinutalapati, "*Detection of Objects in Images*". US Patent 20100246890, 2010.

Internet Mapping

XI. M. Kroepfl, E. Ofek, Y. Wexler, D. Wysocki and G. Kimchi, "*Geocoding by Image Matching*". US Patent 8189925, 2012 (Granted).

XII. E. Ofek, D. Hou, M. Kroepfl, B. Aguera y Arcas, S. Fynn, R. Molinari and T. Ernst, "*Spatially registering user photographs*". US Patent 8295589, 2012 (Granted).

XIII. D. Buchmueller, M. Kroepfl and F. Schaffalitzky, "*Spatial Attribute Ranking Value Index*". US Patent 8429156, 2013 (Granted).

XIV. M. Kroepfl, D. Buchmueller and F. Leberl, "*Online Maps and Cloud-Supported Location-Based-Services across a Manifold of Devices*" in Proceedings of the Conference of the

International Society of Photogrammetry and Remote Sensing (ISPRS) Commission IV, WG IV/5, Melbourne, 2012.

Location Recognition

XV. M. Kroepfl, Y. Wexler and E. Ofek, "*Efficiently locating photographs in many panoramas*" in Proceedings of GIS - Workshop on Advances in Geographic Information Systems, ACM SigSpatial, San Jose, CA, 2010.

XVI. E. Ofek, M. Kroepfl, J. Walker, G. Ramos and B. Aguera y Arcas, "*Viewing Media in the Context of Street-Level Images*". US Patent 20110173565, 2011.

XVII. D. Buchmueller, M. Kroepfl, D. Nistér, V. Cugunovs, R. Sagula, B. Agüera y Arcas, S. Fynn and E. Ofek, "*Spatial Image Index and Associated Updating Functionality*". US Patent 20120155778, 2012.

Mobile Search

XVIII. D. Buchmueller, A. Akbarzadeh, M. Kroepfl, "*Using Photograph to Initiate and Perform Action*". US Patent 20130156274, 2013.

2 Internet Mapping Services

In March 1946, Argentinian writer Jorge Luis Borges described a vision of a map created by *cartographers* with such *exactitude*, that it would occupy the same space as the world it describes at a **1:1** *scale* [53]. Hence 1 km in the real world would correspond to 1 km in the map.

About 6 decades later in March 2005, Bill Gates portrayed his own vision of a *virtual-reality* like map of the world, enabling the exploration of remote locations in 3D. One would be able to walk the streets, see what the traffic is like and enter shops to explore their merchandise. [54] This vision has led to the release of internet-based *2D street maps* in 2005 [55] and a *3D virtual globe* model of the entire planet in 2006 [56] by a newly formed "Virtual Earth" business unit within Microsoft, which later turned into Bing Maps [57].

Inspired by Gates' vision, Leberl [1] estimated that such a digital model of the entire Earth's surface as well as exteriors and interiors of man-made buildings at **15 cm**, **2 cm** and **0.5 cm** ground sampling distance (GSD) respectively, demands the capture and *photogrammetric* exploitation of more than **1 Exabyte** of image data. Currently, this data volume would fill **290,000** hard drives of 4 TB each, occupy a space of **110 m^3** (a 45 m^2 · 2.5 m room) and weigh **200 tons** (the equivalent of a Boeing 787 Dreamliner aircraft). Chances are, that hard drives will continue shrinking, but so may the respective GSD required for mapping.

While a map of the *Borgesian* dimension remains fictive, these figures indicate how progress in cartography during the last decade has come as close to this vision as ever, driven by the advent and evolution of *internet mapping*. Besides Microsoft, other global providers such as MapQuest [58], Google Maps [59], Nokia [60] or Apple, and numerous regional providers like Herold [61] and BEV [62] in Austria or Baidu [63] in China follow similar visions to map the world at unprecedented details.

These advances follow a history of several thousand years of innovations in mapping [64], since the creation of the first hand drawn maps on rock and clay-tablet 4,000 years ago [65]. The availability of printed maps since the 15th century [66] benefitted a broader community than prior manuscript versions, and new surveying techniques conducted by cartographers and explorers in the centuries thereafter led to increasingly accurate depictions of the entire World's geography [67]. The first digital geographic information systems (GIS) invented in the 1960s and 1970s built upon these prior innovations [68], and laid the technological foundations for today's internet mapping services. Mapping techniques such as digital Photogrammetry [69, 1] or Remote Sensing [70] from aerial or terrestrial platforms allow the detailed and efficient mapping at centimeter accuracy of extended areas such as entire cities, provinces or nations.

In this chapter, we provide an overview of internet mapping and its progress. We first want to answer the question “What is a map?” and obtain an understanding of the history of major innovations in mapping. This leads us to explore maps at a variety of scales, projections and types, such as 2D and 3D maps. We then explore the evolution of internet mapping in particular, and describe common data types and sources used in this field. Finally we provide an overview of some of the major research areas in internet mapping and relate them with the contributions of this report.

2.1 Background and History of Mapping

A famous proverb states that “a picture is worth a thousand words”. This suggests that it may reveal many complex details of an idea or a scene that are difficult to verbalize in written or spoken language. Humans are extremely capable of processing visual information, by far surpassing the current state of the art of machine vision in the majority of applications [71]. This fact has been exploited throughout the history of mankind by means of visualizing real-world objects or abstract concepts in man-made depictions.

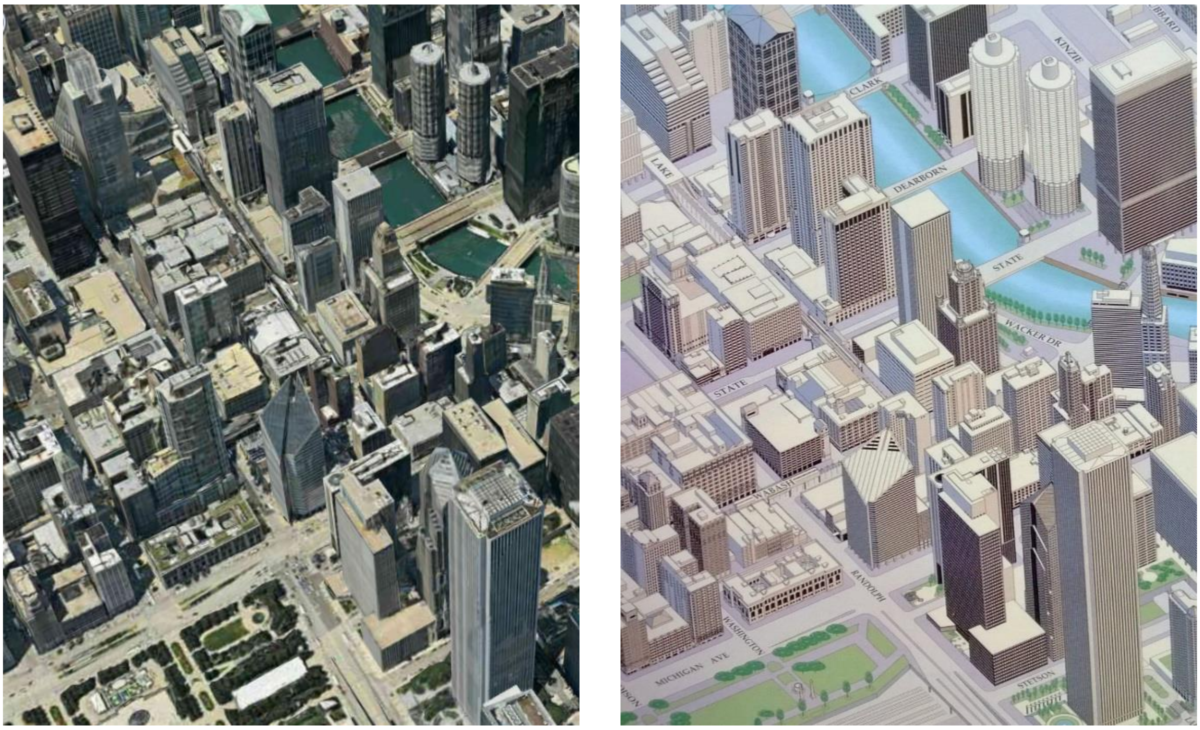


Figure 2.1 3D City Model of Chicago Textured with Aerial Imagery from Nokia Here [60] (Left); Artist’s Drawing of a 3D City Model after Hermann Bollmann [72] Using Abstract Representations of Buildings, Vegetation, Streets and Water Bodies [73] (Right);

In more recent history people have been leveraging the extraordinary human visual processing bandwidth by employing various data visualization techniques, providing easy access to complex information sources. While one may argue that modern *photorealistic* pictures are more detailed and accurate than drawings or paintings, the latter often expose an important property. *Abstraction* is applied as a means of highlighting the essential pieces of information, while deemphasizing or eliminating unnecessary clutter. Maps are a picture-like *representations* of parts of the world, employing varying amounts of abstraction [74]. As a variation of the above proverb, one may suggest that “a map is worth a thousand pictures”, as it employs an additional level of abstraction compared to pictures of the individual elements.

This is demonstrated via the examples of two maps of downtown Chicago in Figure 2.1. A photorealistic *3D city model* from Nokia Here [60] generated computationally by means of *digital photogrammetry* and *computer vision* [3], *textured* using *aerial imagery* and *rendered* using *computer graphics* (CG) [75] is put in contrast with an artist’s abstract drawing of the same scene [73]. In this drawing following principles introduced by Hermann Bollmann [72], complex building geometries and textures have been reduced to simplified structures, streets, water bodies and vegetation are represented by color shades, while text labels for street names have been added.

Both photorealistic and abstract representations are being used in internet mapping for visualizing 2D and 3D maps. While photorealistic maps provide views that are more similar to actual environments, abstract geovisualizations enable the emphasis on essential detail for a given task, say for navigation. Automatically generating abstract versions of 3D city models is therefore an active research area [76].

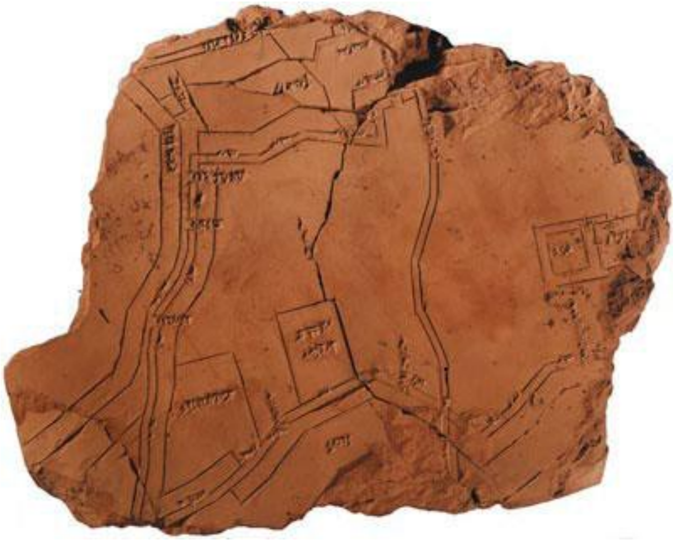


Figure 2.2 Early Map of the Sumerian City of Nippur (dated 1300 BC) – From Hilprecht Collection

2.1.1 What is a Map?

The question “What is a map?” may yield significantly diverse answers depending on who is asked. Most dictionaries or textbooks on cartography define maps rather narrowly as "An accurate depiction of the Earth's surface or part of it at a reduced scale and mathematical projection showing geographical fact." [65] A slightly broader definition is provided in the Oxford English dictionary: "A representation of the Earth's surface or part of it, it’s physical and political features, etc., or of the heavens, delineated on a flat surface of paper or other material, each point in the drawing corresponding to a geographical or celestial position according to a definite scale or projection." [77]

However, both definitions exclude a broad set of depictions of the Earth or its parts. For example historical hand drawn maps and sketches on clay-tablets (see Figure 2.2) are not necessarily metrically correct, nor do they correspond to a specific mathematical projection. Another example related to internet mapping are computer generated 2D "party maps" based on [74], which aim to provide instructions to many people in an area, how to arrive to a particular location.

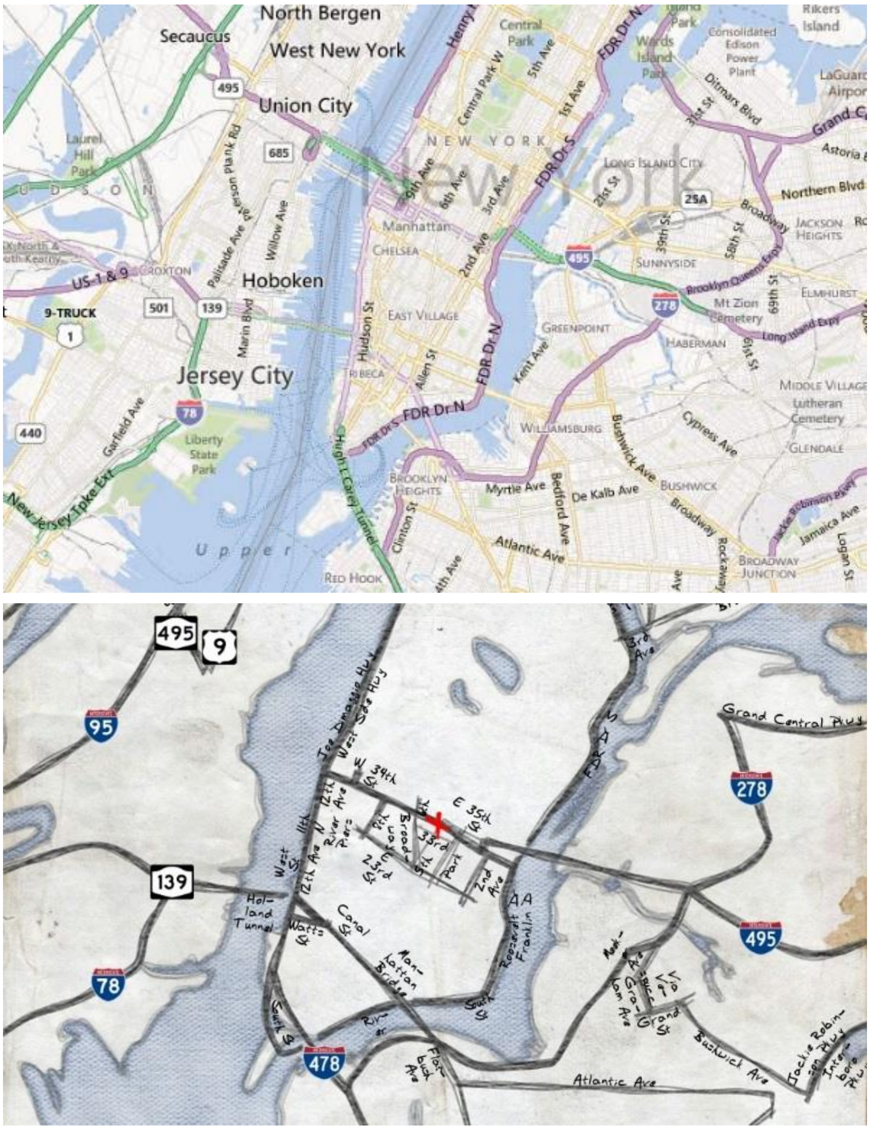


Figure 2.3 2D Vector Map of Manhattan, NY in Bing Maps (Top); "Party Map" to Empire State Building, Manhattan; Generated Using Bing Maps' "Destination Maps" Feature (Bottom);

A respective feature called "Destination Maps" has been integrated in Bing Maps, which uses CG for rendering *conceptualized* maps under intended geometric *distortions*. Same as in case of the 3D model above, this improves the readability for humans by highlighting the essential map elements required for navigation and deemphasizing irrelevant elements. Figure 2.3 compares a common vector map representation of Manhattan, already employing significant abstraction compared to a photorealistic map (top) with a party map pointing out instructions to the Empire

State Building (bottom). Different stylizations of the party map can be used to approximate the appearance of hand-drawn maps.

The narrow "Oxford" definition also excludes maps showing thematic information such as population density, weather statistics, etc. in the geographic canvas of a map of the Earth's surface. As stated by [64], "Maps contain and are more than the simple definition found in a dictionary. To obtain a fuller definition of what a map is we need to better understand for what purpose, why and for whom maps are and were produced."

Wikipedia uses a relatively broad definition for the term "map" in a cartographic sense as "… a visual representation of an area – a symbolic depiction highlighting relationships between elements of that space such as objects, regions, and themes." [78] This definition more closely agrees with the above examples, and with our usage of the term in this report.

2.1.2 Parallels to Hand-Drawn and Printed Maps

Initial terrestrial maps of small areas of land have been dated as early as five-thousand to eight-thousand years ago. Figure 2.2 above shows a clay tabled engraving of the city plan of Nippur, a city in early Mesopotamia which is dated to approximately 1500 BC. It is one of the earliest known maps featuring an *orthogonal* projection roughly to scale with the actual geographic site [79]. Orthogonal projections have been used ever since [5], and are a key element in internet mapping, say for the 2D vector maps shown above in Figure 2.3 or for ortho-imagery further discussed in Section 2.4 and Chapter 3.

Stone and clay engravings were followed by hand-drawn maps on materials such as animal skin, sand, parchment, papyrus or paper. They featured increasingly large areas of the world, defined by the progress made by explorers and cartographers. However, the area of cartography was transformed by the broad availability of printing around the 15th century [66], which allowed for much wider distribution of maps of **1,000** or more prints [80]. This transformation is paralleled, if not exceeded by the advent of internet mapping [81], which raised the frequency of map generation from approximately **800,000** prints every **few years** up until 2008 [82], to more than **1 billion** internet maps every **month** [83].

After cartographers had developed a better understanding of the Earth's spherical nature, more complete and accurate maps such as the "Theatrum Orbis Terrarum" atlas by the Dutch cartographer Abraham Ortelius (1570 A.D.) were created. While still inaccurate compared to a current map from Bing Maps Figure 2.4 (bottom), the example (top) provides a much more comprehensive picture of the world than prior maps, and includes all seven continents [67].

Gerard Mercator, who was a Dutch cartographer like Ortelius invented the "Mercator map projection" in 1569, which preserves equal scales in any direction as well as angles around a specific point on the map. Since the map scale in the Y-direction (Latitude) increases for points farther away from the equator, to keep the same local scale as the X-direction (Longitude), larger objects such as continents become distorted. Therefore maps usually are clipped at a certain longitude angle (e.g. 70-85 degree) to avoid extreme distortions at the poles [84].

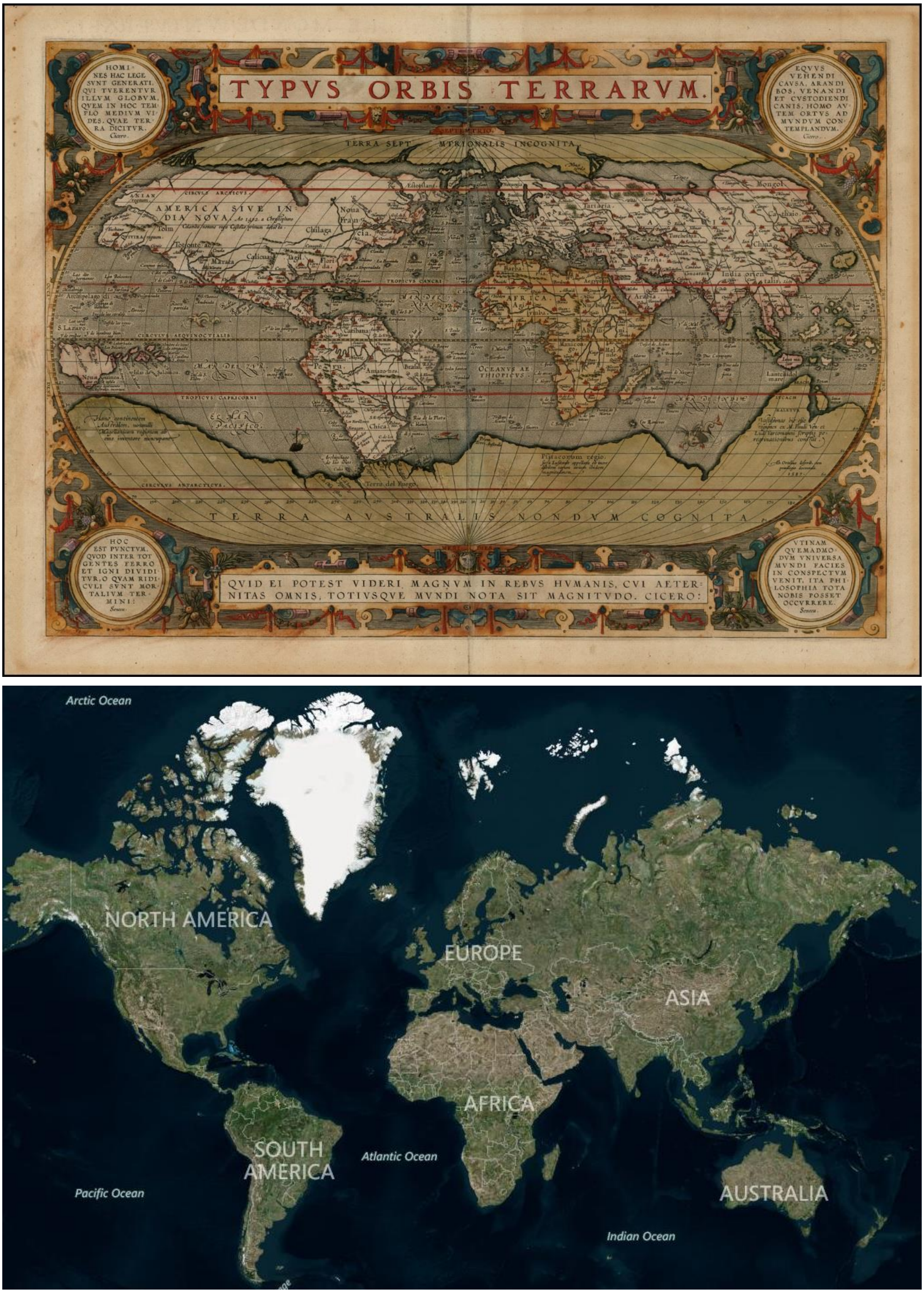

TYPVS ORBIS TERRARVM.
HOMINES HAC LEGE SVNT GENERATI, QVI TVERENTVR ILLVM GLOBVM, QVEM IN HOC TEMPLO MEDIVM VIDES, QVAE TERRA DICITVR. Cicero.
EQVVS VEHENDI CAVSA, ARANDI BOS, VENANDI ET CVSTODIENDI CANIS, HOMO AVTEM ORTVS AD MVNDVM CONTEMPLANDVM. Cicero.
HOC EST PVNCTVM, QVOD INTER TOT GENTES FERRO ET IGNI DIVIDITVR. O QVAM RIDICVLI SVNT MORTALIVM TERMINI! Seneca.
VTINAM QVEMADMODVM VNIVERSA MVNDI FACIES IN CONSPECTVM VENIT, ITA PHILOSOPHIA TOTA NOBIS POSSET OCCVRRERE. Seneca.
QVID EI POTEST VIDERI MAGNVM IN REBVS HVMANIS, CVI AETERNITAS OMNIS, TOTIVSQVE MVNDI NOTA SIT MAGNITVDO. CICERO:
TERRA SEPTENTRIONALIS INCOGNITA
AMERICA SIVE INDIA NOVA
TERRA AVSTRALIS NONDVM COGNITA
Arctic Ocean
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
Atlantic Ocean
Pacific Ocean
AUSTRALIA
Indian Ocean

Figure 2.4 World Map from "Theatrum Orbis Terrarum" (Theater of the World), 1570 A.D. (Top); Satellite World Map in Bing Maps (Bottom);

Most printed and internet maps, for instance the example in Figure 2.4 or the 1965 world map in Figure 2.5, still use the Mercator projection [85, 86]. Elements of the layout used by Ortelius, such as its North-South orientation, or the use of meridians and parallels, served as a reference for atlases printed for several centuries thereafter, and are frequently used in internet maps today.

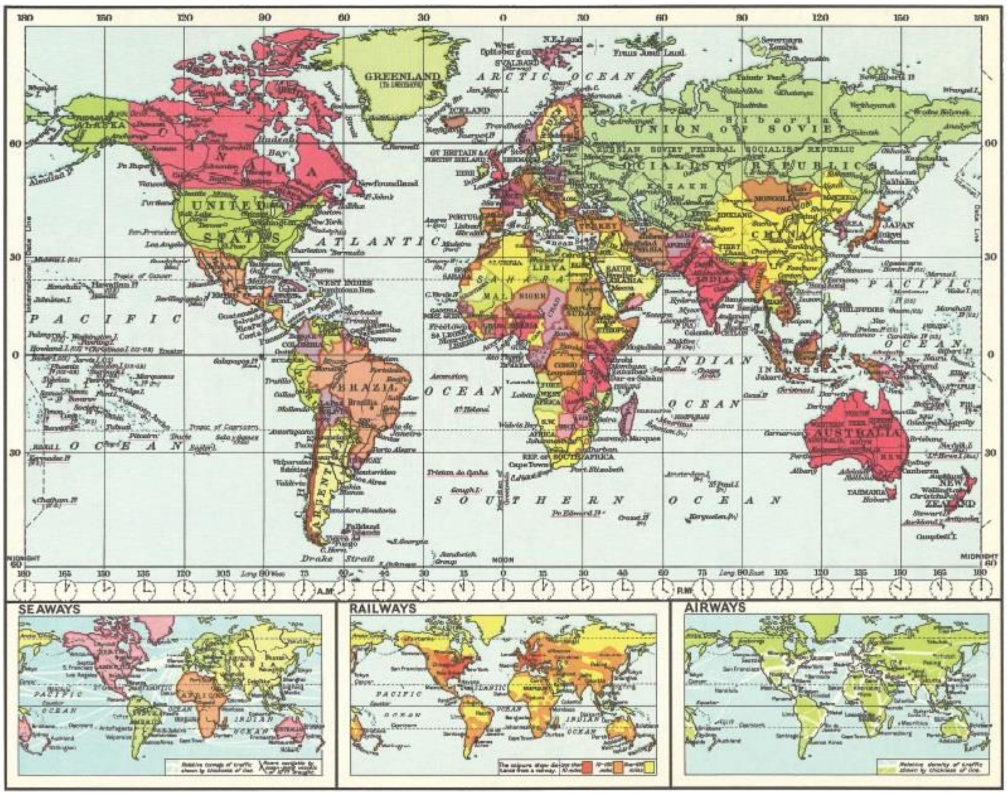


Figure 2.5 Example World Map from 1965 [87]

Along with political boundaries and time zones, the map in Figure 2.5 also contains examples of thematic maps describing themes such as the railway distributions or other statistical data overlaid on the geographic map canvas. In contrast, thematic internet maps visualize live and dynamic information, such as the temperature (see example from weather.com in Figure 2.6).

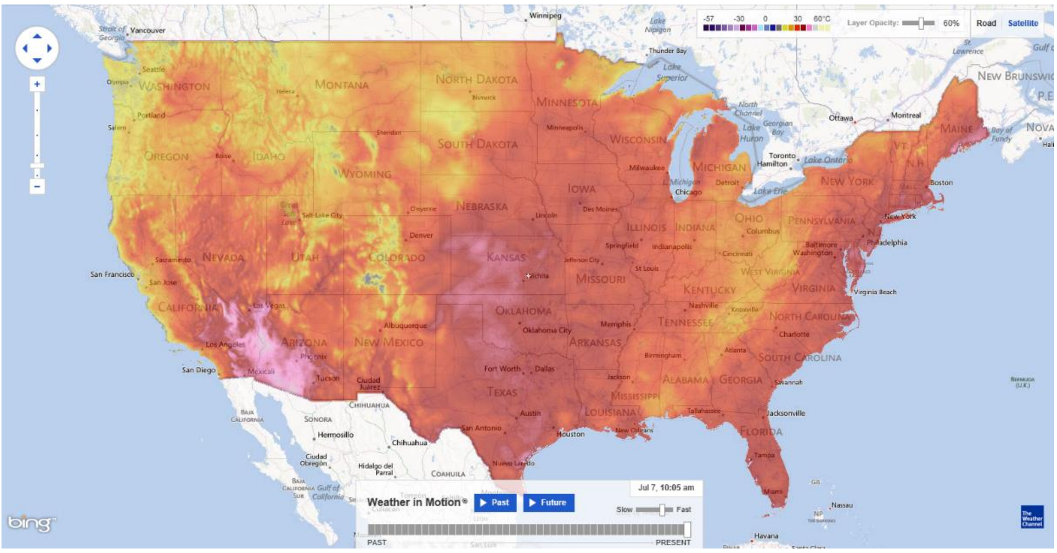


Figure 2.6 Thematic Map of the Live Temperature Distribution in the US on July 7th 2013 [88]

Printing of maps is often based on color separation. A green color shows vegetation objects, blue everything related to water, brown to geomorphology and black to man-made structures such as roads [89]. A similar coloring scheme is still being used for internet maps, as the comparison between a typical printed map and a screenshot from Google Maps in Figure 2.7 illustrates.

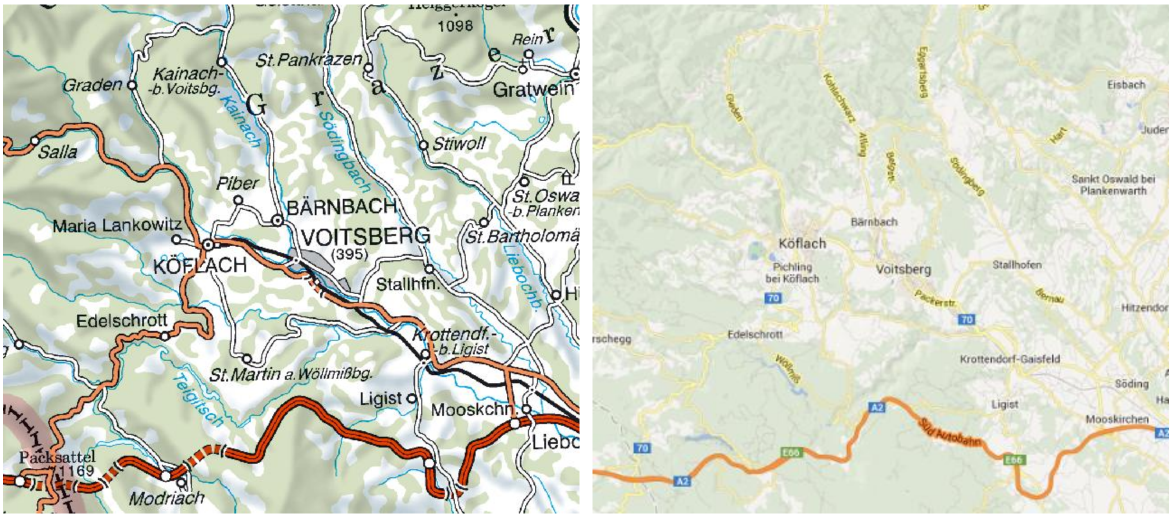


Figure 2.7 „Österreichische Karte“ ÖK 500 Printed at Scale 1:500,000 [62] (Left); Google Map Featuring Similar Coloring Scheme [59] (Right);

Maps get printed at various scales, depending on the application. Large scale maps around **1:100** for instance are used for building floor plans and small scales around **1:100,000,000** for entire world maps [90]. Figure 2.7 (left) shows an example map at scale 1:500,000. In contrast, digital internet maps are not bound to individual scales, as they support dynamic zooming.

Although the above examples only provide a brief insight in the history of mapping and the different types of maps, it is worth noting that a much larger set of maps has been developed over time, focused on specific geographic aspects and tailored towards specific applications. Examples of specialized maps for transportation are nautical and aviation maps, hiking maps, tube maps and road atlases. Other map types include non-metric tourist maps, physical and topographic maps, treasure maps, building floor plans, stellar maps etc. In addition to planar maps, 3D globes, relief maps or miniature models provide a more plastic depiction of real-world landscapes [91].

2.1.3 Digital Maps

Over the course of millennia, maps have been hand drawn or engraved individually or printed in numbers, on a variety of materials such as rock, clay, skin, sand, metal, parchment, papyrus, paper or plastic [67]. This practice has changed starkly with the development of computers with increasing processing capabilities, raster displays and digital representation of maps in various forms [92]. A transition from geodata as a realm of experts, into a commodity for the mass could be noticed since the 1970s [93]. Maps are no longer tied to paper and can be created, distributed, edited and shared much more easily and at unprecedented rates of **tens of thousands of maps per minute** [83]. Digitization in mapping has been an ongoing process since the 1970s, both in the processes involved in producing and distributing maps, as well as in map consumption [93].

Initially digital maps resembled analog maps, as they simply allowed viewing a static segment of the world at a single scale. Printed maps were frequently scanned and converted into a digital PDF format, which allowed digital distribution and viewing, but left out most of the potential advantages of digital maps such as interactivity, multimedia capabilities, and fast updates [92, 94].

"Early examples of more advanced digital maps include the Canada Geographic Information System (CGIS), one of the first operational GIS allowing analysis of geographic data, which was developed in the 1960s and 1970s by the Canadian Department of Forestry and Rural Development". [68] "At about the same time, photogrammetrists started using analytical stereoplotters for measuring elevation profiles in stereo image pairs, gradually replacing analog equipment which had been used for the same purpose." [95] This transition continued by replacing more and more analog steps in the map production workflows with their digital counterparts, such as digitization of analog imagery using specialized scanners [96], and digital processing of scanned imagery in order to obtain photogrammetric measurements. The commercial availability of the first digital aerial metric camera systems in 2003 eventually allowed entirely digital capture and processing workflows [97].

Two major data representations are used for storing and transmitting map data, *raster data* such orthophotos and digital elevation models and *vector data*, including linear features, polygonal features and point features [98]. Both data types can be stored in a myriad of different data structures and formats. More details about the different data types are provided in Section 2.3.

In addition to storing data locally map creators had to find ways of distributing them to a broader set of users. Primarily two forms of distribution for mapping data can be distinguished: Offline and Online distribution. Offline "shrink wrapped" distribution of geographic data to consumers has been available for since the 1980s in the form of digital road atlases and navigation programs stored on digital media such as floppy disks, CDs and DVDs, or in-car navigation systems with pre-installed maps for navigation [93]. Online consumer maps on the internet have been introduced in the early 1990s [94], and dramatically expanded and improved ever since.

In the mid-1990s, despite the growing availability of home computers, network availability was limited. Offline distribution had the advantages that larger data volumes (e.g. 650 Megabyte CDs or 4.7 Gigabyte DVDs) could be transferred more efficiently packaged with interactive user interfaces. Due to the vast improvements and the wide availability of the internet and the World Wide Web, and the emergence of more interactive development tools such as Flash, AJAX and Silverlight this gap has now largely vanished [99].

Examples for offline data distribution are digital trip planning tools such as Microsoft "Streets and Trips", or "ADI WorldMap" by American Digital Cartography which have been available in different versions since 1988 [100]. Microsoft Mappoint is a commercial mapping tool providing geographic data analysis and visualization capabilities to businesses since its release in 2000 [101]. Offline mapping tools are usually commercial and require payment of a license fee, which further differentiates them from most online maps. In more recent versions, many tools use a hybrid approach by combining offline and online data sources and features.

2.2 Evolution of Internet Mapping

The development and advancement of networking technology, specifically of the internet and the World Wide Web, had a similarly transformative effect on digital mapping as the development and growing distribution of computers. While this phenomenon became apparent since about 1995 [102], relatively few publications exist on to the general advancements in internet mapping. The ICA Commission on Maps and the Internet represented by Peterson has provided regular overviews [29, 92, 94, 102, 103] since 1997 of the accelerating transition from a web of documents and static maps, to the current ecosystem of cloud based map services coupled with an armada of GPS enabled mobile devices serving as ubiquitous navigation companions. He points out three primary trends: Internet growth, trends in map types and trends in map use.

2.2.1 Internet growth and internet map growth

The exponential growth of the internet measured in the total number of global web servers (Figure 2.8), during the period observed by Peterson, is a remarkable phenomenon. Similarly, the percentage of the global population with access to internet, has grown significantly during this time. Peterson reported 533 million internet users in 2001, 935 million users in 2004, and 1,300 million users in 2008. According to [104], this number has grown by 70% since then, reaching **2.2 billion** users worldwide in **2013**, which corresponds to **32%** of the world population. As can be seen in Figure 2.9, the per capita internet usage divides the globe, as certain geographic regions lack in internet infrastructure, financial resources or political freedom of information access [29].

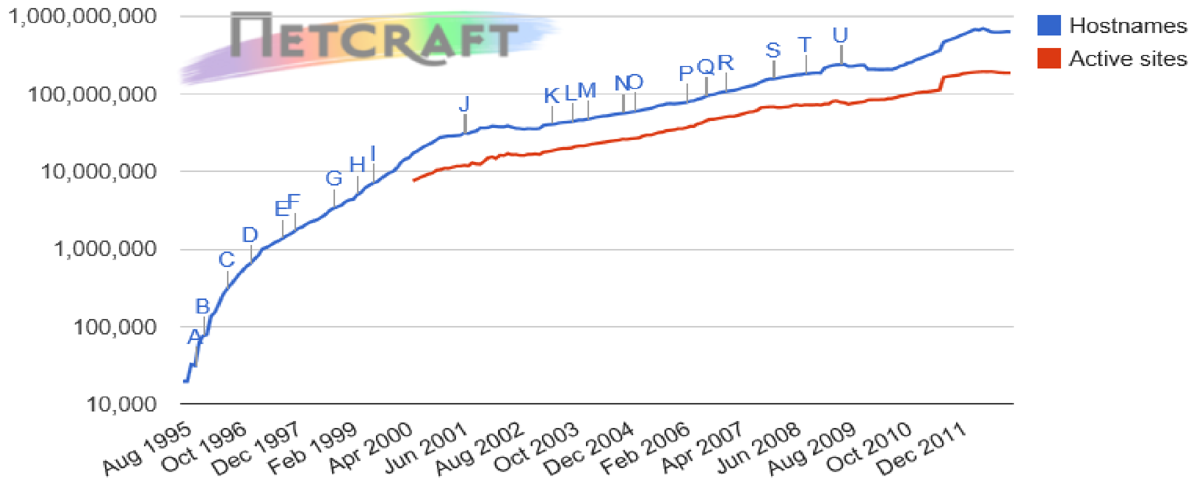


Figure 2.8 History of Total Web Site Count since 1995 from NetCraft [105]

Similar to the growth in internet usage in general, growth in internet map usage has been extraordinary over the last two decades, since the release of the first internet mapping services such as Xerox Parc Map Site (1993), MapQuest (1996), Cern Earth View (1997), Microsoft TerraServer (1997) or Tiger Mapping Service [92]. While little research has been published about the actual growth of internet mapping usage, many factors indicate a growth rate at least as high as for general internet usage.

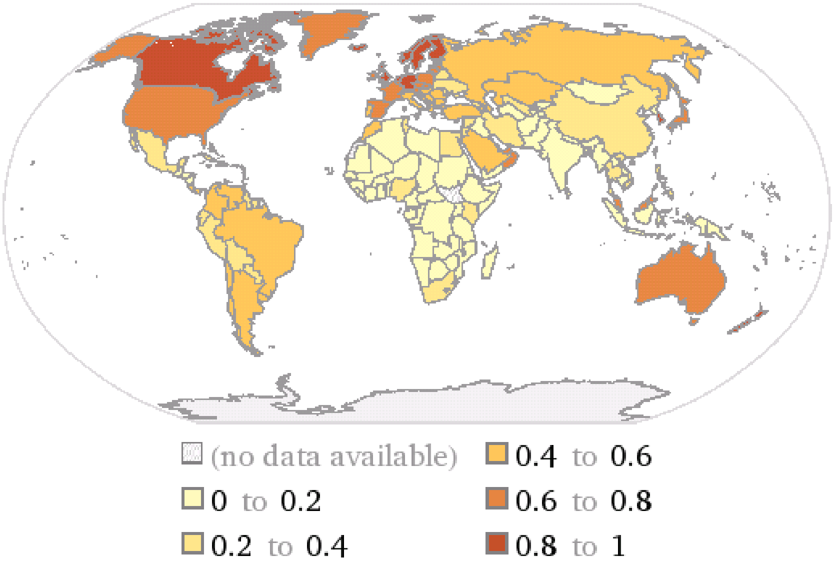


Figure 2.9 Number Internet Users per Capita [104]

A recent shift in internet server technology to "the Cloud" has considerably alleviated the burden for any business or non-profit-organization to operate web services such as for geospatial applications [106]. With the introduction of "cloud computing", servers are no longer operated by individuals, but rather concentrated in server farms and rented out to businesses on a per-use basis. Maintenance is centrally managed by the cloud service providers, such as Amazon EC2, VmWare or Microsoft Windows Azure. This not only helps businesses reduce operating cost, but also creates an enormous flexibility [107]. Virtualization decouples running code from specific physical computers, and scalability allows users to migrate web services onto more or fewer machines almost instantly. The market for cloud computing in 2012 was USD 110 billion, which is predicted to grow at a rate of nearly 20% for the next years [108].

2.2.2 Commercial Internet Mapping on Personal Computers

An indicator for a growing demand is the trend towards commercializing geospatial data and services. While initially map information was often controlled and provided by government institutions or universities, commercial services such as MapQuest [58] took the leadership position in this market [94]. This had a ground-breaking effect on the quality and types of maps. When the first digital maps became available on the internet in the early 1990s, they were still relatively rudimentary, and often represented scanned versions of printed maps served as individual documents via HTML pages or FTP shares. The potential income sources from sponsored business links soon attracted investors such as AOL, which developed the required service infrastructure and more user friendly graphical user interfaces. After its release in 1996, MapQuest soon became the clear market leader in internet mapping accounting for more than 50% market share [92].

This leadership lasted almost 12 years, although other companies such as Yahoo and Microsoft increased the competition by releasing their own commercial internet mapping services in 2002 and 2005 respectively. While Yahoo was able to gain a noticeable market share of around 20%, Microsoft's market share remained relatively constant below 10% during the period reported.

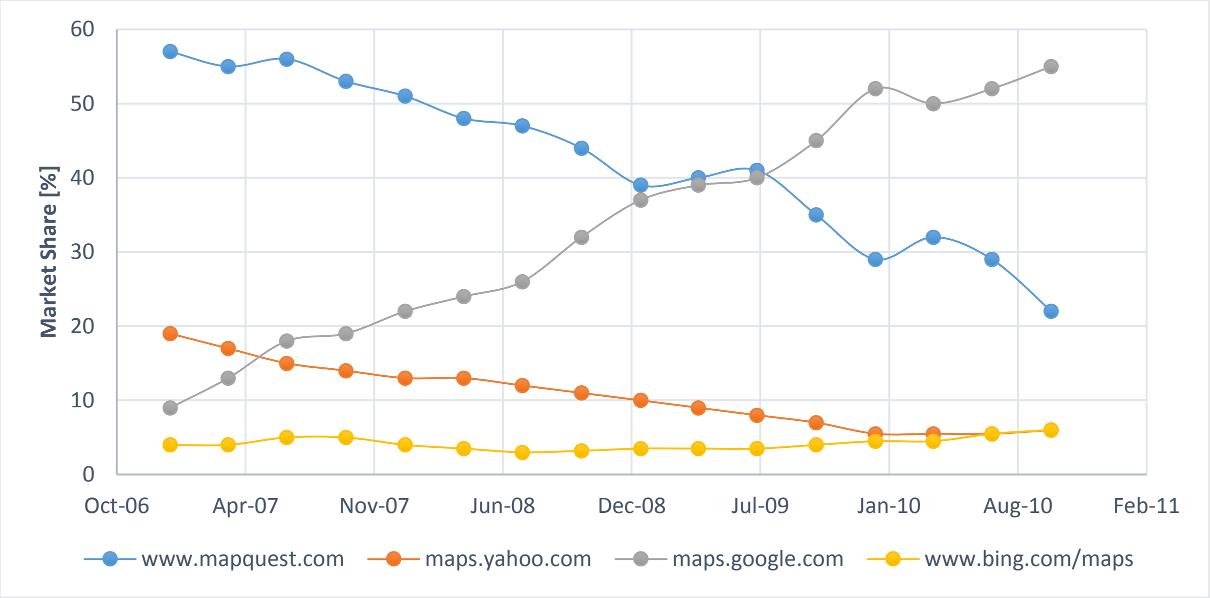


Figure 2.10 Market Share of Major Internet Mapping Services (Source: www.hitwise.com)

A major disruption occurred with the entry of Google Maps in the internet mapping business in 2005, after its acquisition of Where 2 Technologies and Keyhole [109]. Using a significantly modernized and more interactive user interface, it provided pan and zoom functionality with mouse and keyboard as well as street maps and satellite imagery, Google was able to attract the attention of many users almost instantly. Within a 3 year time-period from Mid-2006 to Mid-2009 (see Figure 2.10), the market share of Google Maps grew steadily. It became the market leader by the end of 2009, and has remained in this role since then, while the significance of MapQuest and Yahoo Maps continued to decline. To counteract the leadership of a single company in the mapping area, Microsoft Bing Maps announced a cooperation with Nokia in 2009 in their efforts around the collection and processing of 3D map data and in preparation for a broader cooperation in the mobile devices industry by means of Windows Phone [110, 111].

2.2.3 Common Map Features and Modes

The evolution of a variety of internet mapping services increased the number of features. Initially the primary map type used were street maps, enabling turn-by-turn directions from a point A on the map to another point B, including addresses and points of interest (POI). Later on this functionality got extended by public transit directions, pedestrian navigation and even bike routes. Visualizations of current traffic conditions, as shown in Figure 2.11 further benefit from the real-time updating capabilities of online systems.

In addition to vector-based street maps, satellite and aerial imagery at ground sample distances (GSD) between 15 cm and several meters were released on Google Maps, Bing Maps and MapQuest around 2005, enabling virtual explorations of remote locations of the world from home desktop PCs in photorealistic detail [55, 109, 112, 93]. Google Earth and Microsoft Virtual Earth were released around 2006, providing photorealistic virtual 3D globes, as well as 3D city models [56], followed by Nokia Here 3D in 2012 [113]. More imagery types from 10 cm GSD oblique aerial images [114] down to 2 cm GSD "human scale" panoramas captured at street level or in shops [31, 115, 116], were released later to enrich the visual realism of the mapping experience. More details about different image types used in internet mapping are given in Chapter 3.

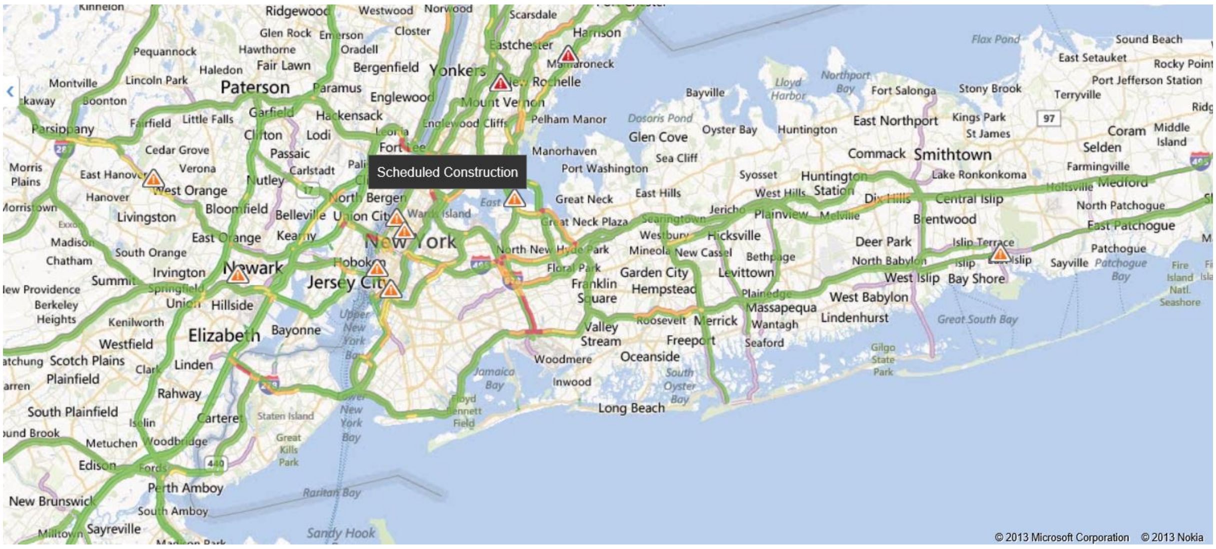


Figure 2.11 Live Traffic Conditions in the New York Area Reported by Bing Maps [57]

Mapping providers also released application programmers interfaces (API) for third party developers, as a basis for developing a range of geospatial applications. Examples of third party applications are real-estate sites, weather maps showing animated cloud coverage, photo sharing sites, and many more geospatial applications. The cost of the API usage is typically based on the number of monthly site visits, although it is often free below a certain usage [117].

2.3 Mobile Maps

After the time period covered in Figure 2.10 little data have been released about the market distribution for internet mapping services. While this can partly be attributed to the leadership position of Google in online-mapping and search in general, it may also be related with another major trend in mapping over the last years. As pointed out in [103], growth in internet map usage is now primarily driven by the broad adoption of mapping on mobile devices, also referred to as "Ubiquitous Cartography" or "Cybercartography".

2.3.1 Personal Navigation

The last decade has seen two major trends in people's map consumption for navigation between places: the decline and predicted death of the paper map [82, 118, 119, 120], and the shift from personal computers and dedicated navigation devices to internet-connected applications on smart mobile devices [121, 122].

While in 2003 chances were people would use a printed paper road map, such as from AAA [123], for route guidance (e.g. during a trip), the emergence of GPS enabled *personal navigation devices* (PND) with digital routable maps has substantially altered this behavior. Printed atlas sales by publishers such as Rand McNally [124] have seen a decline of 7% annually [125] since 2007, while global sales of dedicated **PND** grew from hundreds of thousands to about **40 million** units per year from 2004 to 2008 [121]. However, since 2010 a decline of about 15% annually could be noticed fir PND, caused primarily by a shift to **1.2 billion** [126] internet-connected **smartphone** devices with applications offering similar navigation functionality.

Particularly the release of Apple's *iPhone* in 2007 with an integrated mapping application based on Google Maps, led to a momentous shift in the user behavior. In addition to allowing multi-touch panning and zooming of maps, it also provided functionality for searching of nearby points of interests (POI), and turn by turn navigation [103]. Initial flaws such as small screen resolutions or the lack of GPS positioning were fixed in later versions of the device as well as other smartphones based on *Android*, *Windows Phone* or *Blackberry* [127]. In addition to smartphones, devices with larger screens such as tablet computers have emerged, further extending the scope for mobile map consumption. Since tablet computers such as the *iPad*, *Microsoft Surface* or various Android based versions often share their development platforms with cell phones, the effort for creating applications on either kind of device is reduced.

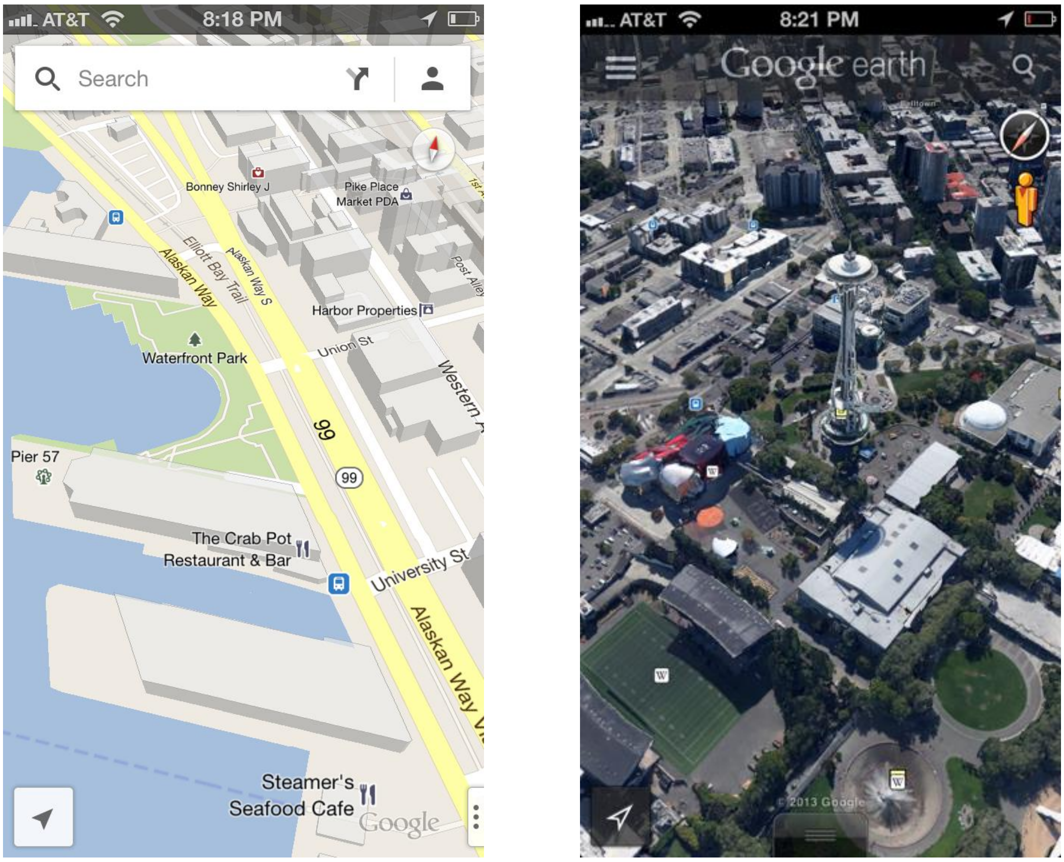


Figure 2.12 Mobile Versions of Google Maps (Left) and Google Earth (Right) showing Downtown Waterfront and Space Needle in Seattle

By 2013 cell phone usage has grown to 0.95 registered devices per capita [104]. An increasing number of mobile phones (22% in 2012) are smartphones with internet accessibility, of which 89% are used throughout the day [126]. New network technologies such as "Long Term Evolution" (LTE) have led to an increase in the available network bandwidth. While 3G networks in 2008 offered bandwidths up to 14.4 Mbit/s for download and 5.8 Mbit/s for upload [103], LTE supports download speeds of up to 300 Mbit/s and upload speeds up to 75 Mbit/s [128].

An indication of the growing importance of mobile mapping is the amount of effort companies put into strengthening their position in this domain. In 2012 Nokia, which had acquired GIS data provider NavTeq in 2009, launched its internet mapping platform "Here" [60] to support mapping applications on new Nokia branded Windows Phone 8 devices. In February 2013 Yahoo retired its own desktop mapping service and replaced it by Nokia's "Here" [60].

To reduce its dependence on Google, Apple decided to replace the mapping functionality on iOS 6 based devices with its own mapping service in 2012, additionally providing turn-by-turn navigation [129]. However, data quality issues caused significant criticism by users, which led Apple to later permit the release of Google Maps and Google Earth for the iOS platform [130], including both 2D and 3D maps as shown in Figure 2.12.

2.3.2 Location Based Services

Besides the built-in mapping functionality of mobile devices, mobile operating systems also provide API's enabling mobile-app developers to create location based services (LBS). LBS are applications using information such as the current location or time in order to achieve a desired result [127]. For instance, GasBuddy searches for the lowest priced gas station in an area [131].

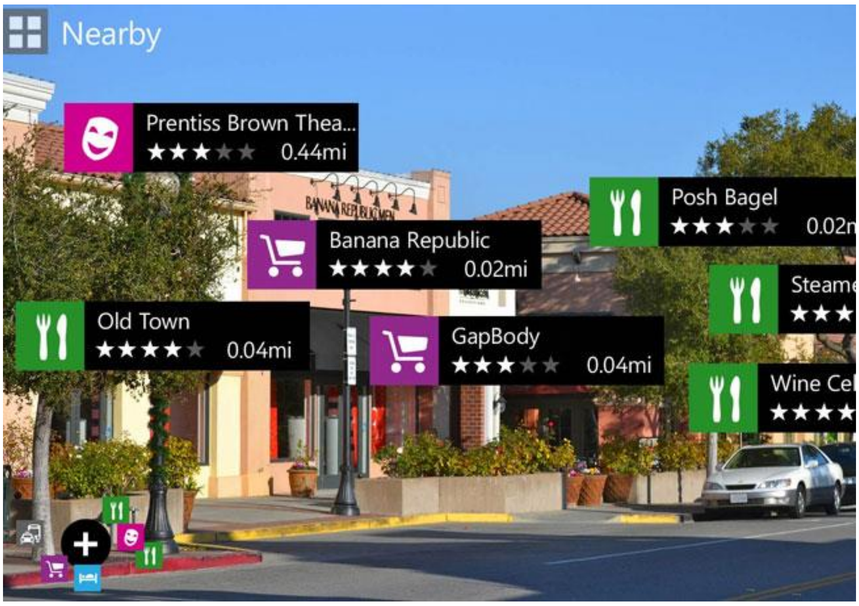


Figure 2.13 Here City Lens Augmented Reality Application from Nokia Shows POI Augmentations Superimposed on Camera Stream; Phone is Oriented via API using Orientation Sensors [132]

Along with providing similar user experiences as on the desktop PC, mobile devices offer new I/O capabilities such as multi-touch, voice commands and voice-guided navigation. Furthermore the availability of inertial sensors and cameras integrated in mobile devices permit new interaction and visualization modes such as Virtual and Augmented Reality (AR). Some AR applications enable the exploration of 360 degree panoramic images, by simply rotating the phone. Such images may have been captured at remote locations and shared via the internet. Other AR applications such as Nokia's Here City Lens (Figure 2.13) superimpose 3D located augmentations related to restaurants or similar points of interest on a live video stream from the phone's video camera, thus enabling new ways of exploring and interacting with geospatial map data [132].

Modern development tools such as the Windows Phone software development kit (SDK) in combination with cloud computing enable multiple new business opportunities by realizing geospatial services with minimal development effort. In [106] we demonstrated that a simple mobile application for geospatial photo capture, involving a cloud based web-service could be developed in only 30 labor hours. We further explained the basic steps necessary to deploy an application, involving a web service hosting geospatial information and a map and camera enabled client software, consuming the web service through an API.

2.4 Common Geospatial Data Types

Internet mapping applications adopted the same data types used in offline digital mapping as well as GIS. Geospatial data are primarily categorized as either *vector data* or *raster data* [98]. A comparison of vector data and raster data is given in Figure 2.14 using the example of the "Maps" and "Satellite" modes of Google Maps.

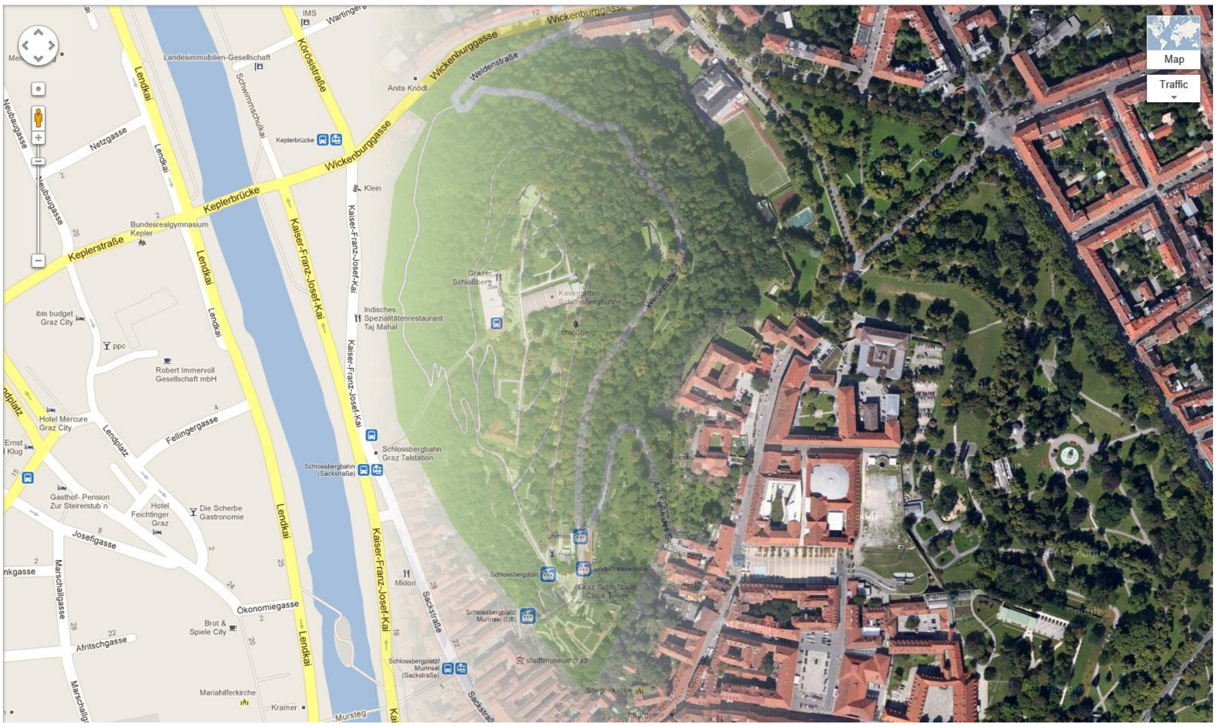


Figure 2.14 Comparison of Vector Data and Raster Data Used in Google Maps (Location: Schlossberg Mountain in Graz, Austria)

2.4.1 Vector data

Vector data comprise the basic data type for geographic information, consisting of mathematical descriptions of different shapes. In a simple case vector data include vertices with geographic coordinates and edges connecting the vertices. Vector data are used for 2D or 3D features. Depending on the shape and size of the feature, three types can be distinguished [98]:

- *Point features* indicate discrete nonadjacent geographic features or abstract points on a map. Examples are points of interest such as restaurants, vista points or bus stops, and abstract points such as city centers or meeting points.
- *Linear features* describe elongated geographic features with a certain linear extent and width. Common examples are rivers, streets or railways. The linear extent of linear features can be measured, such as to determine the distance between two locations on a street. Curvature may be described by polylines or polynomial functions such as splines.
- *Polygons* represent extended areas such as lakes, forests, buildings or city boundaries. As they are two-dimensional both their perimeter and area can be measured.

In addition to geographic coordinates, vector features may contain attributes such as names, addresses, or classifications, which are required for map visualization or analysis. Navigation and route planning require various other metadata such as speed limits, turn restrictions at road intersections or ferry and public transport schedules [133].

To assure that data from different sources can be combined in the same map consistently, a clearly defined coordinate system has to be used. The most commonly used coordinate system is the World Geodetic System 84 (WGS84), a polar coordinate system based on the geoid of the Earth. Coordinates are measured in latitude/longitude/height triplets [134].

Despite the fact that the underlying data may be stored in a vector format, internet mapping often uses pre-rendered "rasterized" versions of vector data. This mainly helps to reduce the rendering effort on the client computer and assures consistent appearance of the map across different devices and browsers. In this case raster images representing the vector data are downloaded to the client [135].

An example for the common use of raster and vector data in a 3D map is the city model shown above in Figure 2.1 (left). In this case, triangles are uses to describe the 3D geometry of buildings, while raster imagery is used to texture the building façades.

2.4.2 Raster Data

While vector data use simple geometric primitives to describe distinct geographic features, raster data provide a continuous grid of values for an area of a map. Each cell (pixel) in a grid of cells is associated with respective values, describing a specific property of that cell [136]. In case of 3D raster data *voxels* represent volumetric units.

Raster data in GIS and internet mapping are used to represent features such as:

- *Images* like for instance orthophotos or oblique photographs discussed in Chapter 3.

- Continuous data such as used for the temperature map in Figure 2.6 above.
- Thematic information such as the colors used to distinguish multiple countries.

An important parameter of a raster map is the spatial resolution defining how many grid cells are covering a specific geographic extent. This is often measured as the distance on the ground spanned by a single cell called ground sample distance (GSD) [136].

While raster data can be stored in flat arrays, hierarchical data structures such as *quadtrees* [137] for 2D and *octrees* for 3D allow more efficient addition and search of individual entries at varying level of detail [138].

2.4.3 Geocodes

A common feature on most internet mapping sites is the search of specific geocode locations on a map based on a POI name, ZIP code or a street address. This process is generally referred to as geocoding. A geocode is a georeferenced latitude/longitude coordinate "… assigned to a specific entity for the purpose of identifying its location on the Earth's surface." [139] The opposite process, which is finding a street address or other semantic location description based on a latitude/longitude tuple is called reverse geocoding.

In order to facilitate geocoding and reverse geocoding functionality, a mapping between addresses and geo-coordinates is required. The accuracy of geocodes depends on the geocoding method and ranges from **several meters** to **several hundreds of meters** [140]. In the US and other countries with grid-like streets, geocodes for individual house numbers may be obtained simply by interpolating across locations of neighboring intersections. For example house number 1450 of an avenue is presumably in the center between 14th street and 15th street. As this assumption is not necessarily true in all areas, the resulting geocoding coordinates may occasionally have large errors up to hundreds of meters.

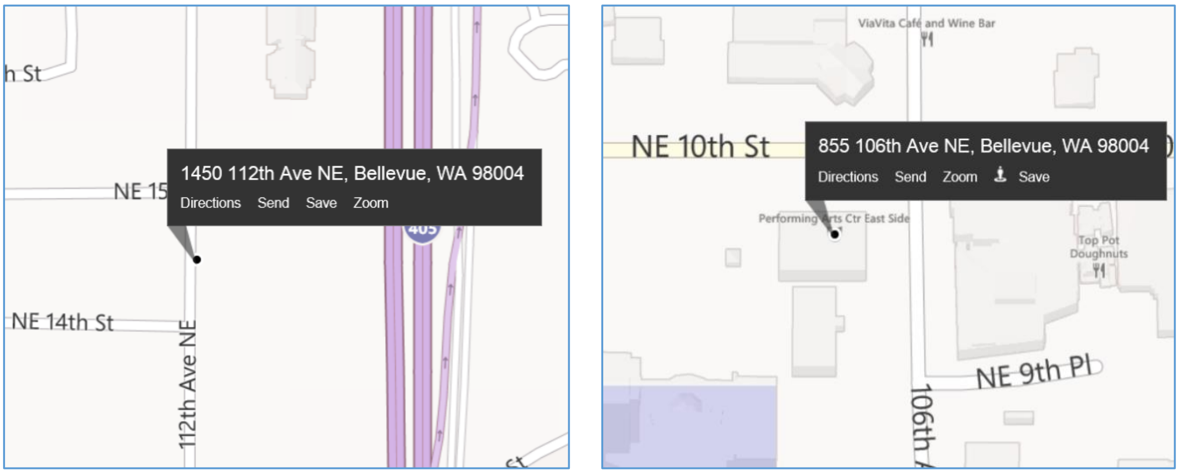

h St
NE 15
1450 112th Ave NE, Bellevue, WA 98004
Directions Send Save Zoom
405
NE 14th St
112th Ave NE
ViaVita Café and Wine Bar
NE 10th St
855 106th Ave NE, Bellevue, WA 98004
Directions Send Zoom Save
Performing Arts Ctr East Side
Top Pot Doughnuts
NE 9th Pl
106th

Figure 2.15 Comparison of Interpolated Geocoding (Left) and Rooftop Geocoding (Right)

This approach is not possible in all locations such as in countries with less regular addressing schemes, and the obtained accuracy level may be insufficient for navigation purposes. Hence alternative methods such as rooftop-geocoding or geocoding using volunteered data need to be used [141]. Rooftop-geocodes are supplied by specific data provides such as MelissaData [142], in which case the geo-location for an address is provided as the latitude/longitude tuple of a building's rooftop. Even more precise geocoding may be required for applications involving human scale imagery [47], in order to guide a user to the correct location of a parking lot or a business front door, which is illustrated in more detail in Section 6.3.1 below.

2.4.4 Data Organization – Quadtree

In order to simplify the generation and consumption of geographic data at various levels of detail, hierarchical data organization may be required.

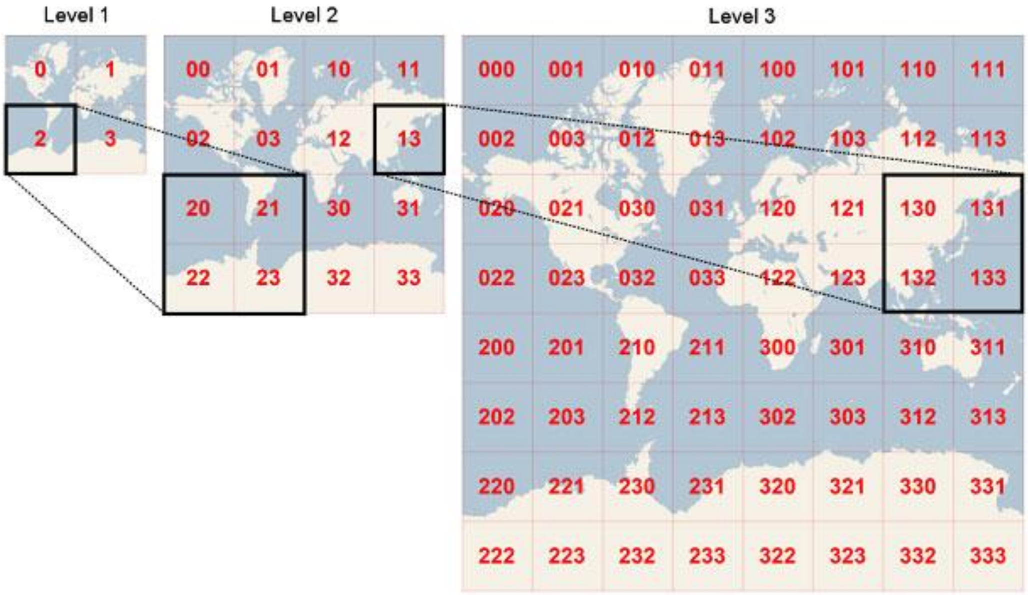


Figure 2.16 Hierarchical Tile System used in Bing Maps; [86]

Multiple providers such as Bing Maps, Google Maps or OpenStreetMaps use an addressing scheme based on the quadtree data structure [138] which has four children nodes for each node in the tree [143]. At the top level of the tree, 4 square tiles with indices {0,1,2,3} cover the whole extent of a map generated using the Mercator projection [84] with a clipping latitude of **85.05 degrees** (Figure 2.16). Each tile image has a dimension of **256*256 pixel**. At the next tree level each of the four nodes branches again into 4 nodes with indices {0,1,2,3}, which is repeated for each consecutive level up to a certain tree depth (19 in case of Google Maps). At a specific level n of the tree the whole map is hence represented by 4^n map tiles. A particular tile of the map can be addressed via a string of length n consisting of the branching indices for each level (such as "120" in case of Central Europe). Depending on the viewport of the map on the client computer, only tiles at the respective level and within the viewport have to be downloaded. Details of the exact addressing schemes for Bing Maps and Google Maps are provided in [86] and [85].

Figure 2.17 shows the relationship between the quad level and the number of tiles and pixels required to cover the whole world, as well as the length of a tile at the specific quad level measured at the equator. In order to cover the surface of the Earth at a tile size of roughly **100 m**, the chosen quad **level 18** results in **275 billion tiles** and **18 Petapixel**.

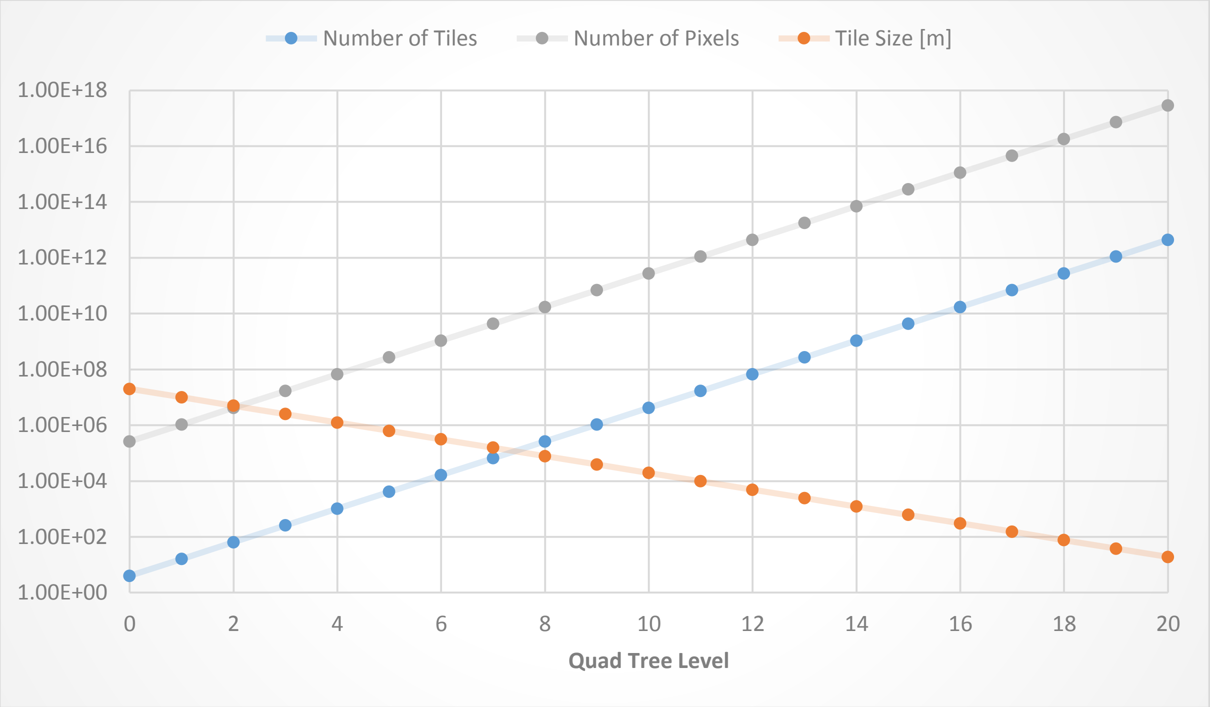


Figure 2.17 Relationship between Quad Level and Number of Tiles/Pixels as well as Tile Size [m]

2.5 Data Sources

Internet mapping services usually integrate a combination of data sources including government agencies, commercial data providers and community sources. The data obtained from these sources include both vector and raster data as described in Section 2.4. The following section focusses primarily on vector data sources, while the sources for different types of image raster data are covered in Section 3.

2.5.1 Government Mapping Agencies

Traditionally mapping data were primarily acquired, managed and distributed by *national mapping agencies* such as for example the “United States Geological Survey” (USGS), “Institut Géographique National” in France, “Ordnance Survey” in Great Britain or the “Bundesamt für Eich- und Vermessungswesen (BEV) in Austria [62].

While these government agencies still exist and play a central role in individual countries, the establishment of navigation systems and internet mapping created requirements for new kinds of data such as streetside imagery, turn restrictions or speed limits for navigation. Additionally a demand for a globally consistent maps with comparable quality and completeness cannot easily be fulfilled by a plethora of state agencies.

2.5.2 Commercial Data Providers

As a consequence of the additional demands for global internet mapping and in-car navigation systems, various commercial companies entered this market. Two of the largest providers of vector data for mapping applications are TeleAtlas and NavTeq which were founded in 1984 and 1985 respectively [127]. In order to acquire navigation data at a large scale, they operate fleets of vehicles equipped with different sensors such as GPS receivers, cameras and laser scanners as part of *mobile mapping systems* (MMS) [32] in order to collect data for offline analysis and data extraction. Images are hence analyzed either manually or automatically in order to detect road lanes and intersections, street signs or business locations.

While both providers are still actively selling navigation data to different companies such as Garmin, Microsoft or MapQuest, the market situation shifted due to the acquisitions of TeleAtlas by TomTom in 2007 for **USD 2.9 billion** [144] and NavTeq by Nokia in 2008 for **USD 8.1 billion** [145]. The prices paid are indicative of the significance of geodata to companies invested in mobile device and internet technology. In 2011, Google largely abandoned the use of third party sources for Google Maps and replaced it with its own data, presumably to reduce cost for licensing and to benefit from its existing assets such as satellite and streetside images [146].

In addition to license fees, restrictive license terms further increase the benefit for companies to be independent from third party data providers. As a sample NavTeq contract [147] indicates, use of the data is often limited to very specific applications not permitting any modifications, adaptations, additions or alterations.

2.5.3 Community Mapping

A relatively new way of obtaining and improving geospatial data is by means crowdsourcing the tasks of data entry and verification by an online community. “Crowdsourcing is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, and especially from an online community, rather than from traditional employees or suppliers.” [148]. “Key differences are the fact that users lacking formal training in map making create the geospatial data themselves rather than relying on professional services; that potentially very large user groups collaborate voluntarily and often without financial compensation with the result that at a very low monetary cost open datasets become available and that mapping and change detection occur in real time.” [149]. While most internet mapping providers such as MapQuest or Bing Maps make use of user edits to individual existing map features, other systems are primarily based on crowdsourcing.

Web 2.0 initiatives such as Open Street Map (OSM, 2004), WikiMapia (2006) or Google MapMaker (2008) have put a strong stake in the ground in this domain by creating wiki-enabled frameworks for collaborative mapping. A significant difference between MapMaker and the other two sites, is that Google uses the community created data obtained via MapMaker in order to improve its commercial Google Maps platform. In case of Open Street Map and WikiMapia, the created data are made available under a Creative Commons [150] share-alike license for non-commercial use and thus remains public domain. Data entry can either be performed using the graphical web interface or by uploading GPS traces recorded offline. Aerial images by Bing Maps or Google Maps as a base layer serve as guidance for the operator when entering vector data.

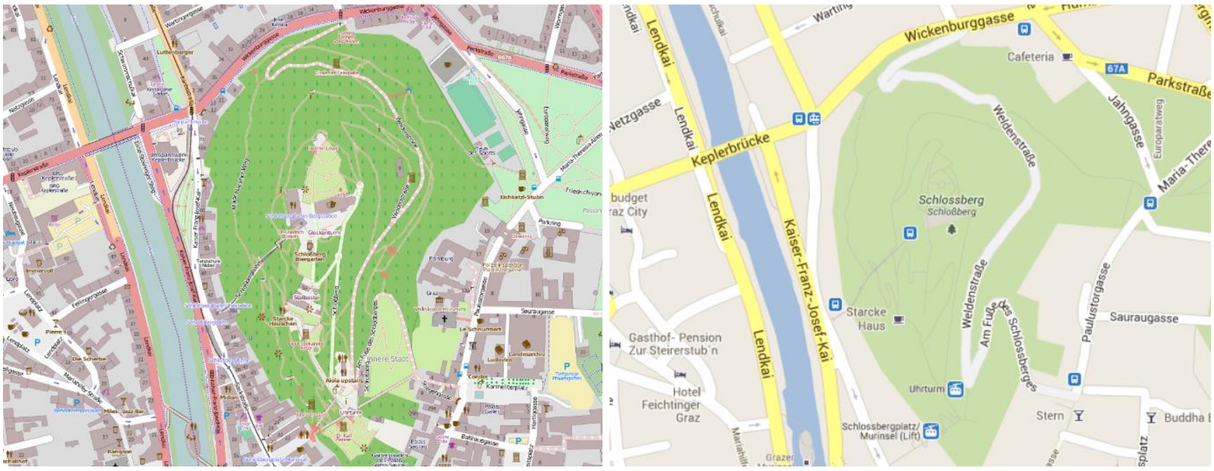


Figure 2.18 Sample Vector Data from Open Street Map Showing the Schlossberg Mountain in Graz, Austria (Left) compared to the same area in Google Maps. (Right)

As pointed out by [103] the vector data created in OSM are often comparable if not superior to commercially collected data. Figure 2.18 illustrates this by showing the same map region from OSM and Google Maps. A study of the accuracy and completeness of OSM data compared to Ordnance Survey data in Great Britain found that OSM information "…was on average within 6 m of the position recorded by the Ordnance Survey, with approximately 80% overlap of motorways between the two datasets." [151] It also found a significant gap in completeness of area, as OSM had captured about 29% of the area of England within about 4 years, with some obvious untouched regions, presumably due to the lack of "volunteer mappers" in that area. A newer comparative study [152] of the accuracy between Google Maps, Bing Maps and OSM for Ireland concluded that while each site showed individual differences, there was no clear winner between the services.

2.6 Research Areas to Advance Internet Mapping

The technical advancements in the area of internet based mapping described above, have fundamentally transformed how people use mapping information, how it gets created and who owns and controls it. The fact that most of this development has happened in less than a decade, in an environment driven by fast technological and economic change, implies that certain unresolved challenges still remain to be solved. Peterson [29] describes a list of such challenges related to both the internet in general as well as internet mapping in particular.

2.6.1 Data Capture

A big challenge for any company entering the internet mapping business at a global scale is the need to initially obtain a sufficiently large dataset of both vector data such as street maps or POI, as well as image data such as aerial, satellite or streetside imagery. While considerable data can be licensed from providers such as NavTeq or Digital Globe, some kind of data such as human scale imagery is not commercially available at a global scale. This creates the necessity for internet mapping firms to operate a fleet of cars equipped with specifically designed camera systems, and to collect crowdsourced data from internet sites. The field of streetside mapping is new and few commercial systems exist which fulfill the requirements with respect to data quality as well as automation during capture and processing. Such requirements include the ability to capture LiDAR depth data in combination with imagery, precisely geocoded using GPS based navigation systems. Hence we propose a novel design for a streetside capture system in Section 4.1 which fulfills the stated quality and automation requirements.

2.6.2 Privacy

One of the primary concerns related to geospatial information is the privacy of individuals who are either using new technologies such as LBS, or people whose privacy is imperiled due to the data capture methods used by providers. Data privacy for LBS in order to anonymize and obfuscate location information is an open research area. The problem of anonymizing people and other private objects in human scale imagery is challenging due to the variability and enormous quantities of such images. In Section 4.5 we propose a novel solution for this problem.

2.6.3 Cost and Ownership

While data licensing and targeted data collection are a necessity for internet mapping providers, they also entail substantial investments. Very few companies have the required financial assets to pass this hurdle, especially in a market practically “owned” by strong existing players such as Google or MapQuest. Additionally the fact that strategically invaluable map data are owned by few commercial entities not only puts these entities in a strong position, but also poses a risk to map users of becoming overly dependent on individual businesses. Hence a democratization of map data owned by many, such as open-source maps obtained by crowdsourcing would be strongly desirable. Non-profit systems such as OSM have made substantial progress by obtaining public domain vector data. However, they are lacking the financial assets required for image data acquisition and hence depend on aerial and satellite imagery from other sources.

In order to support the democratization of image data assets and to alleviate the collection cost, we therefore propose several methods of using open source images from Community Photo Collections (CPC) in the context of mapping. In Section 5.3.2 we describe a geographic image matching framework which is used as the basis for solving several related problems. This framework is used in Section 6.1.6 by connecting community created human scale images with systematically collected data. Additionally we propose an improved method based on [14], to automatically align point clouds derived from shape-from-stereo to an overhead aerial image of the same scene (Section 6.2.3).

2.6.4 Accuracy and Reliability

Another challenge results from the high expectations users have with respect to the accuracy and reliability of internet maps. These expectations are partly caused by people's experience with paper maps and the trust they developed with respect to information from the internet in general. Due to the fact paper maps were prepared mostly by professional cartographers in cooperation with governmental institutions, their accuracy level has been relatively high. Internet maps are created by merging information from a multitude of sources, and are usually considerably more detailed. For example they also contain geocoding information for street addresses, and points of interest. Erroneous address information in a navigation scenario will inevitably lead to user frustration. To reduce the likelihood of such problems, we propose an image-matching based method for correcting errors in business-geocoding in Section 6.2.3.

2.6.5 Freshness of Images

Aerial, satellite and terrestrial image sensor technologies have advanced towards the capture of Petabytes of image data at a global scale annually. Nevertheless cost limits the frequency at which it can be recaptured to reflect recent geographic changes. Particularly human scale images become stale after a certain amount of time, as changes can more easily be noticed in images showing details of streets. In order to counteract staleness, and improve the freshness of data, it is therefore desirable to use other more frequently updated data sources such as community images in addition to systematically collected images. In Section 6.1.6 we show that by augmenting streetside images with community photos of the same location, we can improve the freshness of the data presented to the user.

2.6.6 Infrastructure

As pointed out in Section 2.2.1 a digital divide between the developed and developing countries in the world exists with respect to the availability of computers and mobile phones as well as networking infrastructure. Due to this fact, and since new networking technologies such as LTE are only available in limited geographic areas even in more developed countries, network bandwidth is still a challenge for mobile mapping applications. Hence when building new technologies such as augmented reality applications, minimizing the data transmission requirements is an ongoing concern. We therefore analyze the influence of different image file sizes which have to be uploaded from a client device to a web service on the performance of the image matching system described in 5.3.2 to find an optimal tradeoff setting.

2.6.7 Geopositioning Accuracy

Mobile augmented reality applications such as the example shown in Figure 2.13 primarily use a combination of GPS, cell phone tower and WIFI hot-spot triangulation to obtain a position estimate, and a magnetic compass to obtain an estimate of the viewing direction. Since the accuracy of such positioning sensors is often limited to several meters in rural areas and tens of meters in urban environments, it may be insufficient for precise navigational instructions or augmentations. For some applications (e.g. indicating the entrance of a business) pixel-accurate instructions would be preferable. For that reason we proposed an image based method for obtaining a more precise registration of cell phone images with existing geospatial images such as streetside panoramas, allowing pixel-precise registration of relevant contents

2.7 Summary and Outlook

In this chapter, we have provided answers to research questions I.a through I.d by documenting the evolution of cartography from hand-drawn and printed maps, via the first digital geographic information systems, to the location-aware and ubiquitous internet of today.

We have defined maps as a visual representation of the (spatial) relationship between multiple elements within an area, such as objects, regions and themes. Many of the elements used in internet mapping today have evolved from prior achievements in mapping, which we have pointed out.

Further we have described the main data types used in digital maps such as vector data and rater data, as well as the various data sources, including national mapping agencies, commercial data providers and crowdsourcing.

Finally we have pointed out some of the major research areas in internet mapping, and how they relate to the main contributions of this report.

Internet maps are being used by more than **1 billion** unique users every month [83], on a variety of device form factors, for applications such as personal navigation, weather queries, mobile location based services and for Augmented Reality.

3 Categories of Geospatial Image Data

3.1 Introduction

Image maps have become an accepted data type with the advent of digital image processing. The aerial orthophoto is the preferred standardized image map widely discussed in mapping text books [153]. With the appearance of web based mapping services, images have become very diverse in various formats and scales [3], and are now an integral part of mapping services offered by providers such as MapQuest, Microsoft, Google, Nokia or Apple. This results primarily from the fact that images provide a more natural and visually pleasing experience compared to vector maps as illustrated in Figure 3.1. We are here referring to all sorts of amateur, hand-held camera photographs, as well as images taken from vehicles, flying drones and airplanes.

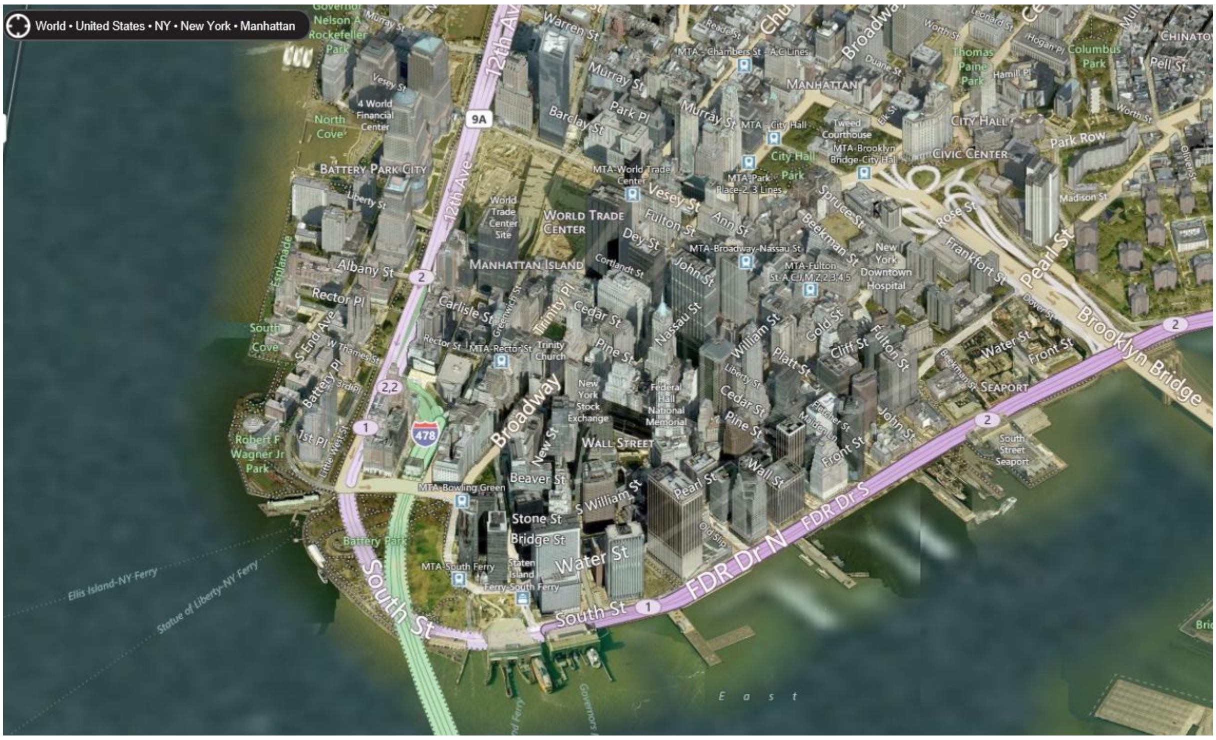


Figure 3.1 Aerial “Birds-Eye” View of Manhattan Island, New York in Bing Maps

We would like to evaluate the criteria by which images get selected for internet mapping. In this context we are interested in understanding what defines an image, and what makes it “geospatial”. We further want to distinguish between source image data and a “map” created from and containing images of various types.

3.1.1 Defining Geospatial Images

In general we care mostly about "images", "pictures" and "photographs" as "visible impressions obtained by a camera, telescope, microscope, or other device, or displayed on a computer or video screen" as defined in the Oxford Dictionary [77]. However we also want to include the results of different kinds of processing of the source images captured by sensors, such as the result of "stitching" [154] multiple source images into a continuous representation. We use the term "geospatial" also in its lexical definition as "relating to or denoting data that is associated with a particular location" [77]. Hence we consider images featuring objects associated with a particular location as "geospatial images" in contrast to those containing only location agnostic contents, and "geospatial relevance" as the degree at which images represent a given location.

In other words, we consider an image that a familiar viewer would likely relate with a particular location or geographic region (e.g. a building, statue, mountain or pond) as more relevant in the context of mapping than a generic image (e.g. of a person, car, food item or book).

This distinction largely agrees with the definition of "geographic relevance" of information in the context of LBS by Reichenbacher et al., who define it as a "quality [that] is expressed as the relation between an entity or its representation (e.g. image) and the actual context of using the representation." [155]

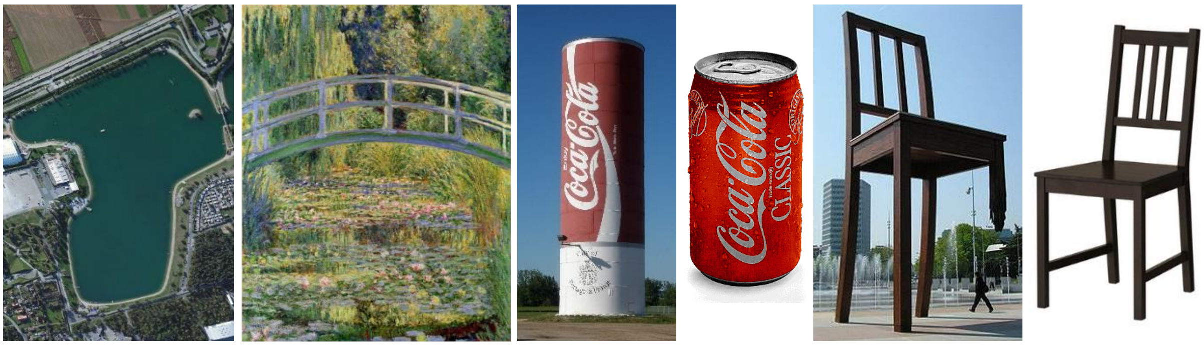


Figure 3.2 Examples of Geospatially Relevant and Irrelevant Images: Schwarzl See in Kalsdorf, Austria [57]; "The Japanese Bridge" Painting by Claude Monet [156]; Water Tower in Portage la Prairie, Canada [157]; Coca-Cola can; "Broken Chair" Sculpture in Geneva, Switzerland [158]; Ikea Chair;

To understand this distinction better we analyze a few specific examples. An aerial photography of a pond and its surroundings (Figure 3.2, left) would most-certainly be considered geospatial, similar to a streetside image of a building façade or a picture of a statue in a particular church. However a picture of the Claude Monet's "The Japanese Bridge" painting showing a pond may only be considered geospatial in the context of the museum where it is exposed (National Gallery, London). The water tower painted as a Coke-can in Figure 3.2 is clearly associated with a geographic location (Portage la Prairie, Canada) and is therefore geospatially relevant. However an image showing a generic object (e.g. a Coke-can) may be geographically relevant only if the object occurs more frequently at a given location than elsewhere or if people commonly relate this object with a location. The same reasoning also applies for the Ikea-chair and "Broken Chair" sculpture example. However one could argue that a piece of furniture can be associated with a location within a house (e.g. kitchen) and would be visible in imagery showing that room. Hence the distinction also depends on the scale at which one looks at a geographic location and whether or not an object is photographed by itself or within the context of its environment.

While we have provided a definition of geographic relevance of images, Epshtein et al. [159] analyzed the extraction of the relevance of geographic locations from a collection of community images. For this purpose they use a different definition of geo-relevance as the importance of a spatial point, proportional to the number of images showing that point. This definition still agrees with ours in the sense that location relevance can only be extracted from relevant images.

3.1.2 Image Selection Criteria for Internet Mapping

As detailed above, in order to be considered relevant for internet mapping, images should exhibit contents which are relevant for the specific location at which they are captured. Apart from geo-relevance, we found several other selection criteria which can be used by mapping providers:

- Comprehensiveness
  Mapping providers are faced with the challenge of obtaining global image coverage by integrating data from multiple sources. Hence, one of the selection criteria used is the data comprehensiveness, such as their broad and continuous regional or global availability. An example for comprehensive image data is the "global basemap" offered by DigitalGlobe for all 500 million km^2 on the planet [7].
- Freshness
  While historic images may provide specific value in certain situations and to some people, mapping usually requires the data to be up-to-date with respect to the actual appearance of the world's locations. Therefore freshness of the data is considered an advantage over stale images no longer reflecting the geographic reality. Especially in situations requiring immediate action, such as natural disasters [160], up-to-date map data is essential, such as provided by Bing Maps in response to Hurricane Sandy [161].
- Quality
  The same quality measures as for digital images in general [162] apply also for geospatial applications. Primarily these are geometric properties such as the image resolution (ground sample distance, GSD) and the geometric accuracy, as well as radiometric properties such as signal to noise ratio, dynamic range, color consistency and the bit-depth used for storing an image.
- Explorability and Discoverability
  We require that image data can be easily viewed and explored. This is characterized by the fact that they must be discoverable by users interested in a given geographic region, and that navigation in-between multiple images must be possible without significant effort. For example Photosynth [163] allows intuitive discovery and exploration of georeferenced image data via Bing Maps.
- Geo-Positioning
  A primary requirement for achieving discoverability is the availability of geo-positioning data for an image, thus the camera pose, which may consist of a Latitude/Longitude

coordinate pair in a global coordinate system (e.g. WGS84), or additional parameters such as the altitude or angular orientation [134]. The positional accuracy requirements depend on the scale at which images are viewed. As Photosynth [163] demonstrates, pixel accurate registration is feasible and sufficient to achieve discoverability.

- Privacy Compliance

  An important aspect of photography in general is the protection of people's rights to privacy, whether they are in a private setting or in public environments such as streets, parks or shopping malls. The release of streetside imagery by Google and Microsoft has raised privacy concerns in many countries worldwide, particularly in Germany [164, 165]. Therefore it is essential for mapping service providers to put protective measures in place to guarantee that privacy requirements are met.

3.1.3 Systematically Collected Images vs. Crowdsourcing

We are presenting here various types of geospatial image data that are being used for online mapping services, along with their advantages and disadvantages with respect to the above criteria. We propose the classification of geospatial imagery in three major categories, based on the capture processes and patterns used for data collection.

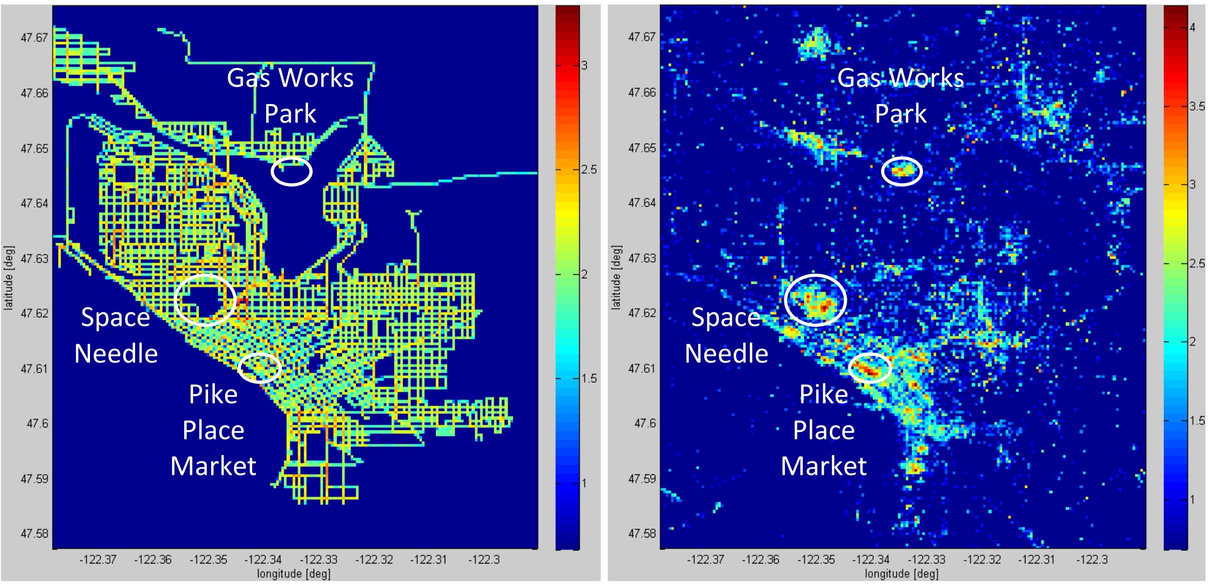


Figure 3.3 Distribution of Systematically Captured (Streetside) and Crowd-Sourced (Flickr) Images for the Same Area in Seattle, WA; Colors show Log_{10} of the Image Density per 100·100 m^2 Tile;

The *first category* are systematically collected images which are obtained by following a clearly defined procedure and capture pattern, so as to achieve sufficiently dense and up-to-date global coverage of Petabytes of images. These capture patterns are under control by an entity such as a mapping provider or a data supplier such as DigitalGlobe, NASA, GeoEye, Pictometry, TeleAtlas or NavTeq. Systematical image collection for online mapping services usually applies to vertical and oblique aerial imagery, providing a top-down view of the Earth, or more recently human scale imagery taken from a vehicle on the ground. While the specifications of these data types differ in many ways, the capture process follows similar rules, such as systematic and clearly defined geographic pattern to assure continuous coverage at the desired geometric and radiometric quality and overlaps.

The usage of systematically captured image data is accompanied by a *second category*, crowdsourced images uploaded by people around the world to online community photo collections such as Flickr [166], PhotoBucket [167], Panoramio [168] or Facebook [169]. In this case the capture process is mostly done by individuals for either their own benefit, to share with the community, or to achieve a certain task. It usually does not follow rules or specifications provided by a particular commercial entity, but rather represents the collective mindset of a community of people. While systematic data acquisition, capture and processing often incurs significant cost crowdsourced data is generally provided for free.

The *third category* are semi-systematically collected images, hybrid forms of the prior two categories. This category exhibits certain attributes of both systematic and non-systematic data collection. An examples of semi-systematic image capture are organized programs for geographically relevant community photo collection such as Foursquare, Yelp or Redfin.

A comparison of typical geographic distributions for systematic and crowd-sourced data in Figure 3.3 indicates their complementary nature. While streetside images on Bing Maps (left) systematically cover the road-network of a city (e.g. Seattle) with a nearly constant image density (~160 images / 100·100 m^2 tile), crowd-sourced images from Flickr (right) are distributed irregularly based on popularity. Flickr images are often clustered around hot-spots (e.g. > 10,000 images per 100·100 m^2 tile) around popular landmarks, such as “Pike Place Market”, “Gas Works Park” or “Space Needle”, while other areas remain unpopulated. Note that the latter two of these hot-spots actually don’t contain any streetside imagery, as they are not accessible by car. By combining the two asset types in mapping, one can thus benefit from the continuous coverage of systematic data as well as the popularity based distribution obtained from crowd-sourced data.

3.1.4 Source Images and Processed Data

We also distinguish between source image data directly captured by a sensor, and processed data products used in the context of internet mapping. Various levels of processing have to be considered, such as radiometric and geometric preprocessing of individual images, as well as the combination of multiple images into photogrammetric data products.

In case of the UltraCam aerial camera system described in Section 3.3, raw images from individual sensors (Level-0) are radiometrically corrected (Level-1) and then geometrically stitched into a single continuous coordinate system (Level-2) containing separate panchromatic high-resolution and multispectral low-resolution images. From this data, Level-3 images in multiple color formats can be generated, such as Panchromatic, RGB, or false color infrared [97]. The processing workflow is detailed in Section 3.3. The procedure to convert human scale imagery from Level-0 to Level-2 and into panoramas, such as the UltraCam-M system explained in Sections 4.1 through 4.4 is analogous to aerial images.

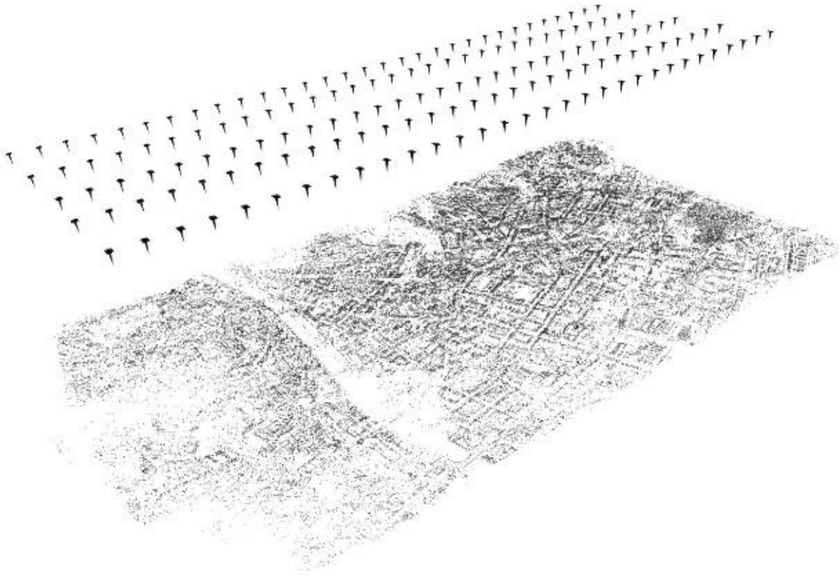


Figure 3.4 Aero-triangulation of Thousands of Camera Locations and 3D Coordinates of Tie Points is an Essential Component of Automatic Photogrammetric Workflows [170]

Based on the preprocessed aerial imagery, various workflows allow the automatic photogrammetric processing into **2D** orthophotos, **2.5D** digital surface models and **3D** point clouds and meshes of the scene geometry, including buildings, trees, bridges etc. An initial step during this process is the automatic "aero-triangulation" (AT) [170] of 2D image correspondences into thousands of camera poses and sparse 3D point clouds (Figure 3.4) as a basis for further processing of aerial imagery.

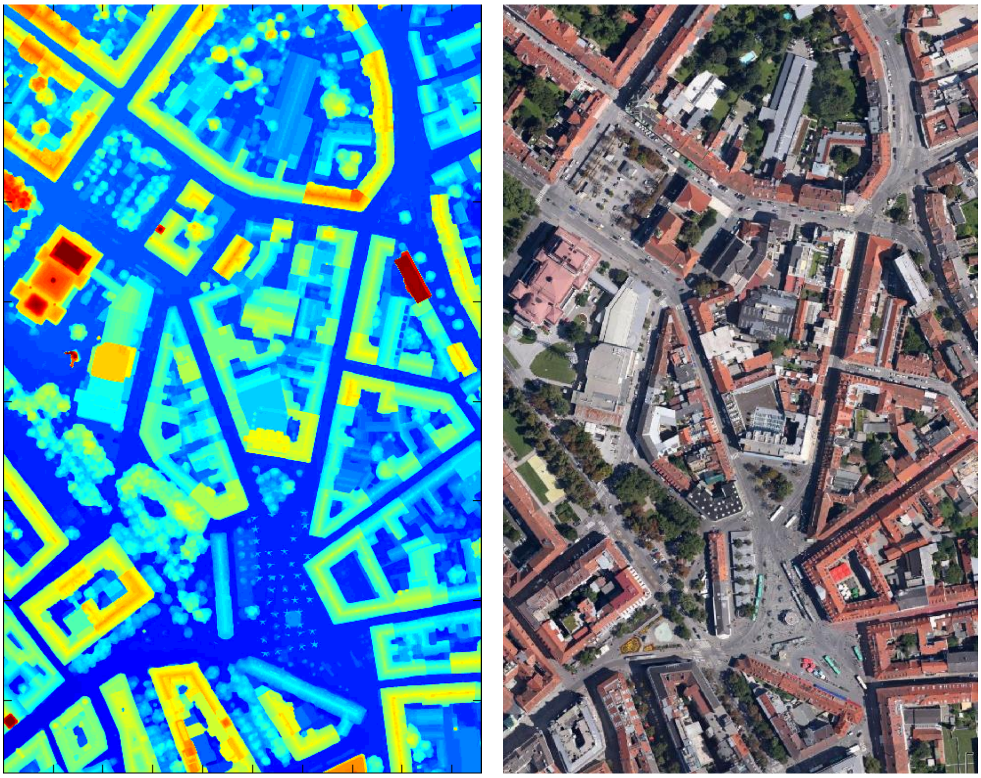


Figure 3.5 Digital Surface Model at 8 cm GSD (Left) and Aerial True Orthophoto at 30 cm GSD (Right); Location: Downtown Graz; [57, 59]

Digital surface models (DSM) are raster images with a specific ground sample distance (GSD) indicating the height of the surface for each cell, thus providing 2.5D data (See example in Figure 3.5, left). In contrast to digital terrain models (DTM) which are a "bald-Earth" representation, DSM include man-made structures as well as vegetation ranging out from the terrain [170]. DSM and DTM are frequently generated using either aerial LiDAR systems as described in Section 3.2.4 or via multi-ray dense matching techniques from aerial imagery [171, 172, 173, 174, 175]. Obtaining DTM from both LiDAR and aerial DSM requires additional filtering to remove buildings and vegetation, which is provided by automatic software tools such as [176].

By using the depth information from DSM along with pixel data from individual raw images, automatic workflows [177, 178, 179, 176] generate synthetic top-down views called "true orthophotos" (Figure 3.5, right). Four steps are usually required to obtain orthophotos: Rectification, color matching, mosaicking and feathering [180]. While the first step transforms input data geometrically into the output format, the remaining steps result in radiometric consistency across a mosaic [180] of input images such as by means of the graph-cut algorithm [181].

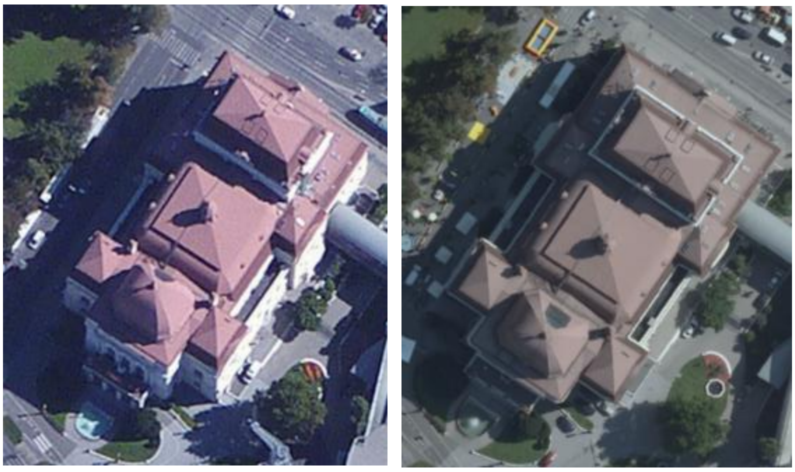


Figure 3.6 Comparison of Building (Graz Opera House) in Classical Orthophoto (Left) and True Orthophoto (Right); [57]

The term "true" relates to the circumstance that actual 3D geometry is used to generate precise orthogonal projections, rather than rectifying and stitching individual images into classical orthophotos. As a consequence vertical building facades do not appear in true orthophotos, as they do in classical orthophotos under varying angles (See comparison in Figure 3.6).

By using true orthophotos to "paint" geo-aligned DSM, more "plastic" views of environments (Figure 3.7) can be obtained than by simply viewing 2D images. Such views can be navigated in 3D and rendered from arbitrary viewpoints. However, 2.5D models of this kind lack any non-convex geometric details of buildings and other structures, as well as texture information for the building facades. They also don't represent individual objects such as buildings or vegetation separate from the underlying terrain.

Therefore alternative approaches [182, 183, 184] obtain actual 3D object models from LiDAR data or imagery based on the output of similar dense matching methods as mentioned above. Building roofs with chimneys, dormer windows, and skylights may further get detected and segmented [185].

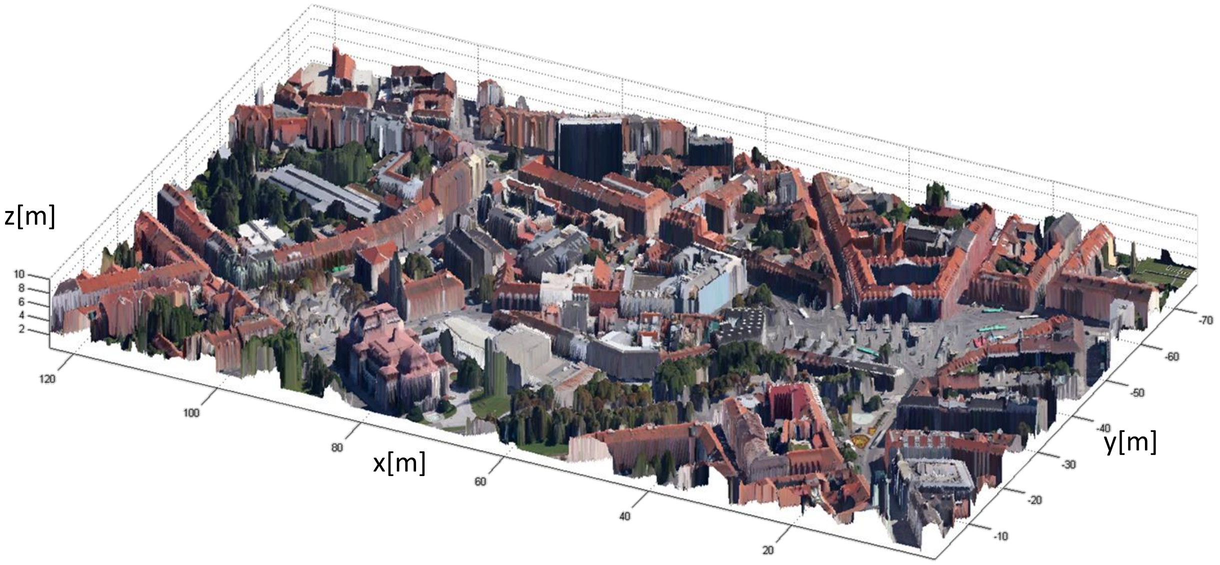


Figure 3.7 2.5D Model Generated from DSM and True Orthophoto

Figure 3.8 gives an example of an automatically created 3D city model from Microsoft Virtual Earth containing individual 3D meshes of buildings which were textured from the raw aerial imagery. Additionally, natural objects such as trees or bushes may be automatically detected and replaced by synthetic models at a higher level of detail [186].

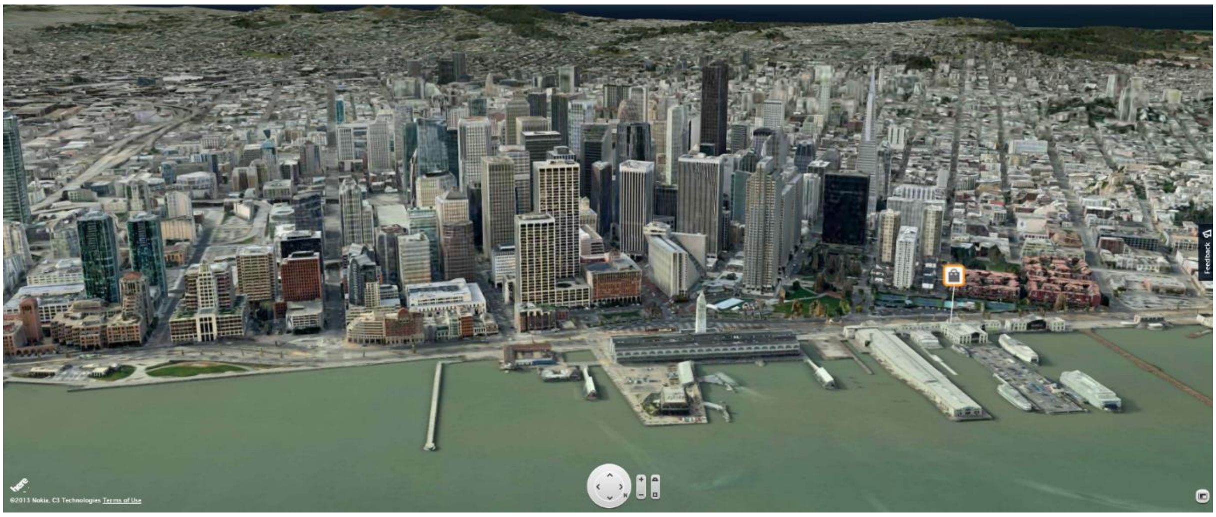


Figure 3.8 3D City Model of San Francisco Automatically Generated from Aerial Images [60]

3.2 Aerial and Space Images

The most common image representation used on web based mapping sites are images showing the surface of the planet in a top-down fashion, by using orthographic projections. This view is more natural, though similar to classical vector based maps in that it represents a 2D image, navigable by zooming, panning or rotation (often in 90-degree steps). Additionally, it can be augmented with traditional vector data such as street networks, building outlines, water bodies, terrain classifications, etc. or other geographically relevant information (e.g. precipitation statistics or population density) as part of a geographic information system (GIS) [68].

Primarily two types of top-down imagery are used by online mapping sites, aerial and satellite imagery. They are taken using specialized cameras carried on Earth observation satellites (EOS) or aircrafts. In addition to classical nadir orthophotography, several providers of web based mapping services also feature oblique aerial imagery in certain regions, providing slanted views at various angles (e.g. 20-60°).

A continuous image representation of the world's geography often requires the integration of a multitude of image sources and types in huge volumes. In fact, image data covering only the land part of the Earth's surface, spanning **149 million km^2** at **15 cm GSD** results in 5.88 Petapixel of net data. As automatic processing workflows often require image capture at 80% forward overlap and 60% side overlap for **10-15** times redundancy [171], the raw data adds up to **220 Petabytes** of 8-bit RGB pixel. For a particular aerial camera (UltraCam Eagle [187]), a total of **316 million** images are required, each with a ground coverage of 5.88 km^2. This estimate roughly agrees with [1] who estimated 190 Petabytes with slightly different assumptions about the redundancy.

Various types of top-down image data are provided by a plethora of aerial survey companies, public organizations and satellite photography providers, using a variety of different sensors.

Figure 3.9 GeoEye-1 Satellite (Artist's View) [6]

3.2.1 Satellite Imagery

Earth observation satellites (EOS) comprise one of the major sources for top-down imagery used in internet mapping. The capture pattern of satellite sensors is systematic, as satellite orbits are continuous and totally pre-defined. Satellites sensors can be directed to different viewing directions though to capture different swaths of the surface. Most EOS such as GeoEye-1 use linear image sensors, which can scan one or multiple lines spanning many pixels across (37,500 in case of GeoEye-1) at high frame rates (10,000 lines per second) and relatively small GSD (0.41 m). Line sensors, which are arranged perpendicular to the satellite's trajectory scan a continuous image swath for each orbit revolution, in a process called "push-broom scanning" [188]. Multiple revolutions allow the capture of multiple swaths, which can hence be stitched into a single image. As individual swaths span widths up to tens of kilometers (15.2 km in case of GeoEye-1), large areas up to the size of Texas can be captured during a single day [6].

Often, satellites contain sensors to capture panchromatic as well as multispectral imagery, of which typically only the visible spectral bands are visualized on online maps. In order to reduce the sensor complexity and data volumes, the multispectral imagery is often captured at a lower resolution (larger GSD) than the panchromatic images. This requires "Pan-Sharpening" to combine high-resolution panchromatic and low–resolution multispectral images into a high-resolution color image. Example methods for pan-sharpening of satellite images have been proposed in [189] and [190], while a similar approach for aerial images was presented by [191]. An overview of various EOS with GSD below 5 m and thus suitable for medium to high resolution image capture is provided in Table 3-1, sorted by increasing GSD and by launch date. While GeoEye-1 already achieves a GSD of 0.41 m, US government regulations require resampling to **0.5 m GSD** for commercial use, such as for internet mapping [192]. The same limitations will likely also apply to future higher-resolution satellites such as GeoEye-2 and WorldView-3.

Satellite Name	Operated by	Country	Launch Year	Panchromatic GSD [m]	Multispectral GSD [m]	Swath Width [km]
Worldview-3	DigitalGlobe	USA	Future	0.31[1]	1.24	13.1
GeoEye-2	DigitalGlobe	USA	Future	0.34[1]	1.36	14.5
GeoEye-1	DigitalGlobe	USA	2008	0.41[1]	1.65	15.2
Worldview-2	DigitalGlobe	USA	2009	0.46[1]	1.84	16.4
Worldview-1	DigitalGlobe	USA	2007	0.5	-	17.6
Pleiades-1A	Spot Images	France	2011	0.5	0.5	20
Pleiades-1B	Spot Images	France	2012	0.5	0.5	20
Quickbird	DigitalGlobe	USA	2001	0.61	2.44	16.5
Eros B	ISA	Israel	2006	0.7	-	7
IKONOS	DigitalGlobe	USA	1999	0.82	3.2	11.3
Eros A	ISA	Israel	2000	1.8	-	14
SPOT-6	Spot Images	France	2012	1.5	8	60
Formosat-2	NSPO	Taiwan	2005	2	8	24
SPOT-5	Spot Images	France	2002	2.5	10	60
Cartosat-1	ISRO	India	2005	2.5	-	30
ALOS	Pasco	Japan	2006	2.5	10	70

Table 3-1 Overview of Earth Observation Satellites with Panchromatic GSD below 5 m; [192, 193]

[1] Image GSD is restricted by US government to 0.5 m or above for commercial use.

Even though the field of view (FOV) of the line cameras usually faces the nadir direction, some satellite sensors additionally capture oblique forward and backward looking swaths. This generates additional images with a stereo parallax. Hence the data can be used for depth reconstruction to obtain digital surface models (DSM) or digital terrain models (DTM), which in turn are needed to produce ortho-imagery [194]. Finally satellite images get transmitted to ground stations, where they are stored, geo-processed and transmitted further for more processing [195].

In contrast to the frame sensors used for aerial photography, due to the nature of pushbroom scanning, each image line has a different capture time and thus exterior orientation (EO). Although the trajectories of satellites are significantly smoother than those of airplanes, any deviations from an ideal path will lead to distortions in the collected imagery which have to be modelled and corrected during postprocessing. Multiple methods exist to correct for such distortions, such as by measuring the various linear and angular motions using inertial sensors, and compensating for these effects during the stitching process. Further correction methods may involve stitching multiple swaths by using feature matching to determine distortions occurred in each swath, or via registration to existing geocoded orthophotos [196, 197].

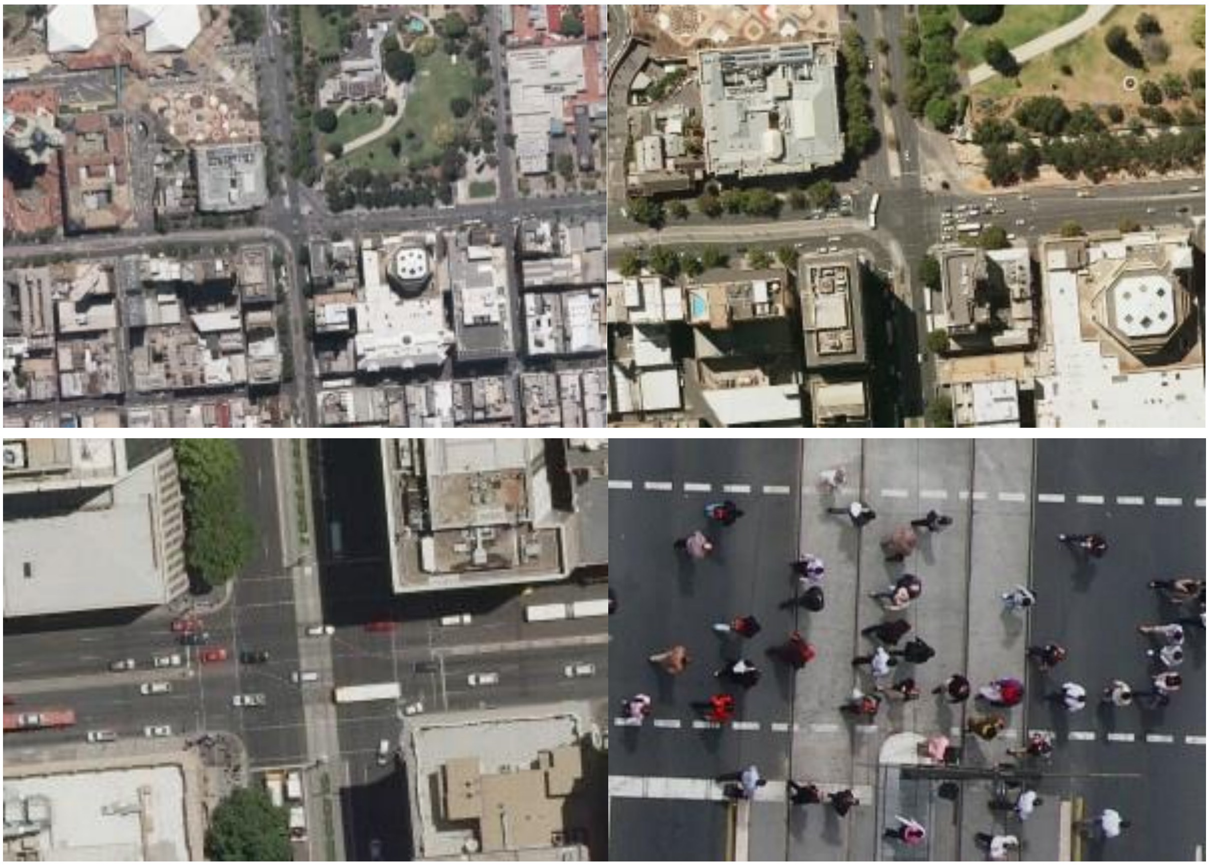


Figure 3.10 Comparison of Satellite and Aerial Imagery of the Same Location: IKONOS Image @ 1 m GSD (Top Left); WorldView-2 Image @ 0.5 m GSD (Top Right); UltraCam-D Image @ 20 cm GSD (Bottom Left); UltraCam-X Image @ 2.5 cm GSD (Bottom Right); [198]

Satellite images don’t look fundamentally different from aerial imagery, and are often presented to users in one and the same map view. Differences may exist in image quality due to atmospheric effects and short exposure times (≤0.1 ms) in satellites versus perhaps 8 ms exposure from the air. A primary drawback of satellite imagery for commercial applications consists of limited GSDs at **0.5 m** per pixel versus **2.5 cm** from the air [187]. Aerial cameras get operated at optimized flying altitudes for best GSD, and for optimum 3D resolution, satellite image overlaps for automated 3D analysis are not easily obtained.

Since the operating altitudes of satellites are substantially higher (e.g. **684 km** for GeoEye-1 vs. **2 km** for UltraCam Eagle for 10 cm GSD), the visibility of the planet's surface from the satellite is often occluded by clouds. Due to the rigid satellite trajectories, recapturing an affected region at a desired view angle is often infeasible, which poses challenges to obtaining continuous high quality images of extended regions such as countries within a single season [199]. However the limited availability of surveying technology and infrastructure in some countries, as well as political regulations related to the capture and publication of aerial imagery often leave satellite imagery as the only feasible option for obtaining nadir coverage [112].

3.2.2 Vertical Aerial Imagery

Vertical aerial photography is the work-horse for precision to perform mapping at accuracies of **1:10,000** geometry - that is with errors as 1/10,000 of the flying height [200]. Similar to satellite images, aerial images are captured using specifically developed aerial sensor systems, carried by airplanes. Approximately **1,000** large format cameras **(≥90 Megapixel)** are operated by a multitude of aerial survey companies [201] and public organizations worldwide [69], besides numerous medium format (**30..90 Megapixel**) and small format (**20..30 Megapixel**) cameras [202]. Aerial images are very similar to satellite images in that they show the surface of the Earth in a top-down view. Apart from the capture process itself, the major differences are related to the improved image radiometry, available detail at GSD values of **2.5 cm** to **20 cm**, and more pronounced overlaps for automatic 3D information extraction.

Small GSD of 2.5 cm or less are achievable with aerial cameras by using long focal lengths and flying low, at perhaps 500 m above the ground. The stark difference between 0.5 m "satellite GSD" and 2.5 cm "aerial GSD" is visualized in Figure 3.10. Typical GSD values used for **online** maps of urban areas range from **15 cm** to **30 cm**, which currently can only be achieved by means of aerial imagery.

The pattern in which aerial imagery is captured generally follows a set of parallel flight lines (Figure 3.11), covering an area in a systematically, similar to streetside imagery as shown above.

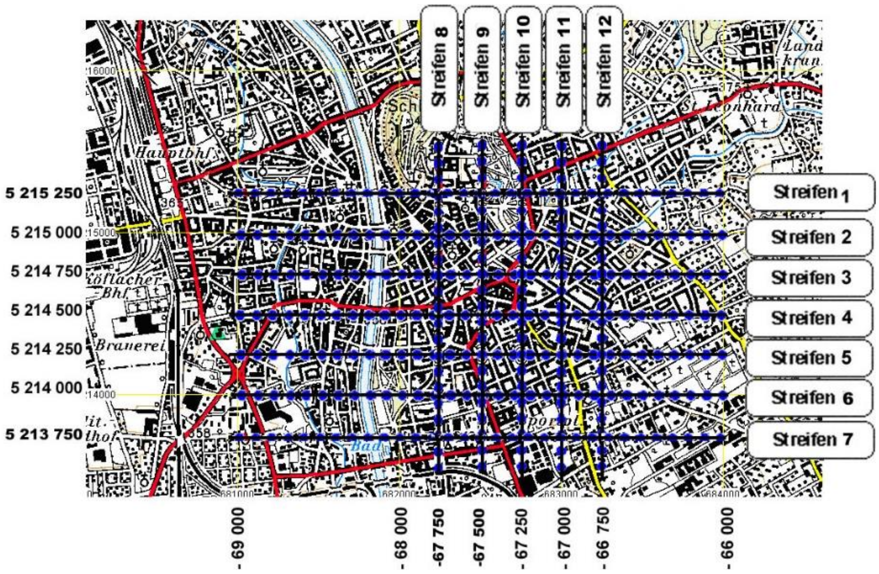


Figure 3.11 Sample Flight Plan for Aerial Image Capture

While aerial image data are usually acquired and captured on a per area basis, large aerial survey companies offer existing data assets such as stitched orthophotos or DSM of entire cities, counties or even states for sale. Such data assets are already in the correct format to be included in online mapping systems. This is useful especially for establishing an initial coverage while scaling up a service within a short amount of time. Nevertheless, acquiring data from many different suppliers, using various sensors and data formats can also be cumbersome, and sub-optimal in terms of the data acquisition cost. Therefore, companies such as Microsoft or Google have spent significant efforts to establish their own infrastructure and workflows for data capture in order to increase control over data quality, consistency and cost efficiency. Microsoft acquired Vexcel Corp. in 2006 to support the development of Virtual Earth (now Bing Maps) by means of its sensor and mapping expertise [203]. About a year later Google acquired aerial camera manufacturer ImageAmerica to similarly support its efforts to capture aerial imagery for Google Maps [204].

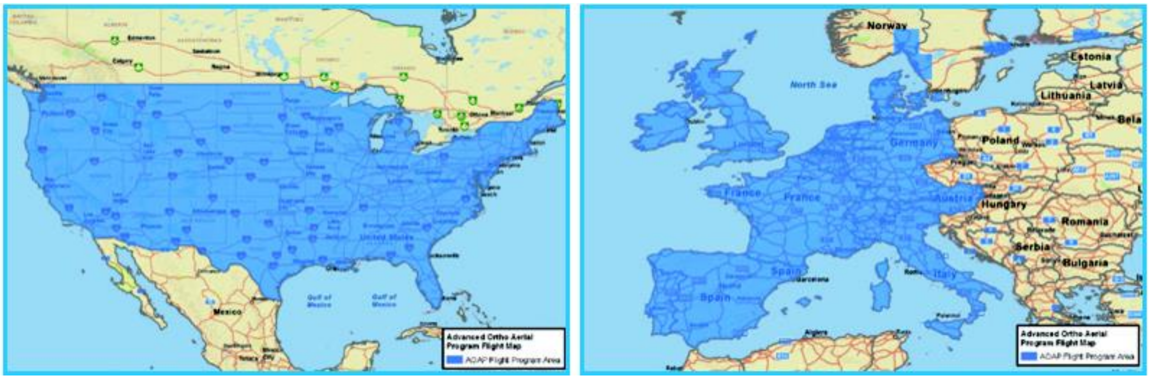


Figure 3.12 Areas Covered by „Microsoft Global Ortho Project“: The Entire Lower 48 US States (Left) and Parts of Alaksa; 14 European Countries (Right);

Microsoft recently completed one of the largest aerial image capture project ever undertaken by creating continuous color orthophoto imagery of the whole area of the continental United States as well as 14 European countries at a consistent GSD of 30 cm. To allow image capture at this **10 million km^2** and **100 Terapixel** scale within a time frame of only about two years (2010 – 2012), a special sensor system (UltraCam-G) was developed by Microsoft Vexcel. Multiple systems of this and other camera types of the UltraCam series were operated in cooperation with multiple aerial survey firms in various regions to achieve this goal. Since data freshness is also a concern for mapping services, in order to reflect recent changes in the geography and man-made structures due to new constructions or natural disasters, 60% of this data will be re-captured within a 2.5 year timeframe. Data from this "Global Ortho Project" have been made available to users via Bing Maps as well as for commercial use through a partnership with DigitalGlobe [5, 205].

3.2.3 A look back

Aerial photography has been acquired essentially since the invention of photography, balloons and airplanes [206]. While originally images were taken simply to document the appearance of objects from the air, this development eventually led to the invention of aerial photogrammetry as an important tool to determine geometric properties of captured scenes, as well as for large area mapping in order to generate topographic maps.

Until about 2003, aerial cameras used film as the medium for capturing and storing imagery which had to be processed using analog equipment. A transition to digital processing started around 1990 as scanning film and processing pixels on increasingly powerful computers led to photogrammetric measurements. Photogrammetric scanners at +/- 2µm geometric accuracy (e.g. Vexcel VC4000 and Vexcel UltraScan 5000) were invented, which served to digitize the analog film-based imagery into 11,000*11,000 pixel arrays at 20 µm pixel size in the film plane [207]. Hence, due to advancements in digital image processing, computer vision and digital photogrammetry the digital images could be processed using photogrammetric software products directly.

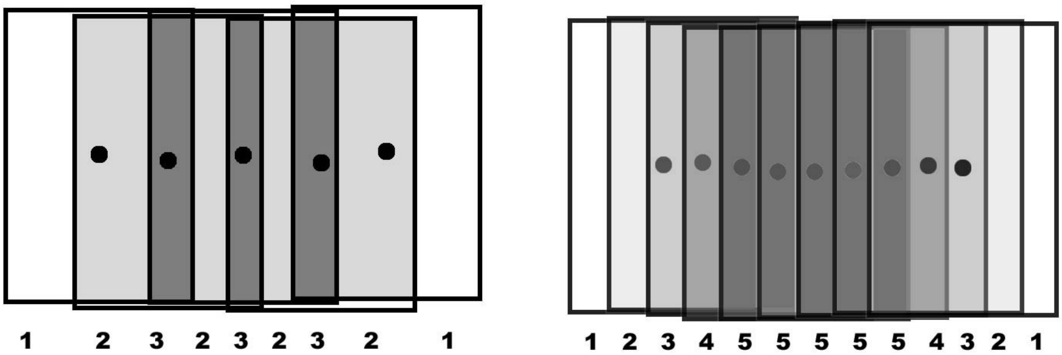


Figure 3.13 Multi-Overlay at 60% Forward Overlap (Left) and 80% Forward Overlap (Right); [171]

The transition from film to digital was completed by the replacement of film-based aerial camera systems with digital cameras since 2003, which led to a paradigm shift in the field of aerial photogrammetry [97]. Substantial benefits for aerial survey companies and data consumers resulted from this development, in terms of the image quality produced (no film grain), the capture and processing workflow (no consumables or chemistry required), and the production cost (huge potential for automation in data processing) [170]. Due to the elimination of cost for consumables, digital images could be captured at significantly larger overlap of 80% in the flight direction and 60% across (Figure 3.13). Note that each point is viewed 5 times for individual flight lines, and up to 15 times in for multiple flight lines with 60% side overlap. This redundancy enables automatic workflows such as multi-ray matching for automatic 3D modelling [171].

Numerous providers offer digital aerial camera systems for medium and large scale aerial photography. Three of the most commonly used systems are the Digital Mapping Camera (DMC) system offered by Intergraph, the UltraCam family of cameras developed by Microsoft Vexcel, and the Airborne Digital Sensor (ADS) family of cameras offered by Leica Geosystems [208]. While the former two systems utilize a combination of multiple area-CCD sensors in a grid arrangement, the latter follows a similar principle to satellite cameras by using linear sensors in a push-broom fashion to scan swaths of terrains at a time.

3.2.4 Aerial LiDAR

Aerial Light Detection and Ranging (LiDAR) systems (laser scanners) from providers such as Leica, Optech, TopSys or Riegl represent a frequently used direct sensing alternative to photogrammetry using aerial images [209]. LiDAR is an optical remote sensing technology, which may be deployed from aircraft to measure distances to the ground. LiDAR sensors actively emit laser beams at a high rate (e.g. **400,000 Hz**) [210] and measure properties of reflected light to determine range or other information of objects. Aerial LiDAR systems typically measure the time-of-flight t_{of} of the reflected laser beams as they pass twice through a medium (e.g. air) with the speed of light c. This time can be converted into a depth measurement d as per

$$d = \frac{c * t_{of}}{2} \tag{3.1}$$

By means of post-processing, a series of depth measurements can be converted into a georeferenced 3D point-cloud in a world geodetic coordinate system such as WGS84. This typically involves the combination of data from three sources: (a) a LiDAR sensor, (b) a GNSS receiver and (c) an inertial navigation system (INS). LiDAR scanning usually follows the pushbroom model by scanning a swath of depth measurements orthogonal to its trajectory [211]. Hence the position and orientation of the aircraft needs to be tracked continuously to allow precise geolocation of the 3D point cloud.

Aerial LiDAR systems frequently allow the capture of multiple reflections per emitted laser beam, in order to obtain depth values for various layers on the ground, such as tree crowns or the terrain underneath. While the former two properties may be used to determine the depth to an object or terrain surface, the latter property serves as a measure of the surface reflectivity [211].

Various techniques have emerged to convert LiDAR point clouds into continuous DSM and DTM models [212, 213, 214] similar to the data presented in Section 3.1.4, which can be used as an input for GIS or internet maps. More research has been done in automatic city-modeling based on LiDAR data in isolation or in combination with aerial imagery [215, 216, 217, 184, 218, 219].

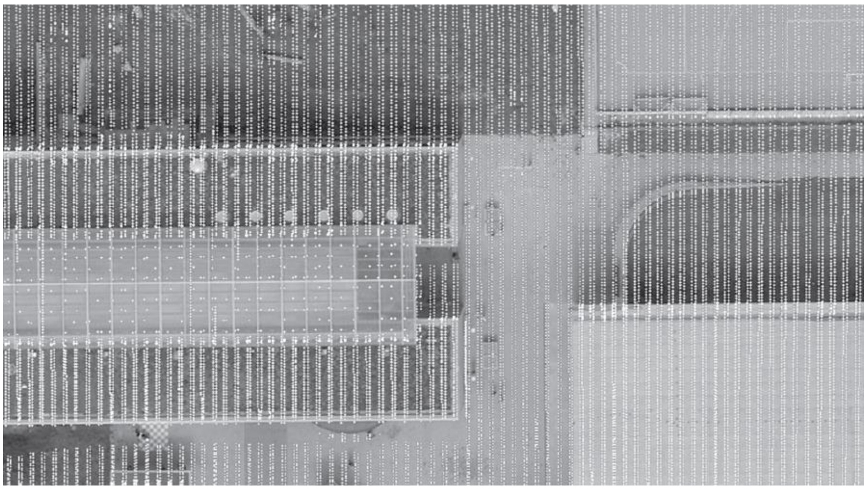


Figure 3.14 Comparison of UltraCam Aerial Photograph at 8 cm GSD with Superimposed Leica GeoSystems ALS50 LiDAR Point Cloud at 70 cm Spacing in Forward Direction and 45 cm across [170]

While LiDAR depth measurements can be relatively accurate on flat surfaces (**2-15 cm**) [211, 170], the spacing in-between individual points (e.g. **40-70 cm**), the point diameter on the ground (e.g. **1 m**) and the accuracy of the GNSS/INS orientation system are often limiting factors [211]. The difference in point spacing compared to aerial image at 8 cm GSD is visualized in Figure 3.14. Therefore using dense matching techniques on 3-30 cm GSD aerial imagery often leads to more detailed DSM and DTM products at comparable height accuracy of 2-3 cm [170].

3.2.5 Oblique Aerial Imagery

Oblique imagery has a history in military reconnaissance over restricted areas dating back to 1920 [220, 221]. Since 2000 it has resurfaced within Internet mapping. In this application it is not used for relative accuracies better than 1/10,000 but to please the eye, and for simple quantitative measurements in urban areas [93]. Due to the focus on urban areas, the data volumes are reduced compared to vertical imagery. In June 2013, Bing Maps reported its total “Bird’s Eye” image coverage to be **0.5 Petabytes** [114].

In general, oblique aerial imagery is captured in a similar manner as nadir images, by using airplane based camera systems operated by aerial surveyors. Differences exist mostly in the setup of the cameras within the aircraft, and the camera system design. Oblique images, in contrast to nadir aerial and satellite photography, are recorded by tilting the optical axis of a camera by some angle relative to the surface normal (nadir), in the range of **20** to **60 degrees** [219].

Due to this tilting angle, objects on the surface appear in a way that is often more natural for viewers to understand than in vertical imagery, as the perspectives closer resemble their daily experience (see Figure 3.15). For example, oblique imagery also shows the side walls of buildings in urban environments, which allows easier recognition and interpretation by viewers than nadir photography.

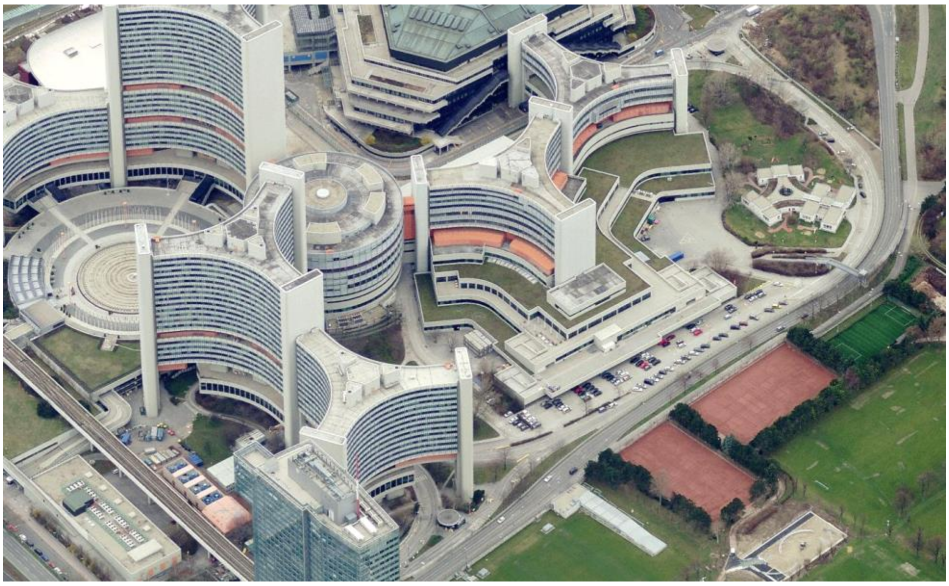


Figure 3.15 Oblique Aerial Photograph in Bing Maps @30 cm GSD (United Nations City, Vienna)

While oblique imagery could theoretically be captured by satellite sensors, the increased amount of atmospheric disturbances caused by haze or clouds at typical satellite altitudes of e.g. 600 km reduce the feasibility. Therefore, most oblique imagery is captured using aerial cameras, the optical axes of which are tilted relative to the nadir direction.

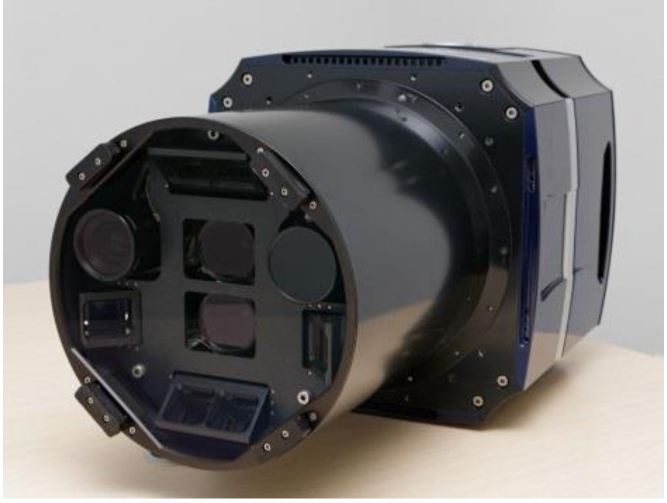


Figure 3.16 Microsoft UltraCam Osprey Oblique and Nadir Aerial Camera

An array of individual camera cones is used synchronously, to generate oblique views in multiple directions at a time. For example, an arrangement can consist of four cones, each rotated 90 degrees relative to its predecessor around the nadir axis. This leads to views of the underlying scene in all four cardinal directions. The incident angle of individual viewing rays from objects entering the camera varies throughout each image by up to half of the camera's field of view.

A common provider of oblique imagery is Pictometry [222], while other companies such as Leica Geosystems and Microsoft have also developed digital aerial camera systems for oblique image capture. A 2008 overview of oblique camera systems [221] lists more than 10 configurations containing between 1 and 8 individual cameras. A more recent (2013) camera design capturing both nadir and oblique imagery simultaneously is the UltraCam Osprey [223] shown in Figure 3.16.

Oblique imagery can be embedded in internet mapping either as a collection of individual images, or as a continuously stitched oblique mosaic. In both cases, geo-registration of the images to each other, and ortho-images is required to provide a smooth transition across layers. Accurate registration between oblique images, 3D elevation models and vector data further enables the augmentation using vector information in a similar manner as ortho-images. Vector information, if rendered correctly by considering the 3D geometry (Figure 3.1 above), can add useful information to the oblique views [224].

Additionally oblique images can be used to texture 3D city models with higher-quality façades than nadir data [218, 219], or for semantic analysis of the building details such as for counting floors and windows and creating semantic models thereof [225].

3.2.6 Aerial UAV Photography

"Unmanned aerial vehicles" (UAV) or more specifically smaller sized "micro aerial vehicles" (MAV) such as the Microdrones md 4-200 in Figure 3.17 provide a low cost, and easy to setup alternative for capturing aerial imagery of small areas such as construction sites, parks, archeological sites, areas impacted by catastrophes etc. They are increasingly used by aerial mapping firms in order to generate mapping products such as orthophotos, DSM or 3D reconstructions of specific locations within a short time frame. Apart from the faster availability and lower cost of the data, other advantages include higher resolution (smaller GSD) of the imagery collected, less dependency on weather conditions due to the low flying altitude and lower emissions [226].

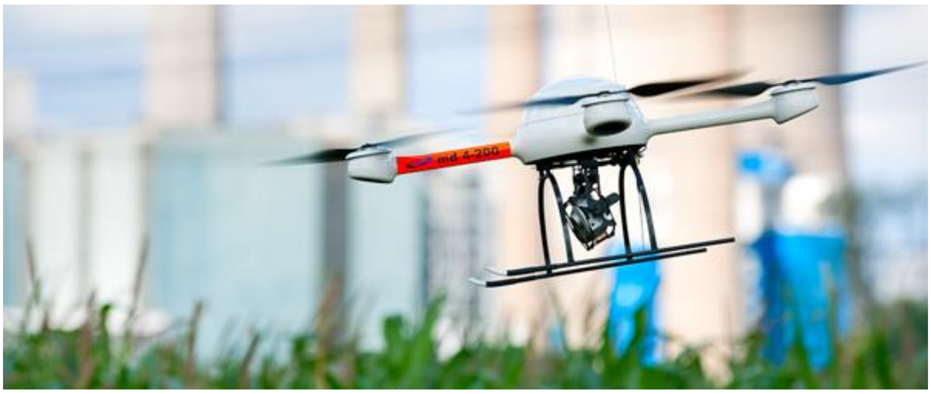


Figure 3.17 md 4-200 UAV [227]

Research by [174] led to Pix4D [228], an automated processing workflow enabling the generation of different mapping products such as the surface mesh shown in Figure 3.18 from UAV imagery. A similar workflow based on SFM and dense matching has been proposed by [175]. While many data products are used for surveys of individual sites such as construction areas at accuracies of **0.02** to **0.2 m** [229], the same mapping products can also be used for internet mapping. They pose an especially attractive alternative in case of disaster response or other scenarios requiring fast turnaround.

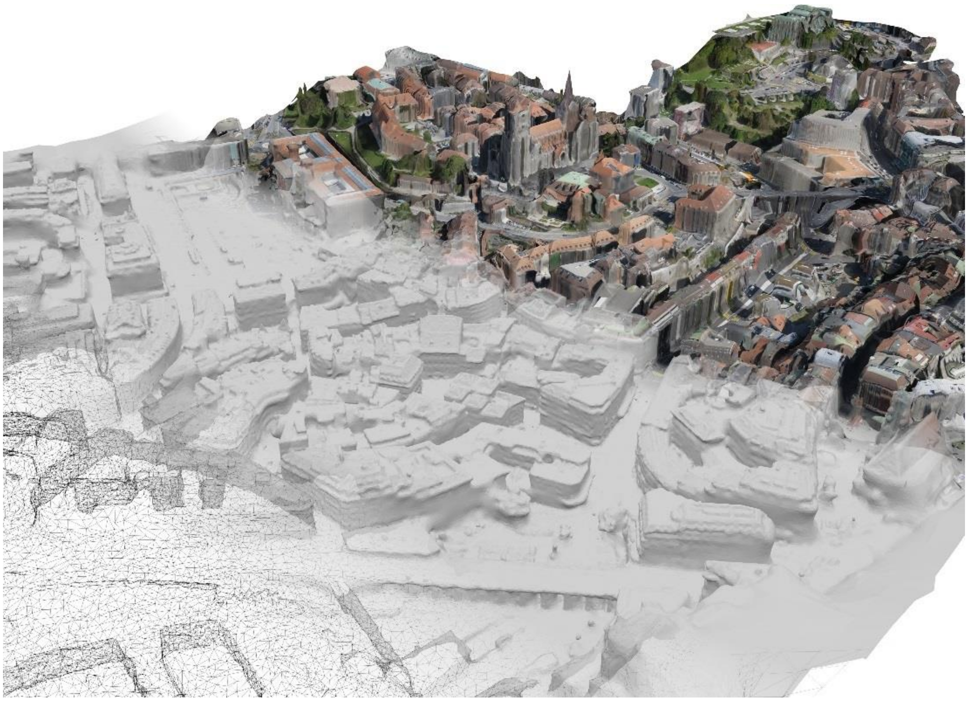


Figure 3.18 Textured Surface Model of Lausanne, Switzerland, Generated from UAV Imagery using Pix4D Workflow; [230]

Depending on whether the flight pattern is pre-defined (e.g. Figure 3.19) or arbitrary, UAV and MAV may be classified as systematic or semi-systematic capture platforms.

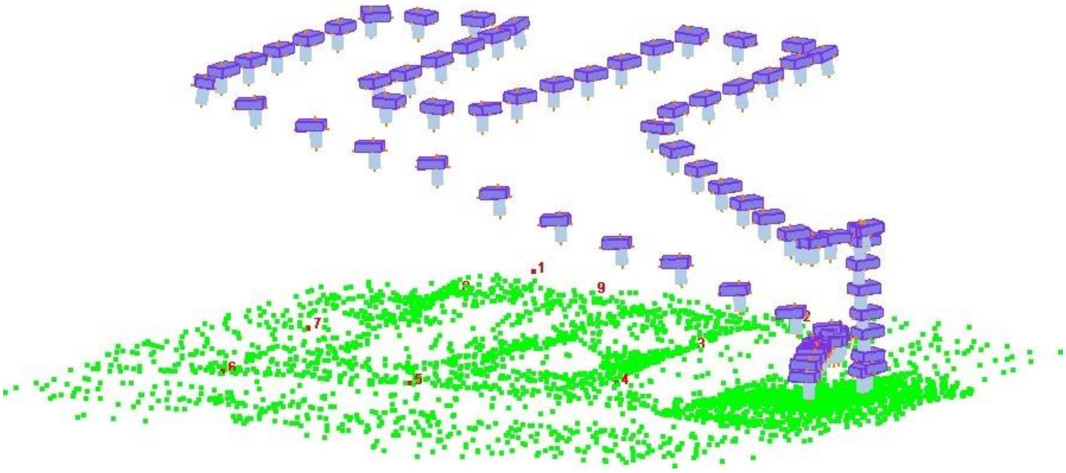


Figure 3.19 Example Flight Path of an Unmanned Aerial Vehicle [226]

3.3 Internet-Inspired Digital Aerial Camera System

We review a specific aerial camera development, as the main system driven to support a global data infrastructure for location-aware Internet search. The UltraCam was originally invented by Vexcel Imaging GmbH. in Austria for the general aerial mapping market and introduced in 2003 [97]. It very much inspired the rapid transition from aerial film to a fully digital workflow of aerial photogrammetry [171, 231, 232, 93].

However the need to add a high resolution image backdrop to emerging location-aware Internet search systems motivated a transfer of the UltraCam inventor into Microsoft's Virtual Earth, now Bing Maps, program by mid-2006 [203].

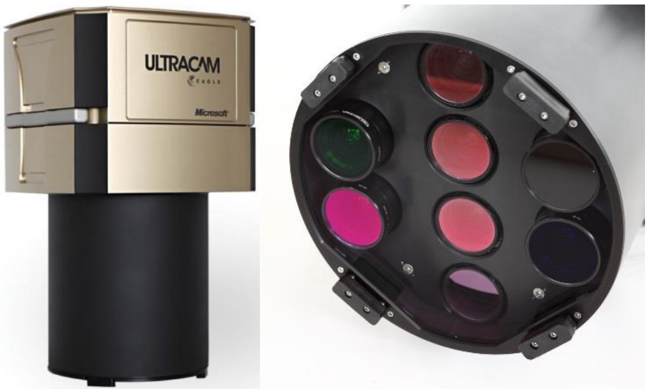


Figure 3.20 Microsoft UltraCam Eagle – Flagship 260 Megapixel Large Format Digital Aerial Camera; 4 Panchromatic Cones are Arranged Vertically in the Center - Red, Green, Blue and NIR Cones in the 4 Corners;

The initial model (UltraCam-D) soon became the leading product worldwide in this area with an estimated market share of 50% in 2012 [233]. Later, more models of the sensor family were released, such as the UltraCam Eagle shown in Figure 3.20 with increased image resolutions, different focal lengths and improved workflows. An overview of the different model revisions including the key specifications is provided in Table 3-2.

Format	UltraCam Generation	Year	Image Format	Pixel Count	Pixel Size	Focal Length	Cones	CCD Count
Large	UltraCam-D	2003	11500 · 7500 @ 12 bpp	86.2 M	9 µm	101.4 mm	4 Pan, 4 MS	13
Large	UltraCam-X	2006	14430 · 9420 @ 12 bpp	135.9 M	7.2 µm	100.0 mm	4 Pan, 4 MS	13
Large	UltraCam-Xp	2008	17310 · 11310 @ 12 bpp	195.7 M	6 µm	100.0 mm	4 Pan, 4 MS	13
Large	UltraCam-Xp Wide Angle	2009	17310 · 11310 @ 12 bpp	195.7 M	6 µm	70.0 mm	4 Pan, 4 MS	13
Medium	UltraCam-L	2009	8000 · 6000 @ 14 bpp	62.7 M	7.2 µm	70.0 mm	2 Pan, 2 MS	4
Medium-Large	UltraCam-Lp	2009	11704 · 7920 @ 14 bpp	92.7 M	6 µm	70.0 mm	2 Pan, 2 MS	4
Large	UltraCam Eagle	2011	20010 · 13080 @ 14 bpp	261.7 M	5.2 µm	70.0 mm / 100.0 mm	4 Pan, 4 MS	13
Medium-Large	UltraCam Falcon	2012	14430 · 9420 @ 14 bpp	135.9 M	7.2 µm	80.0 mm / 210.0 mm	4 Pan, 4 MS	13
Medium, Oblique	UltraCam Osprey	2013	11674 · 7514 @ 14 bpp	87.7 M	5.2 µm	51.0 mm / 25.5 mm/ 80.0 mm	1 Pan, 2 MS, 6 Obl	9

Table 3-2 Overview of UltraCam Aerial Camera Generations; Source: Microsoft

3.3.1 Smart Sensing from the Air

The UltraCam series of cameras showcased a novel sensor concept, using 4 panchromatic cameras as well as 4 cameras capturing individual multispectral channels (Red, Green, Blue and Near

Infrared). Each of the panchromatic camera cones is equipped with the same type of lens with approximately 100 mm focal length, exposing either 4, 2 or 1 CCD sensors:

- C0 (Master cone) holds 4 sensors (a..d), positioned in the corners of a 3·3 grid, providing a stable reference frame for stitching the remaining images.
- C1 holds two sensors (a,b), one in the top center, and one in the bottom center of the grid
- C2 holds two sensors (a,b), one on the left, and one on the right
- C3 holds a single sensor in the center location

The combined fields of view spanned by the different camera cones covers a full 3·3 grid when superimposed onto each other. (Figure 3.21). The combination of the different cones represents the field of view of a single camera with a sensor 3·3 times as large. The main reason for separating the different smaller sensors into different cones, was to enable the use of standard Full Frame Charge Coupled Device (CCD) sensors [234] rather than custom sensors spanning the whole image format.

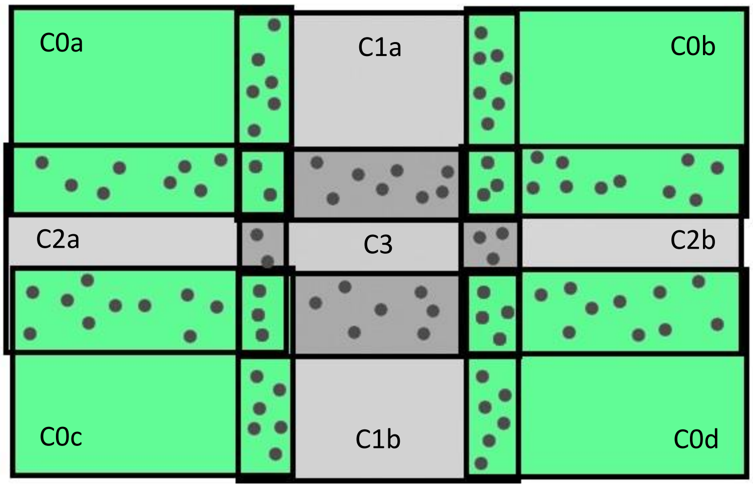


Figure 3.21 Sensor Arrangement for UltraCam Large Format Camera Systems (D, X, X-Prime, Eagle); Gray Tie Points in Overlap Areas are Used for Sub-Image Registration;

The four multispectral cones each use a single CCD sensor of the same type as the panchromatic ones combined with lenses of about a third the panchromatic focal length. They are equipped with color filters, corresponding to the desired spectral behavior for the given color channel (Red, Green, Blue, NIR). This results in a similar field of view as the panchromatic array, at a 3 times increased GSD.

In addition to the basic sensor concept, the UltraCam system featured several key advances, leading to superior image quality and more optimal workflows. Several of these advances were achieved by innovative electronics design, including a pulse pattern generator to control a full frame CCD sensor and other sub-systems described in [235].

- Full-frame CCD sensors with large 9 µm pixel led to superior image quality compared to film cameras (See Figure 3.22), with 60% reduced image noise, higher 12 bit dynamic range and better stereo-matching in low-textured areas while achieving similar sharpness as a 20 µm film scan [236].

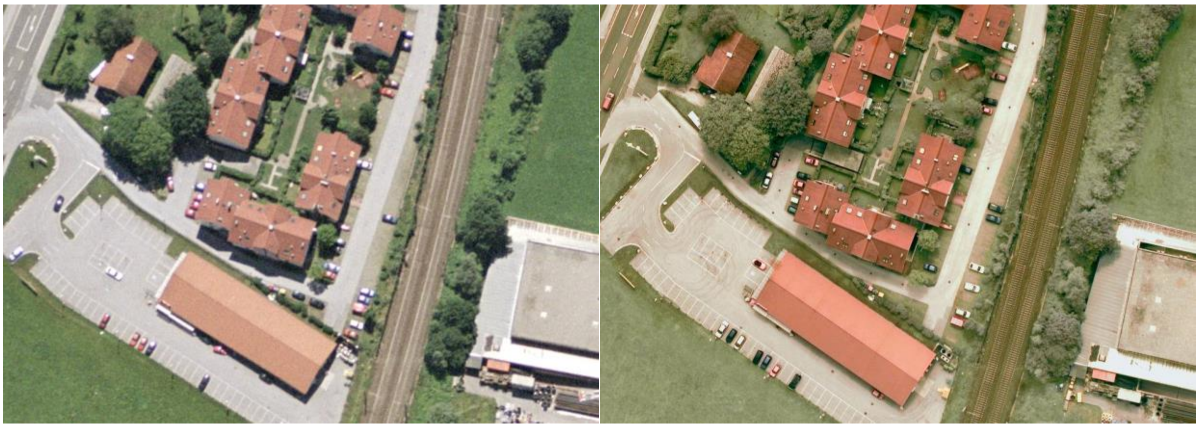


Figure 3.22 Comparison of Film Based Image (Left) and UltraCam-D Image @17 cm GSD (Right)

- Forward motion compensation has been used in analog film cameras to reduce motion-blur caused by the forward motion of the plane. While traditionally this has been achieved by mechanically moving the imaging plane synchronously with the landscape below, the UltraCam featured a novel control electronics allowing electronic FMC by means of time-delayed-integration [237]. Thus pixels are shifted across the CCD sensors synchronously during exposures, allowing increased exposure times (e.g. 10-20 times) for superior radiometry even in case of small 3-4 cm GSD under cloudy flying conditions (Figure 3.23).

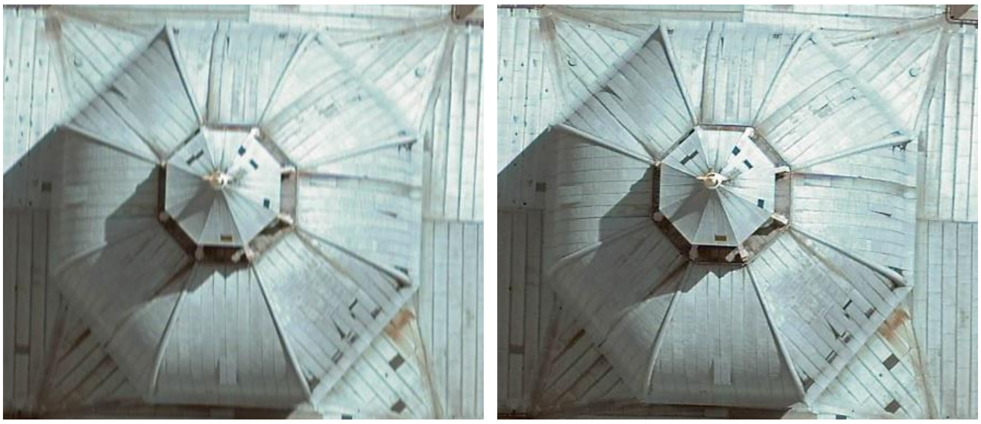


Figure 3.23 Image of the Salzburg Dome Captured @4 cm GSD w/o FMC (Left) and with FMC (Right);

- Variable aperture lenses further helped to accommodate varying lighting conditions (e.g. bright sunlight vs. twilight or cloudy) while using short (<10 ms) exposure times.
- Syntopic triggering was introduced as an electronically controlled measure to avoid parallaxes between individual sub-images due to geometric offsets in the camera body. This means images are triggered in the same location of their trajectory rather than at the same time (synchronous) [237].
- High frame rates of up to **1.3** frames per second (fps) allowed capturing imagery at significantly larger forward overlaps than for film based cameras (e.g. 80-90% instead of typically 20-30%). The added redundancy in observations of the same geographic region enabled substantially different ways of data processing, such as multi ray stereo matching, and automatic DTM / DEM generation [171, 231].

3.3.2 Processing Workflow

Various processing steps are required to obtain image products from raw Level-0 imagery, based on geometric and radiometric calibrations (intrinsics and extrinsics) obtained during a lab calibration explained in Section 3.3.3.

After the radiometric correction of the 13 raw images, to correct for defective pixels, dark level and vignetting, into Level-1, they get geometrically transformed into the respective location in the overall image format. Based on the geometric calibration, and tie-point matches in the sensor overlap areas (see Figure 3.21), bicubic resampling [238] is used to combine the individual Level-1 sensor images into a single large panchromatic and 4 multispectral image layers (Level-2). For this purpose, cone C0 (=master cone) is used as a reference to fit the in-between images. Once the panchromatic sensors have been geometrically transformed and blended into a single output image, the four multispectral images are also geometrically corrected, registered to each other, and combined into a 4-channel multispectral image.

To avoid distortions in the output images due to temperature drifts, an improved stitching algorithm was proposed by Ladstaetter et al [239], taking into account tie-point correspondences between panchromatic and multispectral images. Using a temperature model, temperature drift is estimated and corrected, leading to reduced reprojection errors during aero-triangulation.

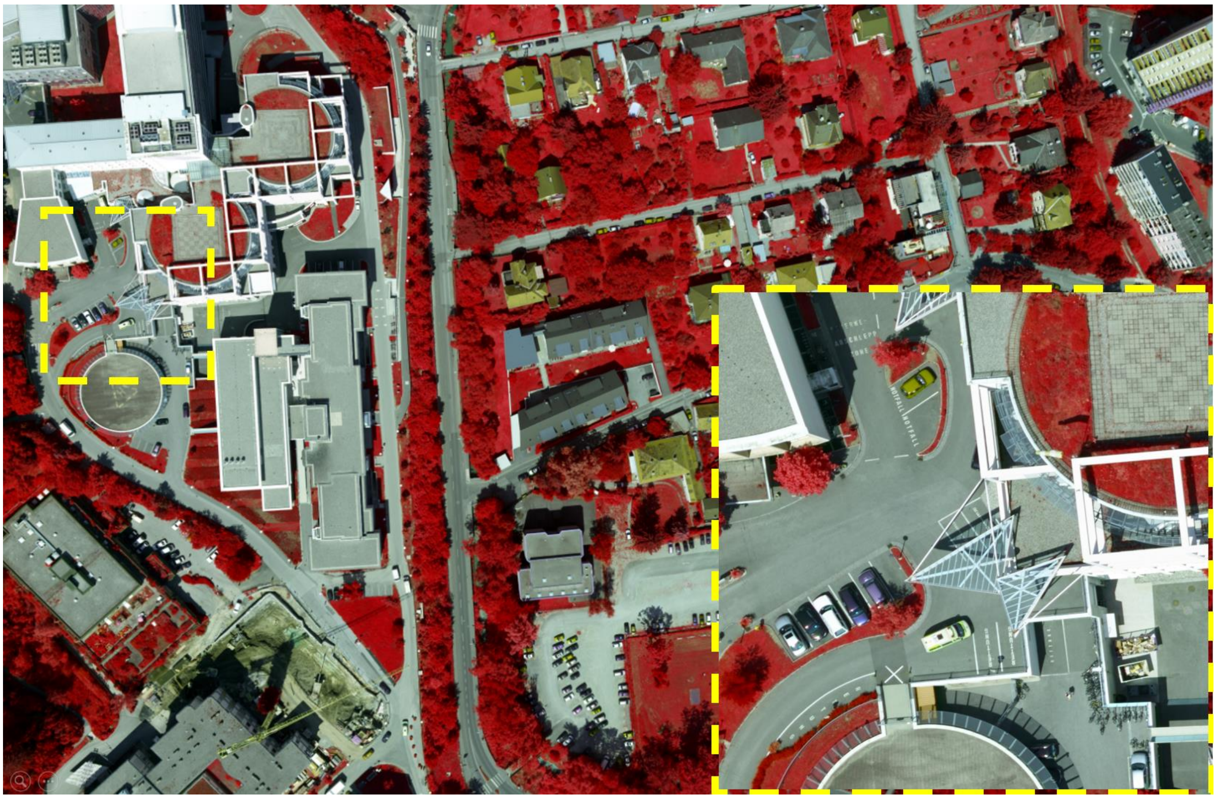


Figure 3.24 Level-3 False Color Infrared Image Captured with UltraCam-D Camera at 4 cm GSD; Red Colors Indicate Vegetation; Location: Children's Hospital in Graz, Austria;

The radiometrically and geometrically corrected panchromatic and multispectral Level-2 images are equivalent to a developed film image, in that the process for obtaining them depends purely on the camera calibration and the scene content.

Based on this intermediate format, various kinds of output image formats (Level-3) can be generated, including radiometrically adjusted high resolution RGB or false-color infrared (CIR) [97] images requiring Pan-Sharpening [191], or high resolution panchromatic images. Further radiometric adjustments such as a gamma-correction or general gradation curves can be applied during this process. While RGB images are commonly used for computing building-textures and orthophotos such as for internet mapping, CIR images (Figure 3.24) may be used to visualize (note the distinct red color) and classify vegetation like trees or bushes [240]. They are generated by mapping the near-infrared, red, and green spectral bands onto the visible RGB bands [241], thus indirectly extending the human's spectral range to the near infrared band.

3.3.3 Achieving High Accuracy Calibration at +/- 1 µm

Calibrating systematic errors of aerial cameras is essential for achieving satisfactory sub-pixel measurement accuracy in photogrammetric applications. In [242] we proposed a method for calibrating the sensor geometry (intrinsic and extrinsic parameters) of the UltraCam system, by means of bundle adjustment of a set of automatically detected 2D observations of a known 3D arrangement of circular markers.

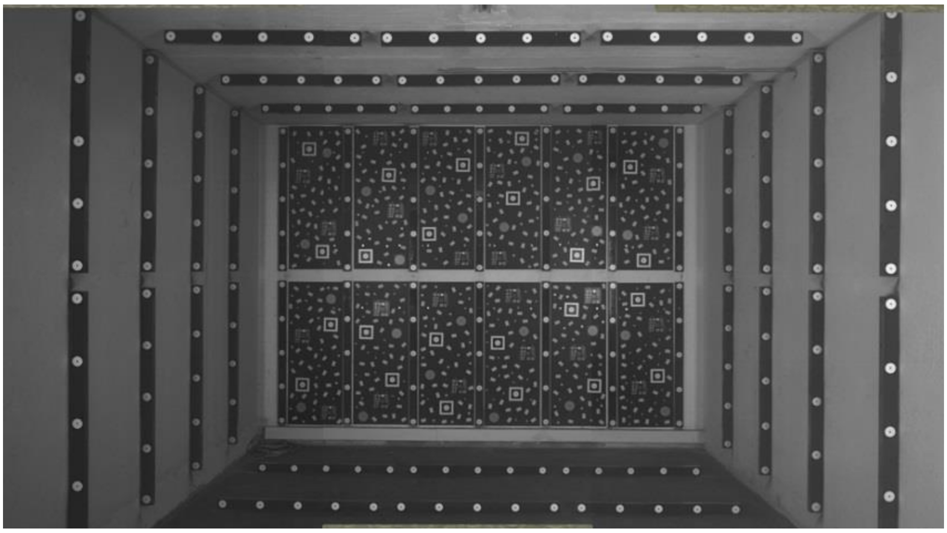


Figure 3.25 Fixed 3D Arrangement of Markers used for Geometric Calibration

For this purpose we use a commercial bundle adjustment tool called Bingo [243]. A sample 3D marker arrangement is depicted in Figure 3.25. The observed reprojection errors of individual cameras are generally better than **+/- 1 µm RMS** in the images. An example plot showing the residual calibration errors of +/- 0.8 µm RMS for a particular camera is given in Figure 3.26.

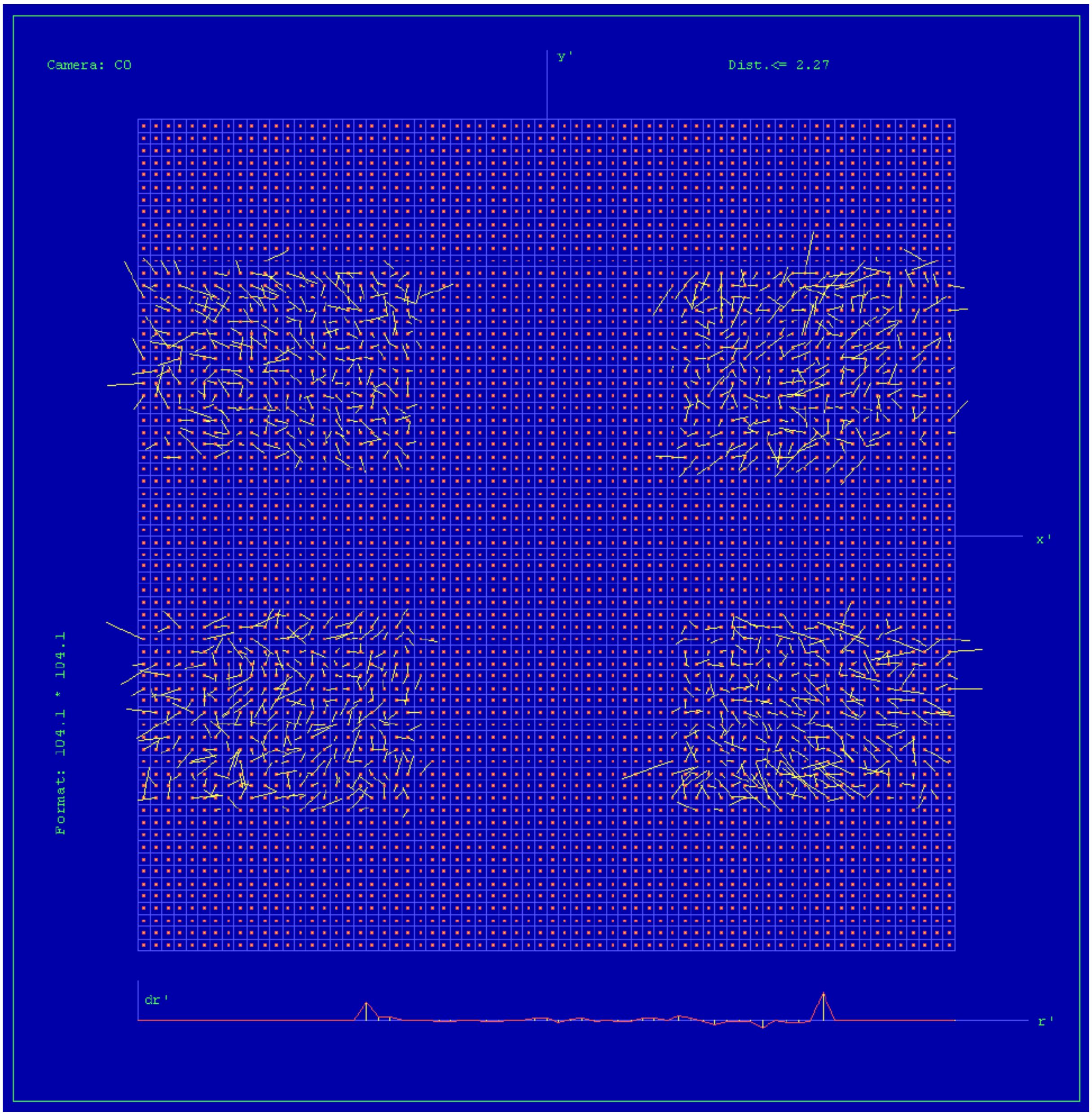


Figure 3.26 Residual Calibration Errors in Image Coordinates for 4 Sensors in Cone C0 of a Particular Camera as Reported by BINGO [243], Resulting in an RMS Error of 0.8 µm RMS [244]

In addition to the sensor geometry, the radiometric properties such as camera vignetting and CCD sensor blemishes (dead pixels) have to be determined, to allow for correction during postprocessing. For this purpose, we capture a set of **7·4** calibration images using a Teflon based diffusor disk illuminated by a set of calibrated light sources to serve as a flat field target, as illustrated in Figure 3.27.

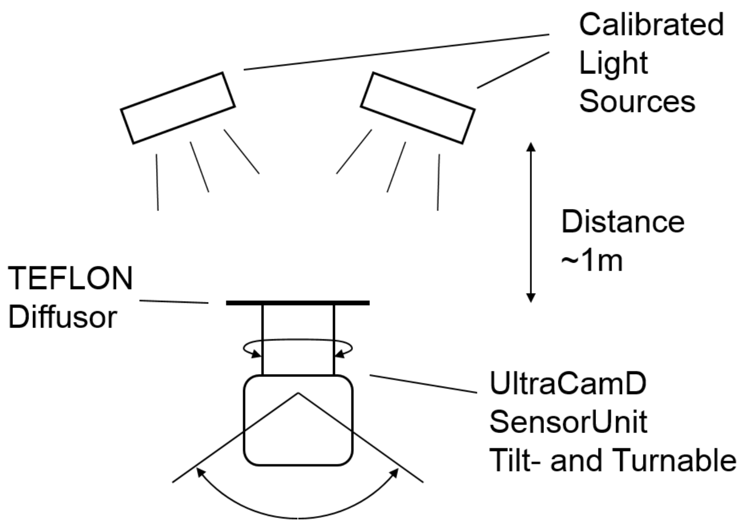


Figure 3.27 Setup for Capturing Radiometric Calibration Images

The average of all calibration images is used as a reference for computing the calibration factors for each pixel (as the inverse of the normalized intensity values), as well as to automatically detect individual pixel and column defects in the images. We rotate the camera 4 times around its axis by 90 degrees, and tilt it in 7 different orientations in 10 degree steps relative to the two light sources to achieve a symmetric illumination pattern. Since the vignetting present in a camera depends on the aperture setting used, the process has to be repeated for each of 5 supported F-numbers (F5.6, F8, F11, F16 and F22). The resulting radially symmetric correction factors for a sample camera are visualized in Figure 3.28.

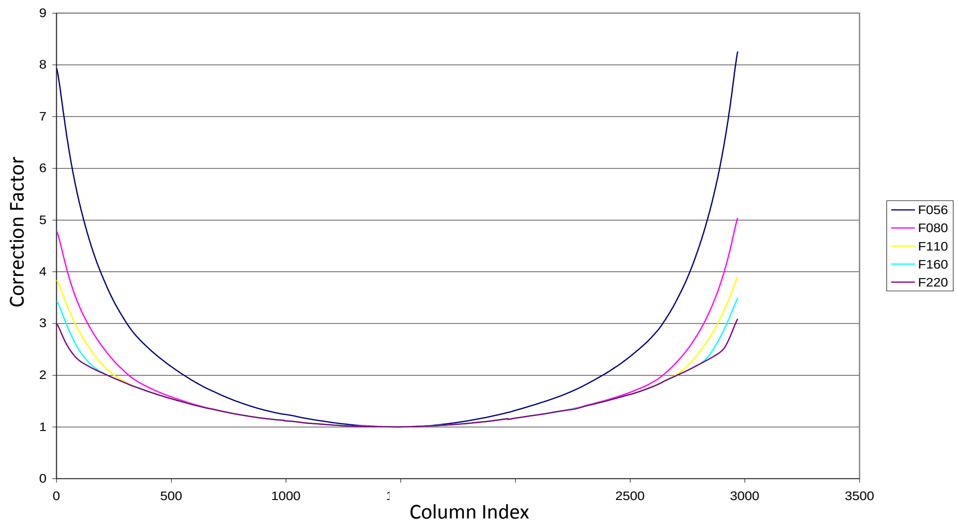


Figure 3.28 Radially Symmetric Vignetting Correction Factors for Different Aperture Settings

As part of the standard calibration procedure, a report of the relevant geometric, radiometric, optical and electronics calibrations is generated [244].

3.4 Human Scale Images and Mobile LiDAR Systems

A novel and distinct kind of systematically collected image data, frequently used for online mapping application, are terrestrial images taken from a "human" perspective, such as on streets, in pedestrian outdoor areas or inside buildings [116, 115].

The main application of human scale imagery is the Internet. Typically, images are captured in a panoramic configuration, giving a viewer a navigable 360 degree surround view of a scene. Advantages of human scale panoramas are the increased level of detail, the more natural appearance, and intuitive navigation within and in-between locations. Prior to panoramic images, mapping sites frequently used manually captured business storefront images showing the appearance of individual businesses to support navigation.

Similar to aerial imagery, specifically designed hardware is required for capturing panoramic images in street networks spanning many thousand kilometers. The lack of existing commercial products led companies like Microsoft or Google to develop their own streetside capture systems.

Human scale data have different image scales compared to aerial or satellite images. While satellite images for internet mapping are typically captured at **50 cm** GSD or higher, and aerial imagery at **15 cm** GSD or higher, the GSD of human scale images may be as little as **2 cm** for outdoors and **0.5 cm** for close-by indoor objects. Similarly, the capture intervals are substantially smaller (several meters versus tens or hundreds of meters for aerial), based on the desired image density at a specific location.

As of now, the concerted capture of human scale imagery has been focused largely on public streets or areas accessible with a trolley, cart, bicycle, or similar form of transportation, while examples of other environments such as indoor venues or private outdoor areas accessible by walking have been shown less frequently.

Leberl [1] has estimated that the total data volume required for human scale data to support a 3D world model at 2 cm urban street GSD and 0.5 cm indoor GSD is approximately **1,500 Petabytes** in addition to **190 Petabytes** for 15 cm aerial coverage. Actually reported data volumes still lag behind this estimate. Google reported in 2012 to have altogether 20 Petabytes of (compressed) street view images released, covering **8 million kilometers**, which corresponds to roughly 300 Petabytes of raw data captured [8].

In the following, we describe the specific aspects of terrestrial imagery used for mapping services, including streetside images, storefront images and images captured using wearable systems. Additionally we provide an overview of different formats used for storing panoramic images. Further requirements specific for human scale data capture are provided in Section 4.3.

3.4.1 Streetside Imaging

The term "streetside images" (or "Street View" in case of Google) usually refers to human scale images captured on public street networks, either in urban or rural settings [245]. As such, streetside images capture objects such as building façades, vehicles, street signs, people, animals, vegetation and various other objects also referred to as “street furniture”. Such data gets captured using custom mobile mapping systems (MMS) carried by vehicles such as vans or cars, operated by vendors such as Facet Technologies [246]. Several such sensor platforms have been built by different companies such as NavTeq, TeleAtlas, Google, Microsoft, as well as numerous universities and research organizations.

Streetside imagery provide views of a location (urban or rural) similar to what a user would perceive on-site, driving a vehicle or walking on the respective street. For example this can be used for navigation, to indicate paths to locations such as points of interest or addresses. It also allows exploration of remote locations, such as when planning a trip or deciding which hotel or restaurant to choose.

Similar to aerial imagery streetside capture follows previously specified capture pattern, e.g. by driving each street of a given city, and taking pictures at a constant interval. Therefore, as pointed out above (3.1.3) they provide largely homogeneous coverage throughout a region (e.g. a city), independent of the significance or relevance of a place within the region (see Figure 3.29).

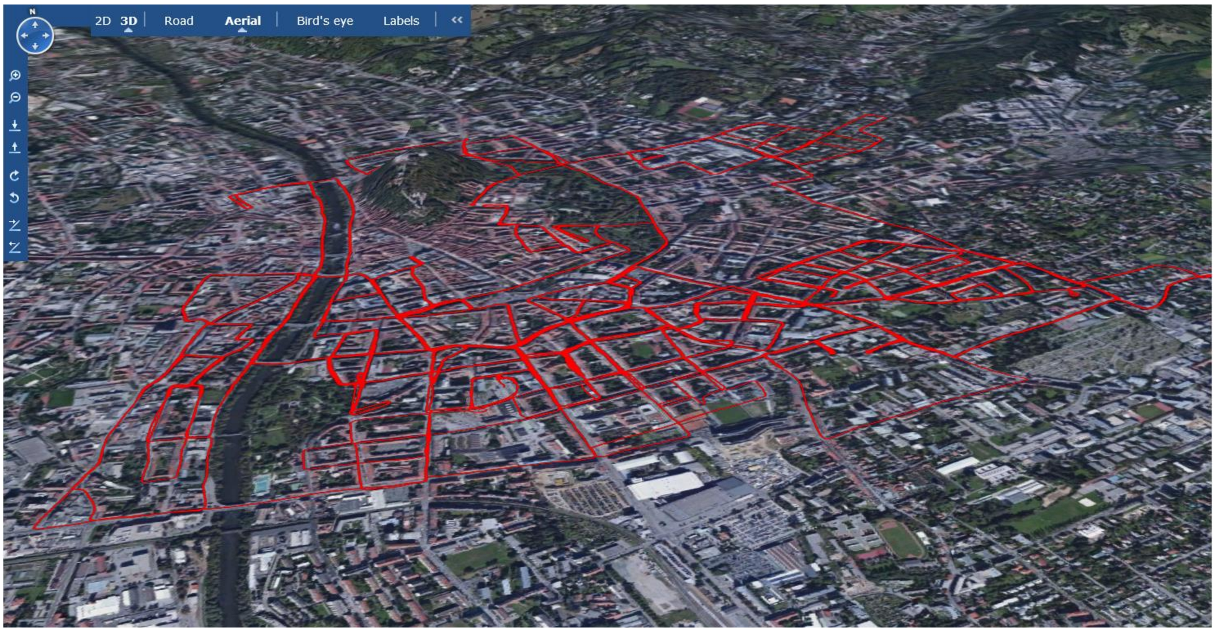
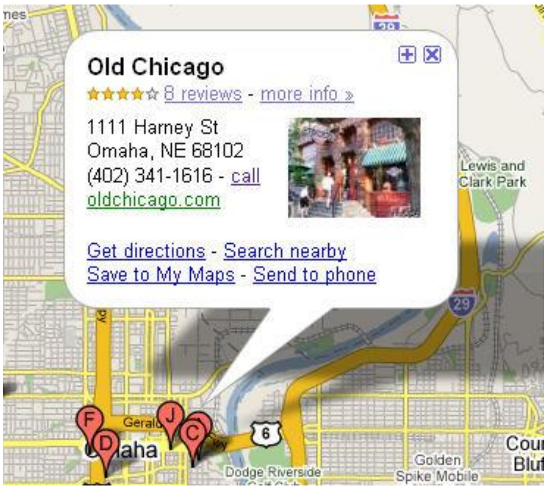


Figure 3.29 Typical Coverage Pattern for Streetside Capture in Graz, Austria with UltraCam-M

3.4.2 Business Storefront Imagery

A specific type of geospatial imagery covers business storefronts, which are associated with points of interest (POI) on the map (Figure 6.15). Storefront imagery have been used by mapping sites prior to the availability of panoramic streetside data to show the appearance of a business from outside. This data type is partially still used today in areas without streetside coverage.

The acquisition is often manual via sub-contracting companies such as InfoUSA [247], which hire a group of people to perform the field capture of images and rough GPS locations for POIs in a defined area. Although the geographic distribution of this type if imagery is not systematic (e.g. in equidistant intervals), we still consider the acquisition process a systematic one. Typically the extent of the region is precisely defined, such as by using street numbers or city blocks, and the task is defined by provided instructions and a list of businesses to be captured.



Old Chicago

★★★★☆ 8 reviews - more info »

1111 Harney St
Omaha, NE 68102
(402) 341-1616 - call
oldchicago.com

Get directions - Search nearby
Save to My Maps - Send to phone

Figure 3.30 POI Listings on Google Maps and Bing Maps were Accompanied by Storefront Images Prior to the Emergence of Street View Data [59]

3.4.3 Mobile LiDAR

Similar to aerial data capture, LiDAR scanning may also be used in terrestrial settings to directly obtain depth information of a scene such as of building facades and other urban objects. Providers such as Faro, Riegl, SICK, MDL, Trimble, and Leica offer various types of LiDAR systems aimed at stationary and mobile data capture as part of a mobile mapping system (MMS) [248, 249, 250].

Depth information is obtained by measuring the time of flight, or phase shift of the reflected laser signal. The laser emitters and receivers often rotate at very high rates, thus scanning a slice of the observed scene during each revolution at scan rates of up to **1 MHz** [251].

Georeferencing of the 3D point clouds captured by MMS frequently involves the same kind of components as for aerial systems such as GNSS and INS [252]. While LMS increase the cost and complexity of MMS compared to cameras, they directly obtain depth without the need for postprocessing.

We describe the use of a mobile LiDAR system below in Section 4.3.4 as part of a particular mobile mapping system design.

3.4.4 Indoor and Unnavigable Areas

Several companies such as Google and Microsoft have been working on solutions for capturing human scale imagery in areas inaccessible by streetside capture systems. Such areas include outdoor venues such as parks, golf courses, skiing slopes and hiking trails, or indoor venues like sports stadiums, shopping malls, restaurants, museums and real estate. Same as for streetside data, panoramic images allow an immersive exploration of such locations.

Imagery can be obtained systematically or via crowd-sourcing. Systematic data collection is often performed using custom capture hardware, by internet mapping firms or by third party sources such as 360 Cities [253]. Mobile versions of Google's capture platform mounted on bicycles, carts or skidoos as shown in Figure 3.31, have been used to capture otherwise inaccessible areas. Crowdsourced data capture uses panorama capture software such as Photosynth Mobile [254].

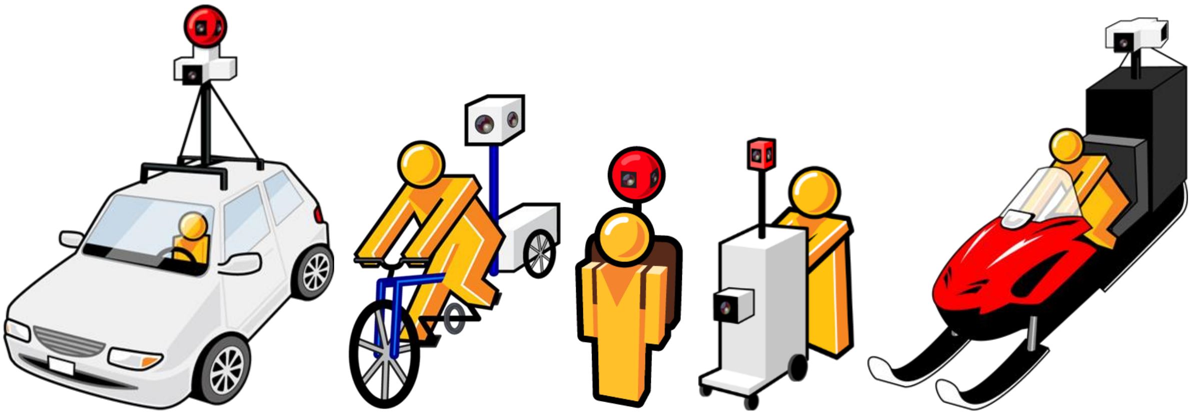


Figure 3.31 Different Google Street View Capture Platforms; Source: [115]

While concerted capturing of such venues at a global scale by far lags behind the scale of streetside capture, the potential data volumes are expected to outgrow currently available images. This results from the large number of locations accessible for pedestrians, and the higher density of images required for such areas.

3.4.5 Panoramic Image Representations

Panoramic images are a popular form of visualizing a specific location from a human perspective, which allows immersive exploration of environments such as street scenes or indoor venues. Viewers can freely select the view port by zooming in and out, rotating around the vertical axis (pitch), as well as up or down (yaw angle). Apart from streetside and indoor captures by mapping providers, panorama stitching tools such as Photosynth Mobile [254] or Microsoft ICE [255] produce similar 360 degree panorama data usable for crowdsourcing.

Panoramas can be rendered for viewing in different fashions, such as via the transformation into a single 2D image using cylindrical or equirectangular (spherical) projection [256]. This format can also be shown in relatively simple client applications (e.g. HTML 4.0) which do not support arbitrary views, while navigation is limited to panning and zooming. Further the views in such panoramas are distorted compared to central perspective images. Figure 3.32 (center and bottom) shows examples of panoramic images under cylindrical and equirectangular projections.

Alternatively, in case the client application supports more complex transformations to the input panoramic image (e.g. homography transformation using a 3D rendering engine such as OpenGL or DirectX), an arbitrary and perspectively correct view corresponding to a virtual camera can be produced as the user scrolls and pans through a panorama [256]. For this purpose, a virtual camera viewport with a specific angle of view, viewing direction and focal length can be placed at the center of the panorama cube, and a viewport specific image can be rendered.

A common representation for storing human scale panoramic images represents the panoramas (see Figure 3.32 - top) uses 6 cube face images of a sub-segment of the overall solid angle. Each cube face further can be stored at different levels of detail in a tiled quadtree structure [137], such that only the relevant viewport and zoom level needs to be downloaded to the client application for visualization. The tiles used to store the panoramic images typically have a constant pixel count (e.g. 256 · 256 pixel) [257].

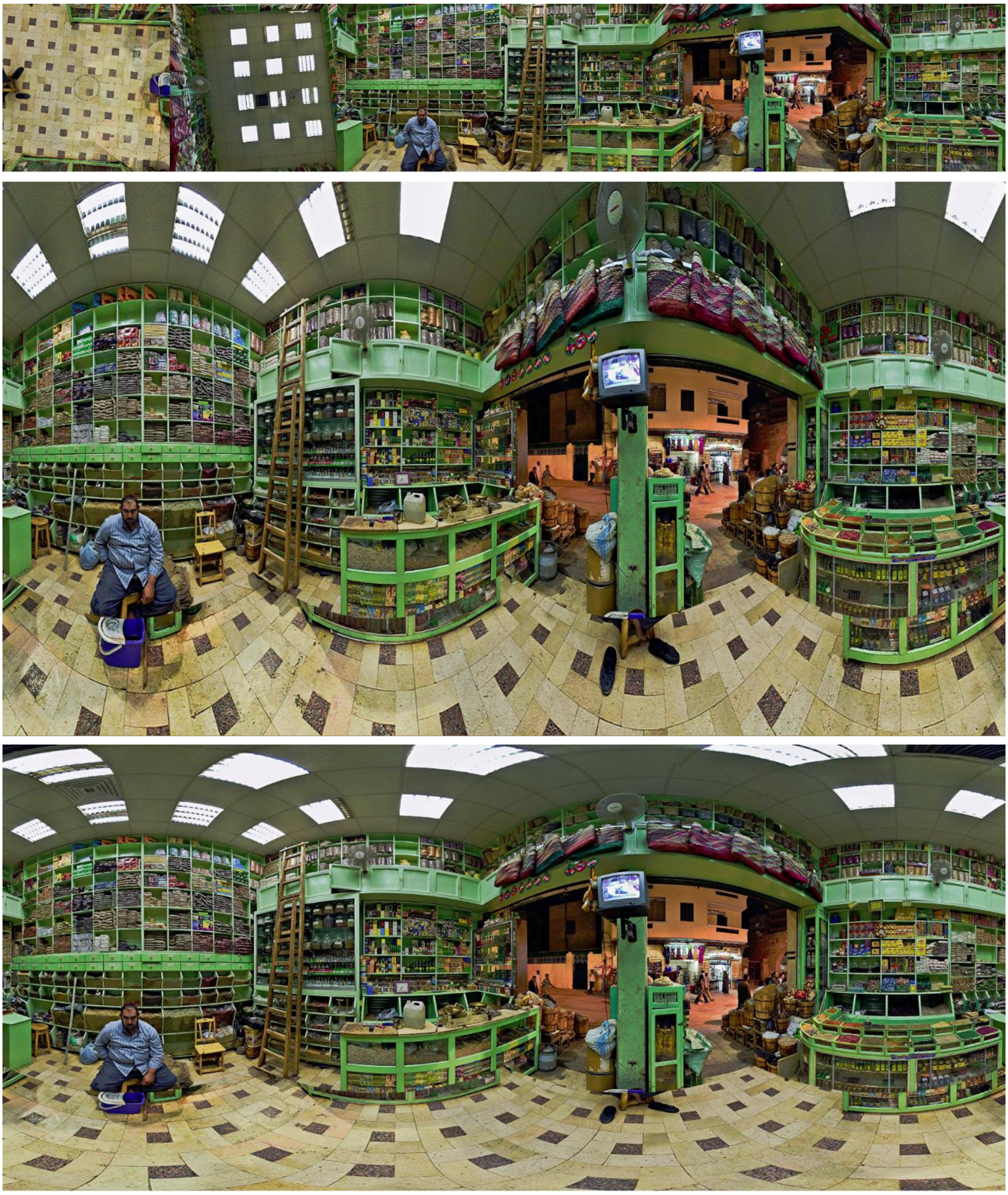


Figure 3.32 Different Formats for Panoramic Images of a Spice Shop in Aswan, Egypt: Cubic Format (Top); Cylindrical Projection (Center); Equirectangular (Spherical) Projection (Bottom); [258]

3.5 Crowdsourced Images

Web based mapping services currently rely on systematically collected images as illustrated above, yet there exist examples of crowdsourced images as integral parts of online maps. The term “crowdsourced” refers to the fact that a large group of amateurs capture imagers, upload these to some web based community portal and share them with a broader community [259].

3.5.1 Digital Visual Memories

Digital cameras and online sharing have created an abundance of collective ‘digital memories’. Pocket point-and-shoot cameras, digital SLRs, camcorders, surveillance cameras and smartphones can quickly and easily document events. The circumstances as well as motivations for taking photographs can be numerous, either for personal or for commercial applications.

Some examples of personal uses of a camera are: documenting important moments in life, recording places visited while traveling, or simply to capture the aesthetics of a scene. People do this either to enhance their own memory, share their experiences with others, create art, or simply because it is virtually cost-free to take photos even without any obvious reason [260]. Professional uses of digital cameras include news reporters, forensic evidence, real estate, surveillance cameras installed for public safety purposes, traffic and weather cameras.

New methods of sharing digital photographs have emerged since 1992 with the introduction of Photo CDs [261] and DVDs, high resolution mobile phones and digital photo frames. The web has also provided plenty of online photo sharing and social interaction websites such as Flickr, Panoramio, Instagram, Photobucket, Facebook or Twitter. These services, also referred to as Community Photo Collections (CPC) [2] host a quickly growing collection as detailed in 3.5.2.

Besides people sharing images with their existing friends, new communities have been formed due to common interests in photography, creating artistic images of places, etc. For example, Flickr “meet-ups” and “photo walks” happen regularly in many different locations around the world [9]. People make it a hobby to create artistic photography by using tools like Photoshop or Instagram, sharing them with their community and commenting and voting on the aesthetic nature of their work. In addition to real photography, even fake imagery often gets created for the same reasons.

Though pictures may not be captured with the intention of feeding data to internet mapping sites, they frequently contain geospatially relevant contents such as buildings or monuments. Agarwal et al. [11] have shown that a model of a significant part of a city like Rome could be created from 150,000 crowdsourced images on Fickr.

3.5.2 Image Sources

Being traditional photo and video sharing sites, Picasa, Photobucket and Flickr were created in 2002, 2003 and 2004 respectively, primarily as Web 2.0 platforms for people to share their images online with other people, be they friends and family [262]. These services allow uploading previously collected imagery captured with various kinds of cameras like SLR, point-and-shoot and mobile phone cameras either directly on their web-sites or using client applications. More recently, social networking sites such as Facebook or Twitter added photo sharing functionality, which gained significant popularity in addition to general social interaction.

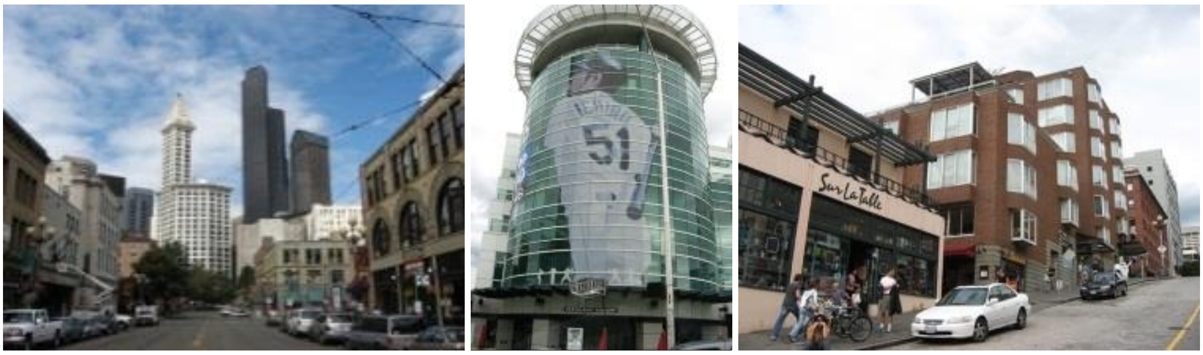


Figure 3.33 Sample Set of Geo-Tagged Outdoor User Photographs from Flickr

The ubiquity of **1.2 billion** [126] smartphones and facilities provided for capturing and sharing pictures, led to a vast trend towards mobile photo sharing with **300 million** photos uploaded only to Facebook per day [18]. Therefore, many of the traditional photo sharing sites and social networks released mobile photo sharing apps or added related functionality to their existing applications. Instagram, which was launched in October 201, is another example of a mobile photo sharing service, primarily as a mobile phone application for the iPhone, allowing application of digital image filters, and sharing of the photos with other Instagram users.

While Flickr added a geo-tagging feature in 2006 [263], certain image sharing services directly associate images with geographic locations or points of interest by means of geo-tagging. Panoramio which was started in 2005 and later integrated into Google Maps, is a crowd-sourcing service specifically aimed at capturing outdoor locations of interest to people, which can be tagged using words describing the image content or location. Google Maps contains functionality for exploring such user photographs in addition to aerial and street view imagery. A set of sample outdoor images which were geo-tagged on Flickr is shown in Figure 3.33.

The integration of images with maps continued with the release of Photosynth, which is based on the work by Snavely et al. [15], and automatically creates a 3D reconstruction of a scene by using a Structure from Motion (SFM) algorithm [24]. “Structure from motion aims to recover camera parameters, pose estimates, and sparse 3D scene geometry from image sequences” [15] based on a collection of photographs taken from different perspectives. An example Photosynth view of a scene as well as the corresponding 3D point cloud is visualized in Figure 3.34. Functionality for georeferencing such “Synths” by aligning the point cloud derived from the 3D reconstruction, to natural features in an aerial view [264] was added later on. This enhances the viewing experience for transitions between the different views [265]. Additionally an option of exploring the Photosynth collection through a map interface improved the discoverability of such data. Later Google added a similar feature called “Photo Tours” [266] which allowed exploration of user-contributed images integrated with in a similar way as Photosynth. However this feature is currently no longer available.

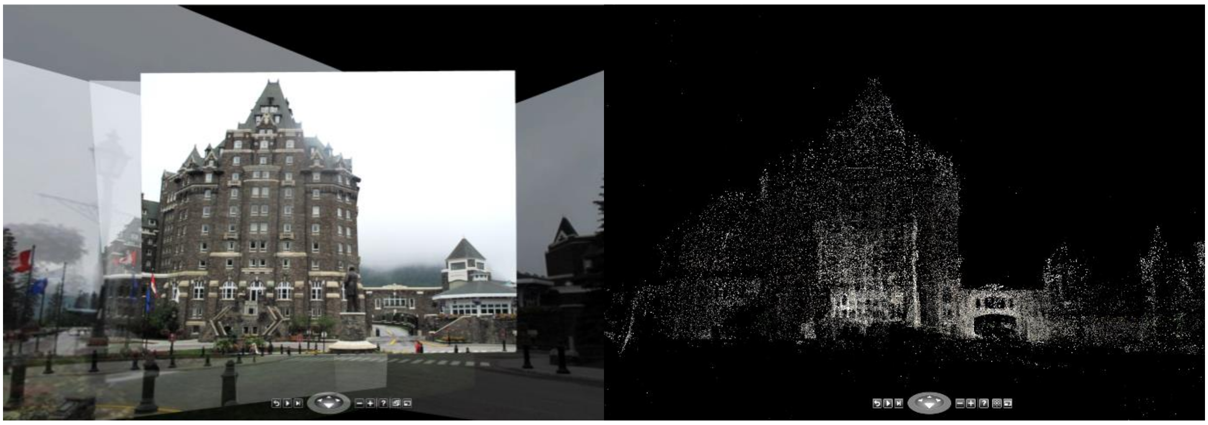


Figure 3.34 Photosynth Generated from Crowdsourced Imagery (Left); Corresponding 3D Point Cloud (Right); Example: Banff Springs Hotel, Banff, Canada

While most shared image content consists of individual photographs, taken with central perspective point and shoot cameras, recent developments of mobile panorama capture applications allow stitching panoramic images directly on mobile devices, and sharing them on the internet. Such services include AutoStitch, 360 Panorama or Photosynth Mobile. The latter enables users to capture images very similar to human scale data shown on Bing Maps, and share them with the community [254]. Desktop tools such as Microsoft ICE [255] also create panoramas from multiple images captured in a panoramic fashion, which can be uploaded to community sites.

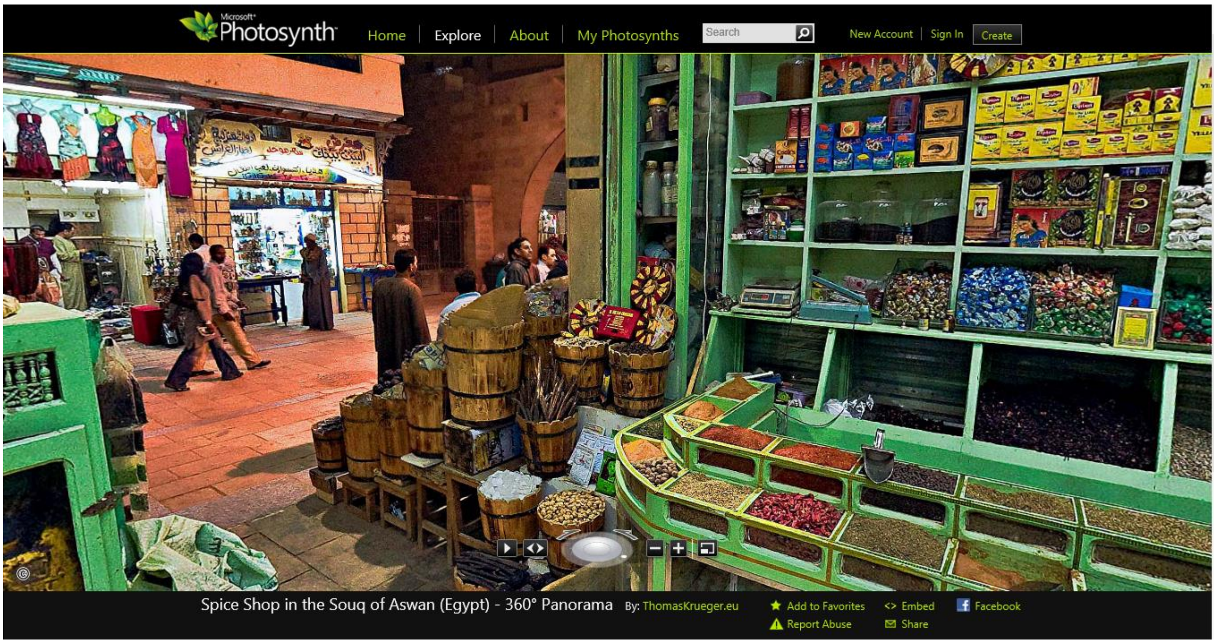


Figure 3.35 Sample Photosynth Mobile Panorama Image in a Spice Shop in Aswan, Egypt (See also Figure 3.32); [258]

Additionally, many more web services with different primary foci exist, allowing community photo-sharing, such as blog sites, Wikipedia, RedFin, Amazon, EBay, SnapChat or Tumblr.

3.5.3 Data Volumes

Established photo sharing sites such as Photobucket and Flickr last reported to have **10** and **6 billion** photos in total [267, 10]. By September 2012, after only two years, Instagram had grown to a base of 100 million users uploading 5 million new images per day. Additionally, Instagram was acquired by Facebook in 2012 for 1 billion USD, which is an indication of the general trend and value of such services [268, 269].

However, these numbers are dwarfed by an asset of **220 billion** photos reported by Facebook alone at the end of 2012, occupying roughly **150 Petabytes** [18]. According to the report, Facebook users upload **300 million** new photos **per day** - 200 times more than Flickr's 1.5 million [270]. At this rate Facebook will add the combined total asset of dedicated photo sharing sites every 70 days. Given that Facebook has roughly 20 times as many active members as Flickr, this indicates that Facebook users are on average about 10 times more active in sharing photos [271].

An overview of the different data asset statistics is given in Table 3-3. No photo upload statistics could be found for Twitter, Photosynth and Panoramio, but we assume these services have significantly smaller data collections due to the short time of availability in case of Twitter and the focus on locations in case of the latter two. The total user count of 500 million indicates a large potential for Twitter to keep up with Facebook's numbers [272].

Provider	Last Reported	Total Photos	Uploads / Month	Total Users
Flickr	8/2011	6 billion	45 million	51 million
Photobucket	12/2012	10 billion	120 million	100 million
Instagram	10/2012	5 billion	150 million	100 million
Facebook	10/2012	220 billion	9000 million	1000 million
Twitter	3/2012	n/r	n/r	500 million

Table 3-3 Data Volume Statistics for Major Photo Sharing Services

Based on an analysis of 11,000 Flickr images in 2009, **23%** were found to be geographically relevant. Assuming that this ratio is approximately constant for other sources like Facebook, this means that about **40 Petabytes** of geospatially relevant imagery exists in CPC today.

Compared to the estimated **1,700 Petabytes** of image data required for a 3D world model [1] this still represents a small fraction (**2.3%**), especially considering the typically uneven geographic distribution of CPC data.

3.5.4 Geolocation

Lately there has been a growing demand for photos associated with geographic locations in a process called "Geo-Tagging". It is a useful way of organizing the information, either for personal use ("find all the photos from the vacation to Hawaii") and commercial use ("What does that neighborhood look like?"). A common approach is to use Global Navigation Satellite Systems (GNSS) such as GPS (USA) or Galileo (Europe) devices to capture the location (latitude and longitude) continuously using satellites or an A-GPS which also uses the cellular network or Wi-Fi hotspots. This information can be stored along with the image data (such as in the EXIF headers of the digital file) and can then be inserted into a spatial index for fast search.

The advantage of geo-tagged imagery is that it can be displayed and browsed in a more natural way. Using a map-interface with push-pins or thumbnails representing each image (or image cluster) has become the de-facto standard, rather than just displaying a linear sequence of photographs. Figure 3.36 provides an example of a map-view with a collection of geographically organized images on Panoramio. Nevertheless, only a subset of users actually make use of geo-tagging functionality when creating and sharing their imagery. A search in May 2013 on Flickr for geo-tagged images returned **220 million** geo-tagged images, which corresponds to **3.6%** of the total number of images shared on Flickr (Source: www.flickr.com).

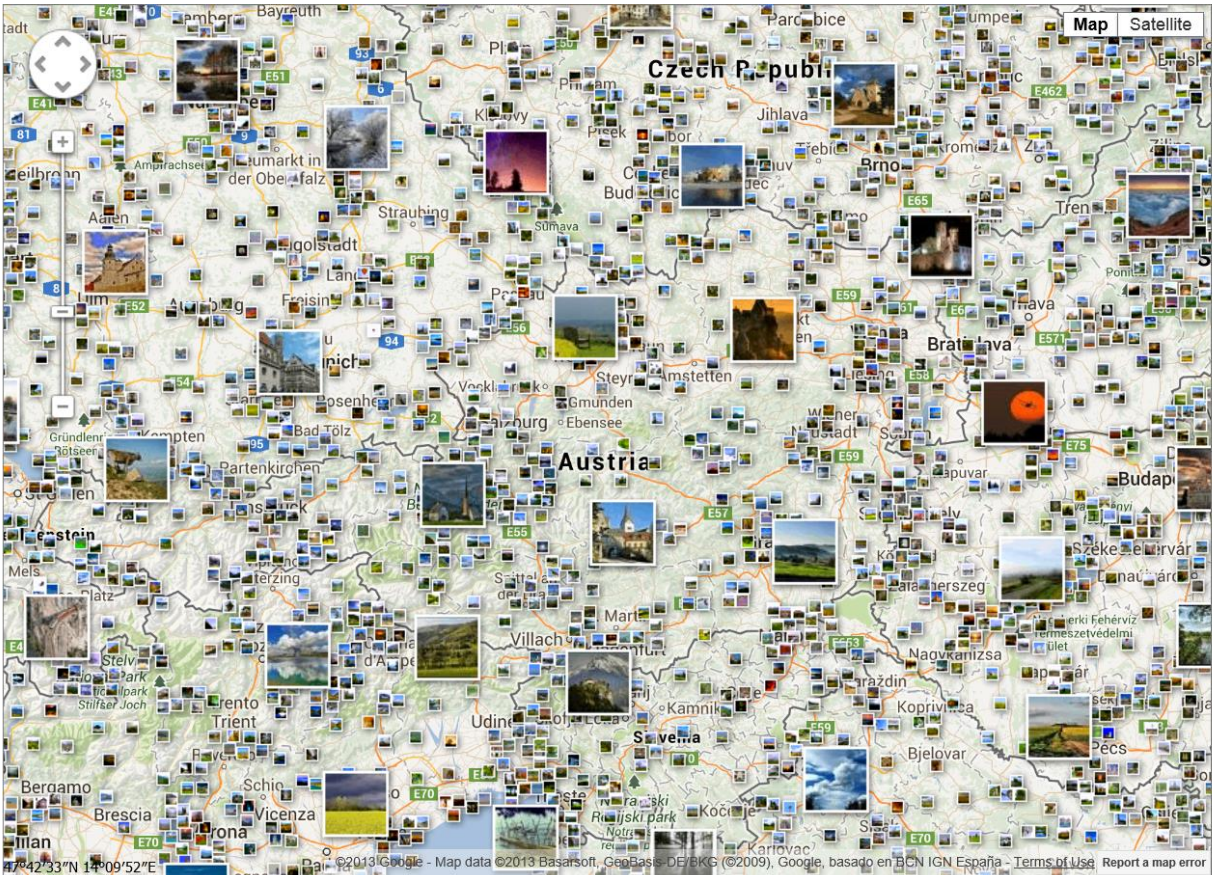


Figure 3.36 Geo-Coded Images in Austria on Panoramio

The use of GNSS for localization of images has two major limitations: availability and accuracy. In case of digital pocket cameras or SLR, the need to carry an extra device just for storing the location is obviously an inconvenience. Furthermore, older photographs, such as historical ones anyways lack such data. As a consequence, most (**96.4%** in case of Flickr) of the available photos in community photo collections are still not geo-tagged. Additionally, many services such as

Facebook allow geo-tagging photos, but don't share this kind of information with third parties through their application programmer interfaces (API) due to data privacy reasons and to protect the value of this information [273].

Accuracy is also a major issue. [274] found that 639 geo-tagged images on **Flickr** showed median positional errors between **58.5 m** and **1,606 m** for various geographic regions, while 794 **Panoramio** images had smaller errors between **0 m** and **24.5 m**. When tagging a mountain, accuracies of this magnitude may be sufficient, but in an urban setting – especially when viewed at a human-scale, such measurement errors are large. Though [275] reports that the median GPS accuracy achieved with an iPhone 3G smartphone device is **8 m**, GPS accuracy quickly deteriorates in urban settings due to various error sources such as multipath and atmospheric effects, and clock offsets [276]. These effects result in errors of up to hundreds of meters which translate into a completely different city block or landmark. Differential GPS or better error modelling to reduce the uncertainty [277] may be used as mitigations. However, the remaining errors may still be too large for some applications. Other infrastructure-based geo-tagging methods use triangulation between locations of known cell phone tower positions, or Wi-Fi hotspots, which achieve even less accurate geo-positioning at **600 m** and **74 m** median error respectively [275].

3.5.5 Benefits from Using Crowdsourced Data

Understanding the value of crowdsourced image data for mapping or general image search, requires the understanding of how they differ from systematically collected datasets, and which additional information can be extracted via analysis. An obvious and huge advantage compared to systematic aerial and human scale imagery, is that community created data come at essentially no cost. Further they get produced at ever increasing rates, on a multitude of different services, which leads to superior "freshness" compared to systematically collected data.

While the reliability of crowdsourced images in terms of their quality, significance to others and accuracy of metadata provided is generally lower for individual images than in case systematic collections, the power of using crowdsourced data comes from accumulating signals provided by many people [149]. Individual photos may either be of low quality, have incorrect geocoding [274] or contain incorrect or irrelevant tags. By combining the information from many users, the significant signals can often be separated from the noise, and hence provide new information.

An example of added value is the potential of finding and suggesting locations relevant to people. Due to the fact that crowdsourced images are captured by a huge community of people in many locations, with different intentions, and without a common schedule, their geographic distribution differes starkly from the regularly spaced flight lines of aerial imagery, or dense coverage of street networks with streetside imagery. A key aspect that becomes obvious from the geographic distribution of geo-tagged crowdsourced images (Figure 3.37, left), is that certain locations (e.g. Pike Place Market, Space Needle or Pioneer Square in Seattle) have a very high photo-density, while in other areas, much fewer photographs are available. Typically, these "hot-spots" are areas that are more popular, such as touristic attractions, sports stadiums, parks or other public places. [12] and [278] describe approaches for automatic hot-spot extraction from CPC data.

Not only do the distributions of community created photographs reveal interesting places which systematic patterns ignore (3.1.3), the coverage patterns at finer scales even indicate the exact objects of interest, such as statues in a park, or pieces of arts in a museum. An example scene from [159], illustrating this fact, was reconstructed in 3D from images contributed by multiple people. The different camera frusta for all pictures have been intersected in a top down view similar to "flash lights" (Figure 3.37 right), pointing out the location of a hot spot in the scene. In the shown example the hotspot corresponds to the location of a painter's canvas in an arts exhibition.

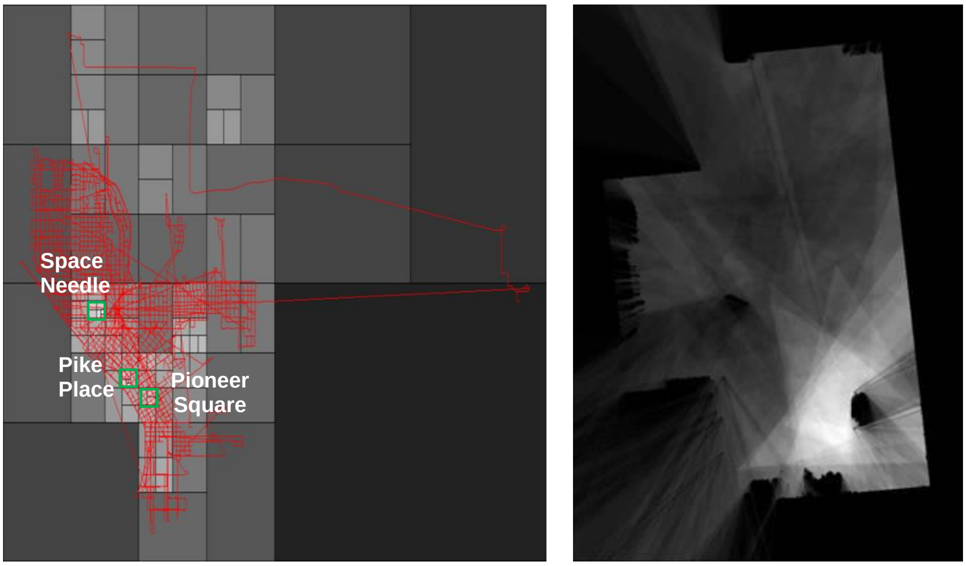


Figure 3.37 Distribution of Geocoded Images from Flickr Shown as Quadtree (Left). Smaller and Brighter Areas Have Higher Density; Example from [159] - Intersection of Multiple Camera Frusta Highlight Popular Object (Right);

By utilizing metadata information (Section 3.5.7) associated with CPC imagery, further information about the image content can be extracted, such as the name of a piece of art, commonly used adjectives for a certain location or object, or other related information.

3.5.6 Challenges

Challenges for using community created data assets for commercial applications arise from the fact that the content holders, such as Facebook, Yahoo, Google or Microsoft are not always willing to share these with outside parties. Even if programmatic data access to images by means of public application programmer interfaces (API) is provided, metadata such as geo-codes or tags are frequently only available to the content holders [273].

Further restrictions for using the uploaded data are by the content owners themselves, who impose certain license terms. While by default rights to use the data belong only to the content owner, many popular photo sharing services use the Creative Commons framework [150] providing people the ability to restrict only particular (e.g. commercial) use of their data or to make them freely available for all applications. Despite the benefit for the community of sharing images without license restrictions, individuals hesitate to do so. An analysis of 20,000 images uploaded to Flickr showed that about **95%** of images are restricted for private use only, thus preventing any commercial exploitation of the data.

Another challenge arises from the fact that photo assets and their metadata are not organized in a consistent way across different CPC services. This requires specific implementations when crawling third party CPC for relevant image data to aggregate them into a single database.

Further, as mentioned above, the reliability of crowdsourced data is frequently low, especially for individual datasets or users [149]. Intentionally created spam by malicious users is a common problem, which mostly can be resolved by congregating data from many users via majority-voting or removal (blacklisting) of spammers [279].

3.5.7 Metadata

While systematically collected data types like aerial and human scale imagery often include very precise metadata about the location (e.g. +/-2 m) and orientation of individual shots and even individual pixels within the frame, they often do not directly include additional metadata such as visible contents or background information about the image.

In contrast, crowdsourced image data often have less precise geo-tagging information, but instead are linked with additional kinds of metadata:

- User name / id of content owner
- Capture / upload time and date
- Latitude / longitude where data was captured
- Title and description
- Comments from different users
- A collection of image-tags

Particularly image titles, descriptions, comments and tags contain valuable semantic information, allowing further analysis by combining the data from multiple users. Tags are used to describe the image contents, location, people, etc. in order to allow searchers to find images concerning a certain topic such as a place name or subject matter.

An analysis of 122,000 geocoded Flickr images in Seattle showed that **68.6%** of all images contained a valid title (not including automatically created names such as image file names), and **85.9%** contained at least one tag. On average images contained **2.85** tags (350,000 total). Assuming that these ratios are representative for all 220 Million geocoded images, it means that Flickr data alone contain **150 Million** geocoded image titles and **630 Million** geocoded photo tags.

One way of extracting valuable information from crowd-sourced data is by accumulating data from many users in multi-dimensional histograms and projecting them onto individual dimensions. Example projections of the frequency for popular photo tags in the latitude/longitude plane of a map, as well as projections onto the time axis are visualized in Figure 3.38. The user frequency maps are computed as the logarithm of the number of distinct users using the same photo tag within a latitude/longitude bin. In this example a minimum of 2 users was required per bin in order to eliminate irrelevant tags used only by a single user.

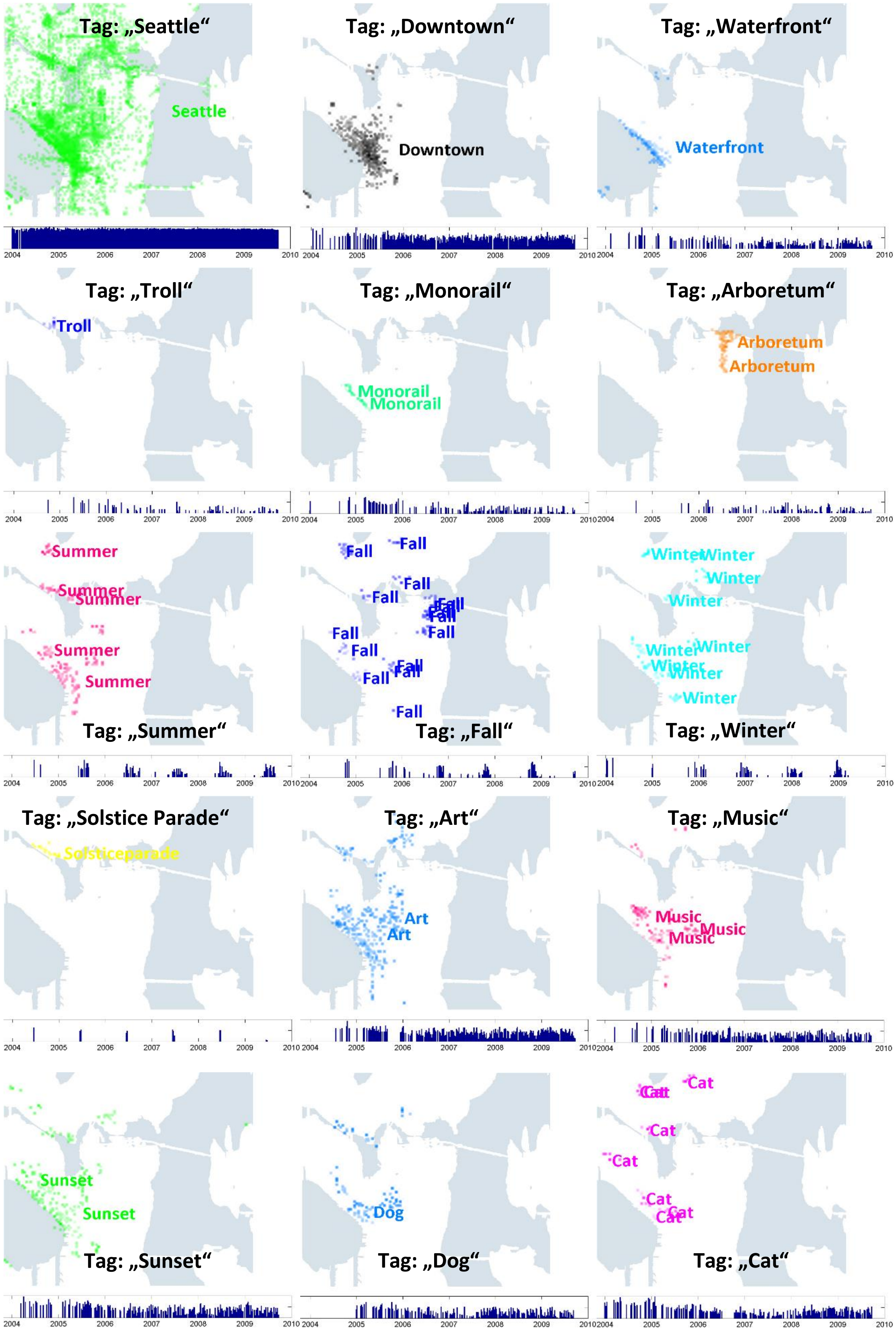


Figure 3.38 Sample Geographic and Temporal Distributions of Photo Tags

The temporal distribution (from 2004 to 2010) shows the actual number of occurrences for a given tag normalized by the total tags count within a particular time-span.

Many interesting questions can be answered based on density analysis results. What are popular destinations for photographers? Are they more popular during certain times of the year? Which events recur repeatedly? Where and when certain subjects (e.g. cats or dogs) are more frequently observed? Where to go for taking sunset pictures of the city skyline?

Some of the examples in Figure 3.38 reflect the boundaries of **defined geographic regions** (e.g. Seattle, Downtown, and Arboretum). Those regions are usually already known to maps providers, as they reflect areas such as cities, quarters, districts or parks, or linear features like streets or rivers. Note that the temporal distribution shows no significant trend or pattern for the tags "Seattle" or "Downtown".

Other tags point out interesting places which may not be included in traditional maps, but still constitute meaningful **geographic entities** for people (e.g. Waterfront, Troll or Monorail). These places may already be present as POI in map data provided by systematic sources like NavTeq or TeleAtlas. However their exact boundaries may be fuzzy. Therefore mapping providers could use the "crowd wisdom" to obtain better boundary definitions of places.

Examples of **temporal** entities with characteristic geographic distributions are the four annual seasons. It is remarkable that according to the distributions people apparently prefer different parts of a city during the summer, than during the fall or winter. **Recurring** organized **events** such as parades, which have a clear geographic scope as well as temporal pattern, can be detected using the same visualizations. For example the tag "solstice parade" occurs exactly once a year in a specific region in Fremont in Seattle. While information of this type may also be available from other sources such as web sites or blogs, crowdsourcing may provide another source to learn about popular events. The final tag category discussed here are tags describing representative **properties** or **scene contents** (e.g. Art, Music, Sunset, Cat or Dog) which are typically not part of existing maps. Note that the tags "art" or "music" have a distinct geographic distribution within the city, which suggests that the areas with higher density are more popular venues for the respective category.

All of the above types of tags can be used by online mapping services to enable users to search for events or entities according to their specific search intent or general interest, such as to answer the questions mentioned above. They further enable other scenarios such as auto-tagging of new photographs based on frequently used tags in particular geographic regions or during particular time-spans [280]. Similar to [281] and [282] we have proposed a method for automatic tag-ranking based on geographic relevance of tags obtained from CPC, enabling tag-recommendations as well as tag-based image search [283].

3.5.8 User Query Images

Query images play a specific role as they are either intentionally or unintentionally generated by the users of image based search application while submitting search requests. Although they are not captured with the intention to add information to an existing system, such as a mapping service, product database, or other visual service, they potentially may serve as a valuable source of information providing relevance feedback to the system [284]. If, for example users reject a specific result provided to them repeatedly, there is likely a problem with the specific data asset which should be investigated and fixed. On the other hand, if users repeatedly select a specific entry in a list of results not ranked highly, this result could potentially be ranked higher for future queries. Tracking statistics about the actual query counts, as well as recall and precision metrics is further helpful to identify problems with the service.

Further, the contents of the query itself, or related metadata identifying the identity or location of device or user can be used to personalize the search results [285].

One possible way of feeding back query information to a mapping system, would be to ingest the query data with the associated metadata to the system, if it can be confirmed that they actually contain geographically relevant data. This confirmation can happen automatically via image matching, or manually via data labelling. Sometimes, images of the same object or location have a large variation, such as images taken during the day and at night. Therefore, querying with an image taken at night, may have a lower likelihood of success if the index contains only daytime photos. However if a night-image query returns a successful match, adding it to the index may improve the recall rate for future users submitting night-images [48].

3.6 Semi-Systematically Collected Images

Many image sources can be clearly classified as systematic or crowdsourced data. However, there also exist hybrid forms of the two categories. On one hand programs are created by companies or other groups to motivate community data collection in certain areas, or according to specified rules. In this case the data captured are partially systematic because of the common set of rules, and partially unsystematic, due to the individual decisions during the actual data collection. An example for this case is Yelp, which enables users to upload and share imagery of food items or menus in a list of restaurants.

On the other hand we count UAV and MAV described in Section 3.2.6 as semi-systematic capture platforms, if they don’t follow systematic capture patterns, but record imagery in an arbitrary flight pattern above a particular area. However we consider the same platforms systematic, if they are controlled in more systematic flight patterns such as lines, or paths within a road network.

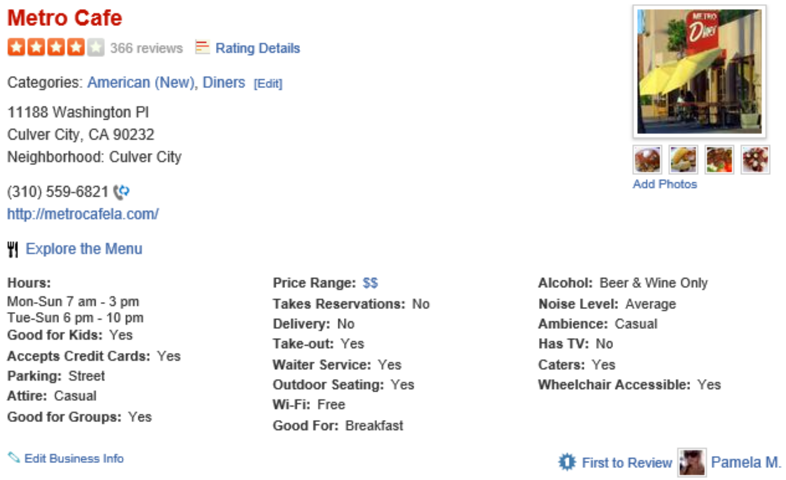
Public Programs for Community Collected Images

Business and nonprofit organizations often have the desire to acquire certain kind of image data such as pictures of storefronts or restaurant menus. However, they may lack the required financial assets or infrastructure to perform this tasks on their own at a large scale. Therefore, a motivation exists to leverage the community by means of crowdsourcing of defined tasks according to a set of rules and specifications. As individuals perform the captures according to their own schedule, but following pre-defined rules, we consider the process semi-systematic.

Organizations follow different approaches to incentivate participation in crowdsourcing. While some efforts actually involve micro-payments of a few cent per captured dataset [286], others rely purely on voluntary participation in the sense of a public Wiki [287]. In the latter case, typical motivations include the involvement in a community of people with the same goal, the satisfaction of improving a public service by one's own contributions, as well as the creation of virtual gratifications. Examples of virtual gratifications are achievements points for community contributions, badges for exceptional achievements, leadership perks and other similar rewards originally used in computer gaming [288].

Microsoft added a capability to its Photosynth Mobile application [254], allowing users to capture 360 degree panorama images inside of restaurants or other venues, and to link them to the respective business entries on Bing Maps. This enables community contributions to a mapping service by generating human-scale-like experiences of public points of interest, and is an example of purely voluntary participation.

Typically, location based social networks such as Yelp, FourSquare, UrbanSpoon and Google Places, make use of virtual rewards in order to motivate participants. These services enable users to explore public POI such as restaurants, hotels, museums or shops, checking in at the respective locations and providing feedback about them. Functionality for uploading and viewing user photographs of the respective places such as their storefronts, interior images and products is commonly provided, as shown for the example of a Yelp business listing in Figure 3.39. Storefront and interior images clearly have a geospatial nature as they relate to the location. Note the "First to Review" perk in the lower right corner, which is an incentive for people to contribute to the service. In order to keep the quality of the added data high, the community itself is involved by rating others people's comments and flagging inappropriate imagery to avoid scam.



Metro Cafe

★★★★☆ 366 reviews Rating Details

Categories: American (New), Diners [Edit]

11188 Washington Pl
Culver City, CA 90232
Neighborhood: Culver City

(310) 559-6821
http://metrocafela.com/

Explore the Menu

Add Photos

Hours:
Mon-Sun 7 am - 3 pm
Tue-Sun 6 pm - 10 pm
Good for Kids: Yes
Accepts Credit Cards: Yes
Parking: Street
Attire: Casual
Good for Groups: Yes

Price Range: $$
Takes Reservations: No
Delivery: No
Take-out: Yes
Waiter Service: Yes
Outdoor Seating: Yes
Wi-Fi: Free
Good For: Breakfast

Alcohol: Beer & Wine Only
Noise Level: Average
Ambience: Casual
Has TV: No
Caters: Yes
Wheelchair Accessible: Yes

Edit Business Info

First to Review Pamela M.

Figure 3.39 Yelp Business Listing including Crowdsourced Storefront Photograph and "First to Review" Perk

Other projects are actually set up as games with the goal of capturing many images in specific places for mapping. PhotoCity [289] was an alternate reality game (ARG) aiming at creating 3D reconstructions of university campuses from community images. Rewards were given to individuals or teams for adding new imagery in areas where gaps in the 3D reconstructions existed, thus motivating people to generate complete representations of a place to help support their team. Recently Google's Niantic Labs released an ARG called "Field Trip", which motivates users to explore nearby historic places, museums, restaurants etc., thus providing additional valuable information to Google [290].

3.7 Non-Geographic Images

Non-geographic applications are not directly the topic of this work, yet some methods used for general image matching are of interest in the geo-application, either for the purpose of location search or to create 3D reconstructions of scenes.

3.7.1 Image / Object Search

Several software companies such as Amazon, Google, Microsoft or A9 have released smartphone applications enabling visual searches by snapping real world objects. Products with planar front and back covers such as CD's, DVD's, Books, etc. are commonly indexed by such services, making use of image retrieval approaches using local image features [40, 30] such as KD-Tree based nearest neighbor search [44], bag of features [30], or more recent developments such as vector of locally aggregated descriptors (VLAD) [291]. The same methods are also applicable to non-planar objects, although different optimizations have to be made.

Various methods of searching real world imagery exist, such as by recognizing printed bar-codes, QR-codes [292] or Microsoft-Tags [293], by using optical character recognition (OCR) [238] to detect and search by using text, or via facial recognition [294, 295]. However, these alternative methods are out of scope of this thesis as they are only partially applicable to online mapping.

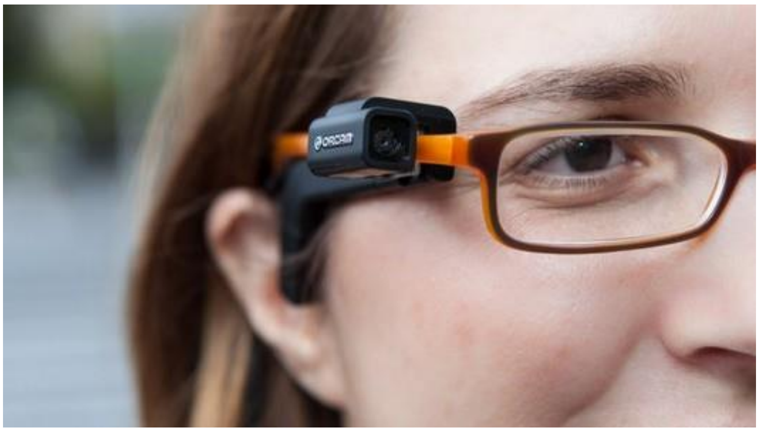


Figure 3.40 "OrCam" Wearable Computer Vision Enabled Device offering Visual Object Recognition for the Visually Impaired [296]

Wearable, camera-enabled devices such as Google Glass [297] or OrCam [296] (Figure 3.40) have taken this idea one step further, by providing continuous object recognition functionality, and thus extending the sensory and cognitive abilities of people.

3.7.2 Product Images

Searching for relevant product information within seconds by snapping a photo is a relatively new way of mobile search. This may be motivated by the need to obtain product reviews, price information, technical specifications or other helpful data for the user to decide whether or not to buy a product. It often involves querying a web-services hosting tens of millions of product images and metadata. Nistér et al. [30] first demonstrated a solution for this problem, based on TF-IDF scoring using quantized local image features, on a dataset of 40,000 CD cover images, which matured into a visual search feature in Bing powered devices such as Windows Phone. Figure 3.41 features a search result returned by "Bing Vision" [298] for a query image of a book cover.

A database of product information and product imagery is required for this purpose. Similar to geospatial applications, the imagery as well as product information can either be obtained in a systematic manner, or via crowdsourcing. Systematic images for product search get acquired by accumulating data from content providers such as online warehouses (Amazon), publishing companies, or companies specialized in creating product databases [299, 300, 301]. Alternatively, crowdsourced data acquisition may be used, which follows essentially the same rules as for geospatial data, by giving people the option of uploading imagery for missing products to a system, and offering information or other benefits in return.

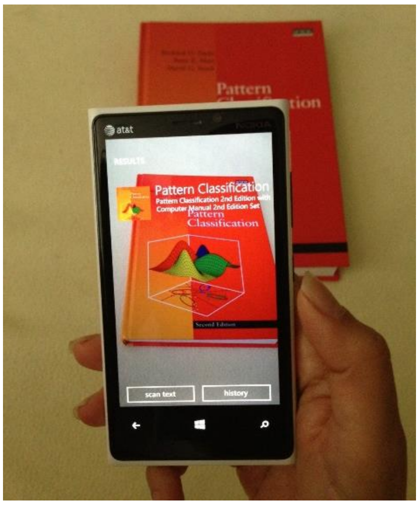

Pattern Classification
Pattern Classification 2nd Edition with Computer Manual 2nd Edition Set
scan text
history

Figure 3.41 Sample Result of Visual Product Search using Bing Vision on Windows Phone

The product search problem is non-trivial. Product database sizes often exceed many millions of individual products, which poses challenges to the scalability of the algorithms and infrastructure required. Large variations in the appearance of the product due to differences in the camera types and poses, lighting and occlusions pose challenges for an image retrieval system to achieve close to 100% recall rate. On the other hand, having a very large index of candidate images, and potentially many non-relevant query images, the problem is also hard in terms of the reliability (precision) of the results. Finally, due to limited data upload bandwidth on 3G network connections, only a limited data volume (e.g. 20 kB) can be uploaded per query, in order to minimize the query time and to maximize the user experience.

Although the index of products served by Bing Vision currently only includes planar products like books, CDs, DVDs and computer games, extending the scope to 3D objects such as product packaging or geometrically rigid products would be a feasible extension. However 3D object retrieval poses additional algorithmic challenges which are an active research area [302, 303].

3.7.3 Non-Geographic 2D and 3D Objects

While product search constitutes the bulk of non-geographic visual search, other application extend beyond this scope, including search of general 2D and 3D objects such as toys, logos, traffic signs, magazine and newspaper pages, artwork, faces, text or 2D barcodes.

Products such as Microsoft's Bing Vision or Google Goggles include functionality for searching objects in the above categories using a combination of algorithms developed for the different search verticals [298, 304]. For example, Google Goggles added a feature for visual recognition of Sudoku puzzles as shown in Figure 3.42.

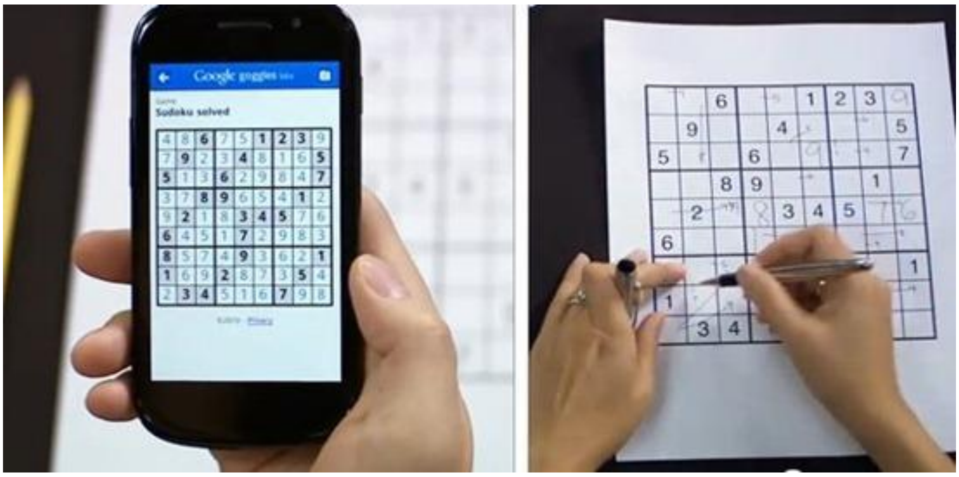


Figure 3.42 Demonstration of Sudoku Search Feature in Google Goggles [305]

3.8 Summary and Outlook

In this chapter we have defined a set of selection criteria for imagery in the context of internet mapping, including a definition of "geospatially relevant images", and furthermore provided an overview of the various types of images available. Parallels to non-geographic images in terms of the potentially applicable image matching techniques have been pointed out.

We have presented a classification of the various image types based on the capture patterns and processes used, into systematically, crowdsourced and semi-systematically collected images. While systematically collected data are evenly distributed, crowdsourced data from CPC often are clustered around popular hotspots and contain potentially useful community-created semantic metadata. We have found that the estimated data volumes required for a 3D world model of **1,700 Petabytes** [1] by far exceed the available geographically relevant CPC data assets of **40 Petabytes**. Therefore, while CPC data cannot fulfill the requirement for a comprehensive coverage of all locations in the world anytime soon, they can still provide useful signals about popular locations and user behaviors. Furthermore, work by [11] has shown that particular locations in the world such as Rome or Dubrovnik are already covered densely enough in CPC to allow 3D reconstructions of large parts of the city.

Based on the understanding of the characteristics of systematically collected and crowdsourced image data provided here, we will evaluate possible ways of integrating these assets with the aim to advance internet based mapping services in the following chapters.

4 Human Scale Image Data in Bing Maps[2]

This chapter covers two main contributions to the introduction of human scale image data in internet mapping. Both innovations matured into a practical application in Bing Maps.

First we develop key requirements to efficiently capture human scale imagery on public streets. We then propose a camera system we call UltraCam-M to satisfy these requirements. Figure 4.1 gives a preview of the image data and 3D geometry captured by this system.

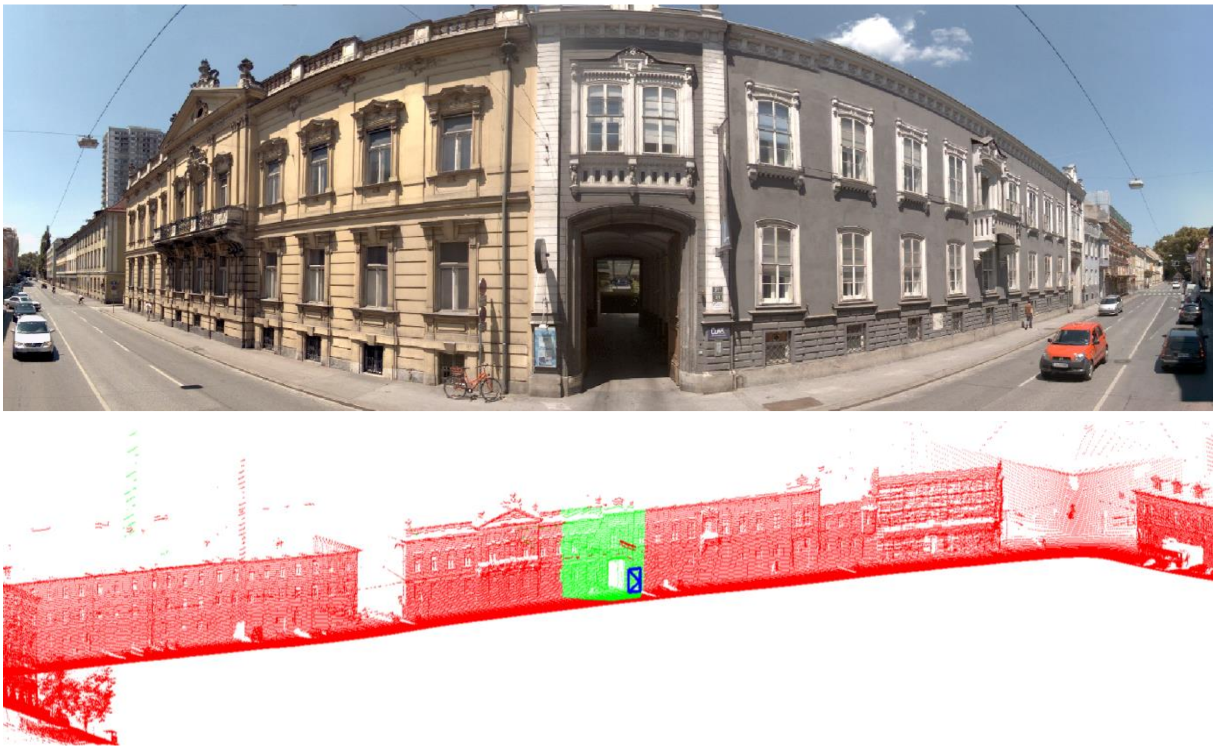


Figure 4.1 Bing Maps Streetside Panorama (Top) and Corresponding Point Cloud (Bottom) Captured with UltraCam-M System in Graz, Austria

Since any internet publication of human scale data requires the compliance with privacy regulations of different countries, one needs a system able to detect and obfuscate private license plates and people in streetside images. We present a workflow to achieve such privacy protection

[2] Both the streetside capture system as well as the workflow for privacy protection described in this chapter are the result of project work by the Microsoft Vexcel team in Graz, Austria and the Virtual Earth Research team in Bellevue, WA and Boulder, CO. Streetside imagery in many different location in the US and Europe obtained with both systems are available via www.bing.com/maps.

and show its application at a success rate above 95% for clearly visible and 80% for vaguely visible private objects while achieving a relatively low false positive rate (FPR) below 2.5%.

4.1 Streetside Image Capture for Internet Mapping

2005 saw the successful release of orthomaps and aerial photography based 3D city models on mapping platforms such as Google Maps and Bing Maps [109, 55, 56]. As these image types lack in similarity to the "human experience" of walking and driving, there was a natural desire to add images at higher levels of detail. Streetside images were the logical choice at the time, due to the higher fidelity of the images, as well as the wide range of possible applications. Streetside imagery could be used to texture the building models at higher resolutions, or to support 3D reconstruction at a much higher level of detail.

A problem to be solved in this context was the efficient data capture over large areas. In contrast to a person walking through an area and taking images of building façades and other urban objects manually, e.g. with an SLR camera, many of the steps required during this process have to be automated for efficiency reasons. This includes the decision, when and where a new image should be captured, which camera settings (such as the exposure times) should be used based on the scene appearance and vehicle velocity and how best to support the automated data management. For example, a system needs to keep track of where images have already been captured, in order to manage redundancy in the capture process.

4.2 Mobile Mapping Systems

The area of mapping using terrestrial mobile sensor platforms has been an active research topic since about 1990 [306, 307, 308]. A number of research and commercial systems for so-called "Mobile Mapping" existed in 2005, and have evolved since. Multiple overviews of such Mobile Mapping Systems (MMS) and their progress were compiled [309, 310, 311, 32]. One driving idea is the autonomous vehicle based operation of a variety of sensors, both imaging and non-imaging. Another driving force is the documentation and management of outdoor facilities with road networks, building-façades, traffic signs, vegetation along roads etc.

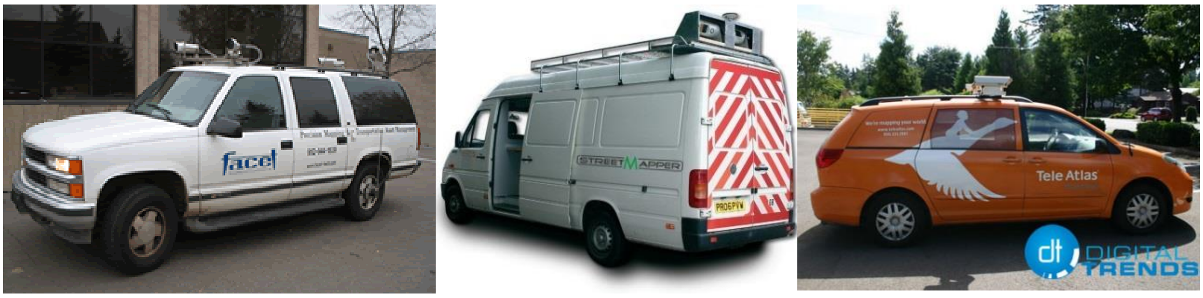


Figure 4.2 Different MMS available in 2005 and 2006: [246] MMS for LBS Data Capture (Left); [248] Street Mapper MMS for Precise 3D Mapping (Center); TeleAtlas MMS for Street Mapping (Right) [312];

Hence during the advent of human scale imagery in internet mapping, MMS were primarily used in the context of GIS and navigation systems and to support fast collection of mapping data for applications such as asset monitoring, disaster response and accident investigation. MMS typically include a combination of different cameras, positioning sensors (such as GPS), inertial and magnetic sensors, odometers as well as distance measurement devices (LiDAR). Additionally they require some form of (ideally redundant) data storage, an uninterrupted supply of electrical power during capture, as well as an easy to use user interface providing feedback about the status of the system, and allowing to control the data capture.

Despite the existence of several commercially available MMS in 2005 (See Figure 4.2), they were usually optimized for very high precision mapping and surveying of small areas, or the documentation of street signs for navigation purposes in large street networks. However no such system satisfied the internet requirements, particularly the need for capturing immersive 360 degree images of street locations, which became the de-facto standard for internet mapping today.

In the following we describe the requirements for a MMS aimed at mapping large street networks efficiently as well as our approaches to comply with them. Finally we provide an overview of the key elements of our proposed system.

4.3 Requirements and Proposed Solutions

4.3.1 Image Data

Requirement:

For internet mapping, we primarily are motivated by visually pleasing and immersive image data products such as the 360 degree panorama images described in Section 3.4.5. In order to capture such data in high quality, a cluster of individual cameras arranged in a hemisphere is used to capture raw imagery, which then gets stitched into a seamless panorama image.

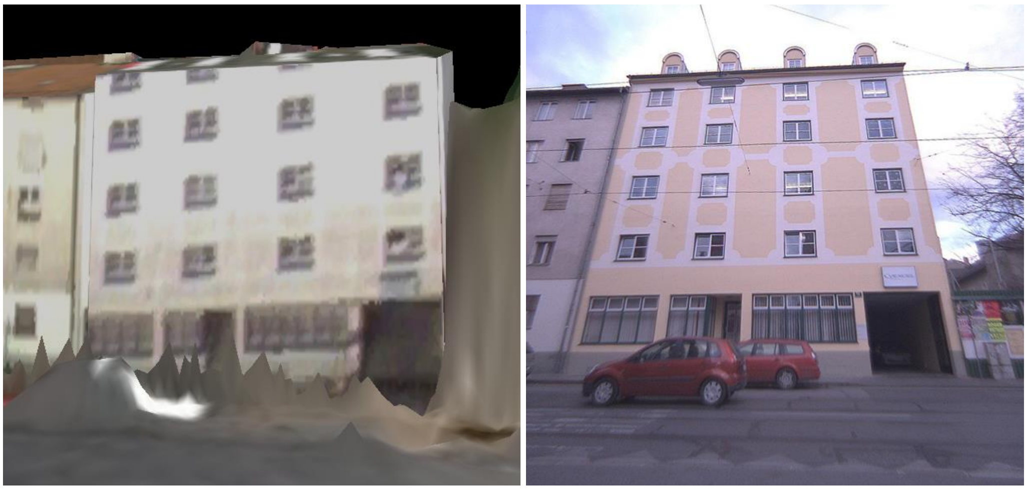


Figure 4.3 3D Building Model Textured with Aerial Imagery at 30 cm GSD (Left) and Potential Improvements using Streetside Image Captured at 2 cm GSD (Right).

A simple requirement defining the sensor resolution required is the GSD at which objects such as building façades are obtained. Based on the desire of capturing business names and traffic signs with stroke widths of **25.4 mm** (1") at typical distances of up to 20 m, the angular frustum of a pixel needs to be **1.27 mrad** or smaller [313].

A second motivation in addition to immersive panorama exploration is the need to add high resolution textures to existing building geometries obtained by aerial 3D reconstruction [231]. The quality difference between 30 cm aerial GSD with additional foreshortening due to the capture angle and 2 cm streetside GSD is evident in Figure 4.3. In addition to improving the texture resolution, façade modelling techniques [314, 315, 316] can be used to generate significantly better 3D models than from aerial images alone.

However, texture resolution can alternatively also be improved by means of procedural façade descriptions and generating synthetic façade views at arbitrary levels of detail [225, 93, 317, 314]. Thus the two images in Figure 4.3 may eventually lead to the same procedural description and texture quality in a synthetically created view.

Proposed Solution:

The proposed system, which is explained in more detail in Section 4.4 involves a total of 12 individual cameras in an arrangement fulfilling the requirements both for panoramic image capture as well as 3D building reconstruction.

The angular resolution requirement of 1.27 mrad is approximated by using a **6.45 µm** CCD camera (Prosilica GC1380) with an image size of **1.4 Megapixel** (1024 horizontally * 1380 vertically) in combination with a **4.8 mm** focal length lens (Schneider COMPACT 1.8/4.8). The resulting field of view of the individual camera images is **65 degrees** horizontal and **90 degrees** vertical.

4.3.2 Automatic Exposure Control

Requirement:

Exposure control is an essential topic of automated camera systems in order to obtain optimal image quality using a sensor with a given bit depth and sensitivity. Typically the aim of exposure control is to achieve median intensity values in the center of the dynamic range despite variations in the illumination and scene content [318]. Neither over-exposure (clipping) nor under-exposure of images is desired as it irreversibly affects the image quality. Depending on the scene to be observed, the illumination and the camera sensitivity, the exposure time needs to be set such that the desired radiometric parameters of the recorded image are achieved. Since the proposed system contains multiple individual cameras, the exposure control needs to happen for each camera individually.

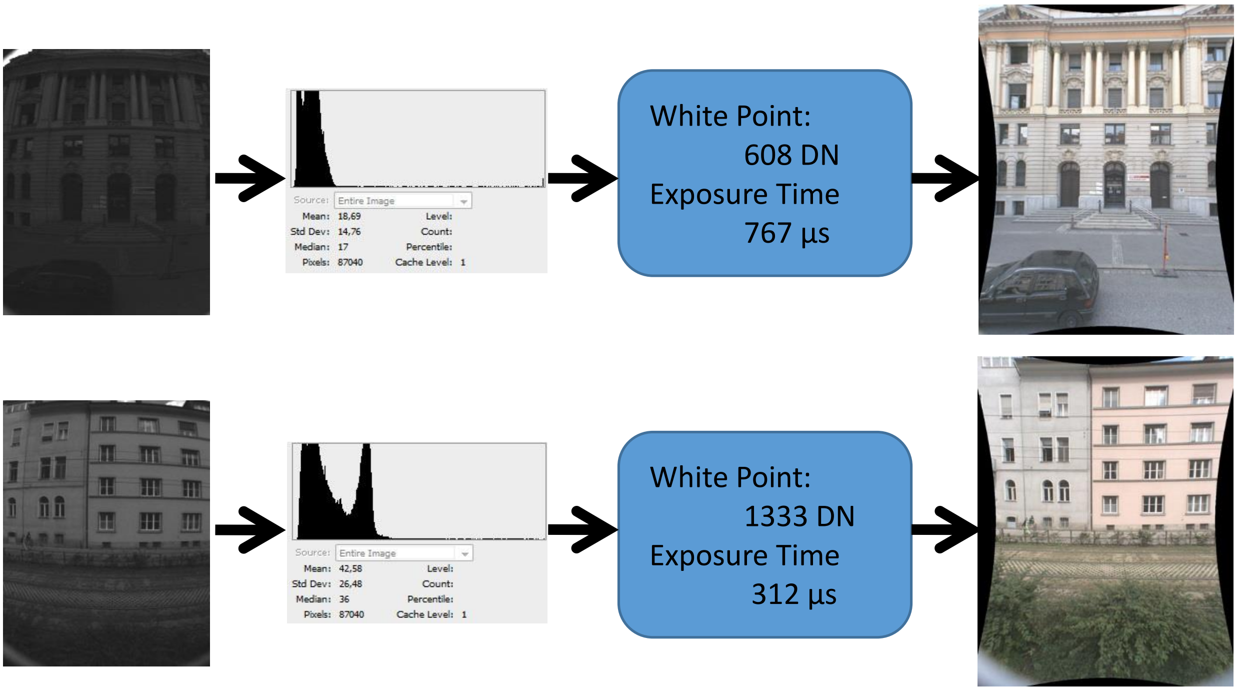
In a dynamic application such as mobile mapping, where scene content and illumination can change significantly from image to image, this requires fast exposure time updates within less than approximately **50 ms** to cope with such changes (Assuming a vehicle speed of 50 km/h the forward motion within 50 ms is 0.7 m).

The maximum valid exposure time is defined by motion blur. For a given forward motion velocity of 50 km/h and a maximum forward motion in object space of 1 pixel (25.4 mm on the façade) the maximum exposure time is **1.8 ms**. This value needs to be updated dynamically based on the current velocity and distance to the façades.

Proposed Solution:

Many available camera systems perform exposure control based on the analysis of a series of previously captured full-resolution images, a method which only responds slowly to changes in illumination and image contents in dynamic applications [318]. Other cameras make use of a separate light meter [319] observing the same field of view as the camera as an input for exposure control. While this enables relatively fast exposure time updates within milliseconds, it requires additional components (electrical, optical) and hence make the system more complex.

In [34] we proposed a novel method of performing exposure control for dynamic scenes. It requires collection of a quick view image shortly before each full resolution image using pixel binning, from which the ideal exposure time for the next full resolution image gets computed by means of histogram analysis. Particularly the white point, which is the 99th percentile gray value in the image is used for this determination.


Source: Entire Image
Mean: 18,69 Level:
Std Dev: 14,76 Count:
Median: 17 Percentile:
Pixels: 87040 Cache Level: 1

Source: Entire Image
Mean: 42,58 Level:
Std Dev: 26,48 Count:
Median: 36 Percentile:
Pixels: 87040 Cache Level: 1

Figure 4.4 Example of Exposure Time Control based on Histogram Analysis of Quick View Image (White Point Detection)

Because the quick view image covers largely the same field of view as the subsequent full resolution image, the full resolution image is typically exposed well and contains a higher dynamic range for each image than it could be achieved with conventional methods. The low resolution is advantageous because the data transfer and evaluation of the smaller image size can happen much faster (within less than 1 millisecond). During this time, the image content should only change marginally before the full resolution image can be recorded.

Two examples for the exposure time computation are given in Figure 4.4. In the first example, the quick view image is darker than in the second example, which leads to a lower white point and higher exposure time than in the second example. As a result, even though the two quick view images differ significantly in brightness, the full resolution output images have nearly the same brightness.

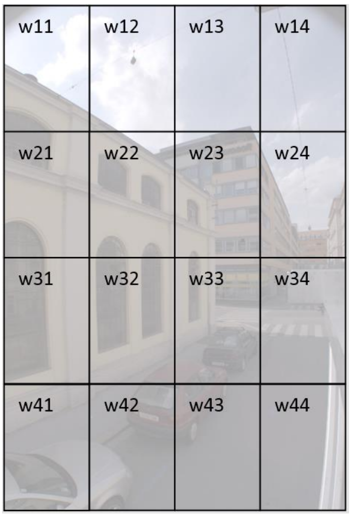


Figure 4.5 Weight factors for different areas of the image

A modified variant of the exposure control may be used to avoid that a particular region of the quick view image (e.g. top part including sun or sky) influences the overall brightness of the output image too strongly. In contrast to the method described above, the histogram, white point and exposure time are not computed for the whole image at once, but for sub-regions of the image. These sub-regions can be arranged arbitrarily. An example of a 4·4 rectangular tiling is shown in Figure 4.

Each sub-region is associated with a weight factor, based on the emphasis that should be given to the regions for the exposure control. Hence, the resulting overall exposure time for a sensor is computed as the weighted average of the individual exposure times for each region according to

$$t_{exp} = \frac{\sum_{i=1}^{n} w_i * t_{exp,i}}{\sum_{i=1}^{n} w_i}$$

4.3.3 Automatic Trigger Control

Requirement:

To achieve full automation during capture, the various image sensors need to be triggered by an automated mechanism at intervals determined by the respective application. While for internet exploration of streetside panoramas, capture intervals of 2 to 5 meters are likely sufficient and reflect the data on mapping sites today, other applications may have different requirements.

Particularly for 3D reconstruction of building façades a certain amount of overlap between consecutive frames is required. Based on experience with aerial imagery **60-80%** forward overlap [171] may be a valid choice to be evaluated. Figure 4.6 visualizes **60% forward overlap** in the streetside case.

Using a fixed camera trigger rate (e.g. 10 fps) is often insufficient for this purpose. Apart from the trigger rate, capture interval and forward overlap are also affected by the forward velocity of the platform, the distance to the façade as well as the rate of turn of the vehicle in curves.

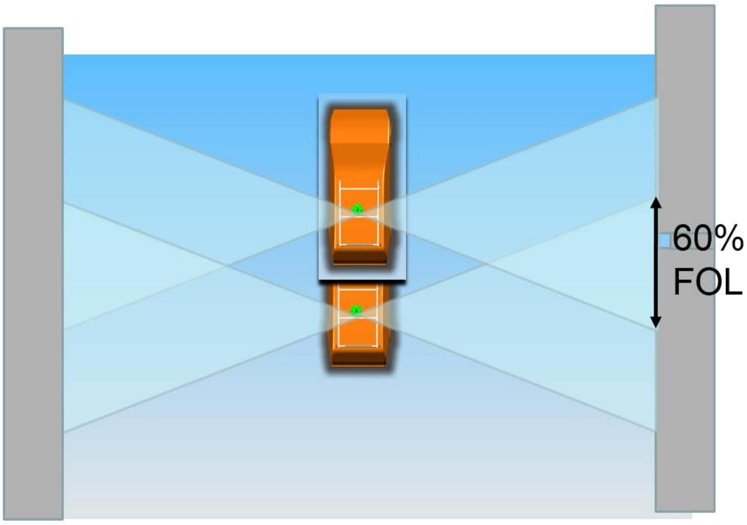


Figure 4.6 Visualization of 60% Forward Overlap for Façade Images

In case of a capture velocity of 50 km/h the required frame rate for the shortest realistic façade distance of 3 m and a desired forward overlap of 60% is **8.4 fps**. For 80% overlap this number increases to **16.8 fps**.

Typical approaches for the trigger control of mobile mapping system are based on position measurements from a global navigation satellite system (GNSS) receiver, such that images are taken in equidistant positions [320]. The disadvantage of this approach is that it fails in case of bad satellite reception. Additionally the effective image overlap varies as a function of the distance to the façade and the rate of turn of the vehicle. To mitigate the problem of bad satellite reception, relative position measurements using odometers or inertial sensors may help, but even then façade distance and rate of turn are not taken into account. Visual Odometry using images [321] would potentially provide the required information, although real time processing requires additional compute resources.

Proposed Solution:

In [37] we proposed a novel solution for the problem of achieving a minimum desired forward overlap for the side looking cameras by means of trigger control. The method takes into account variations in velocity, rate of turn and object distance to the façade. For this purpose, the current velocity of the vehicle is measured using an odometer or distance measurement instrument (DMI), the rate of turn is measured by an inertial navigation system (INS), and the effective object distance to the façades is scanned by making use of a range measuring device such as a laser measurement system (LMS).

4.3.4 Depth Information

Requirement:

In order to provide a more intuitive and tangible experience of a real world environment, human scale functionality on web-based mapping sites often makes use of depth information in conjunction with image data. This allows exploration of the geometric scene structure by hovering over it with a 3D mouse cursor and selecting a specific point on a scene surface by clicking. The latter may initiate a transition to the image closest to the respective point. Figure 4.7 visualizes this behavior in case of Google Maps. The "cursor" is displayed as a rectangle placed parallel onto the scene geometry.

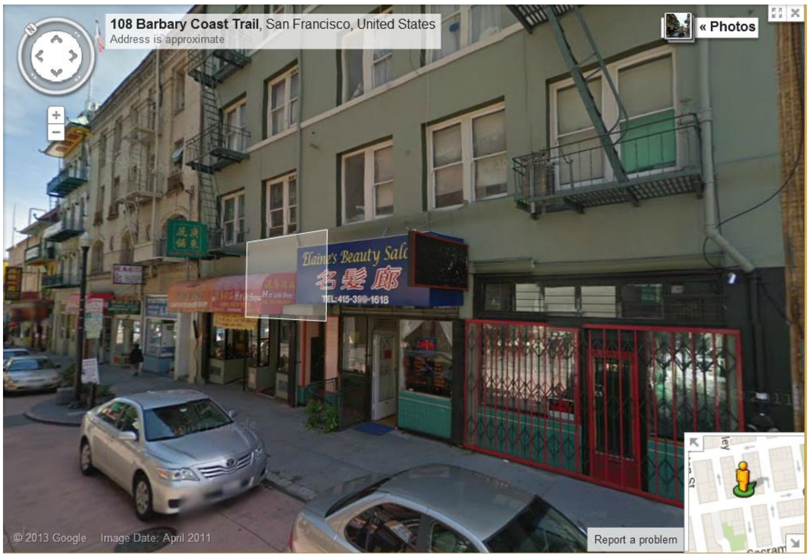


Figure 4.7 Visualization of 3D Cursor in Google Maps Used to Explore and Select 3D Scene Elements

Depth information for each panorama can further be used to facilitate visually pleasing transitions between locations, by rendering the image data superimposed onto the scene geometry, and blending between multiple views during geometric transitions. This approach of combining depth information with photography is similar to combining DSM with aerial orthophotos as explained in Section 3.1.4. Due to the subsequent mesh simplification, we assume that an accuracy requirement of **5%** of the measured raw depth is sufficient.

Proposed Solution:

Primarily two approaches are available to obtain depth information: image based and laser based. Terrestrial LMS are commonly used as part of MMS to obtain 3D geometry for streetside scenes, as described in 3.4.3. As LiDAR data may still contain noise due to measurement errors or caused by dynamic scene content, it is common practice to apply further processing on the raw 3D geometry estimates in order to filter out noise. Further, to facilitate transmission of depth data over the internet and client-side rendering, the geometric structure have to be simplified via "mesh simplification" [322] to a low polygon count.

Alternatively, a range of image based solutions have been proposed since 2005 to extract depth information of façades and other streetside objects by means of SFM to in combination with dense matching techniques [172, 323, 314]. While such solutions don't depend on additional sensors apart from cameras, they require a significant amount of computation either on regular CPU cores or graphics processing units (GPUs) optimized for parallel processing [324].

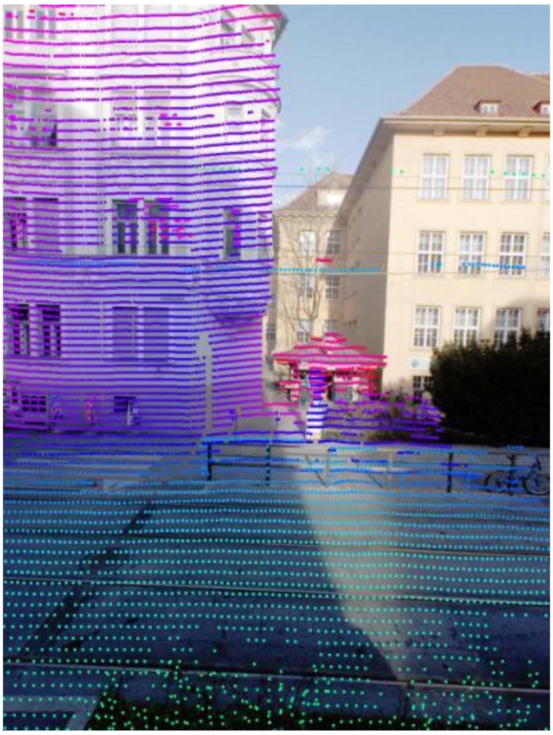


Figure 4.8 Depth Profile from Vertical Laser Scanner Overlaid onto Image during Postprocessing

In 2006, to reduce the dependency on then less advanced computer vision algorithms we used a setup of **2** or more **LSM** [325] in the proposed system in order to obtain depth measurements up to **80 m** away from the capture platform at accuracies of **5 cm** or better. Figure 4.8 shows an example of the depth information obtained using one vertically arranged LSM superimposed onto one of the side-facing camera images. Due to the forward motion of the vehicle the geometry of the scene can be determined although the laser scanner only measures individual 1D depth profiles at a rate of **75 Hz**. The exact layout of the LMS is described in Section 4.4.

4.3.5 Positioning and Orientation

Requirement:

With the aim of facilitating the correct placement of image data on the map, as well as to support the logistics during capture, continuous tracking the position and angular orientation of the sensor platform at all times is a key requirement. Therefore, MMS usually contain some form of navigation system, tracking the location and orientation of the platform over time. The recorded data can hence be used to localize individual panorama images and laser scans.

The required localization accuracy depends strongly on the intended application. If the orientation data should be used directly for positioning panoramic images in internet maps, stricter accuracy requirements need to be fulfilled. Based on our estimate of the user tolerable errors a positional accuracy of **2 m** in combination with a rotational accuracy of **5 degrees** seems plausible. In case the initial panorama orientations are corrected during postprocessing by means of "Shape from Motion" (SFM) using images [24], the tolerances for raw positional errors depends mostly on the robustness of the method used. For this case we assume that a **5 m** positional accuracy should be sufficient to facilitate the correspondence search around intersections during such postprocessing, and that SFM algorithms can deal with rotational prior uncertainties around **20 degrees**.

Proposed Solutions:

Typical solutions consists of one or more of the following:

- An absolute positioning system providing coordinates in a local or global coordinate system. Positioning systems often rely on some form of infrastructure such as a network of satellites, dedicated positioning anchors or mobile communication infrastructure. Examples of global satellite navigation systems (GNSS) are the Global Positioning System (GPS), Global Navigation Satellite System (GLONASS) and GALILEO.
  Surveying grade GNSS accuracy may vary from sub-meter to tens of meters depending on various environmental factors. Errors caused by atmospheric effects, can be reduced by using differential measurements relative to satellite ground stations. However GNSS accuracy may still be limited in urban areas due to bad satellite visibility. Hence GNSS often are used in conjunction with complementary systems such as inertial measurement units or odometers, which provide more accurate relative pose updates [134].
- An inertial measurement unit (IMU) providing relative acceleration measurements in 3 axes, and angular velocity measurements in 3 axes, which can be integrated to obtain relative velocity and angular differences. Due to the integrative nature of these measurements, they are prone to drift errors, and therefore should be used in conjunction with a complementary drift free system such as GNSS or a compass.
- A magnetic compass provides an independent and drift free measurement of the sensor orientation based on the Earth's magnetic field. Magnetic sensors work more reliably in rural environments as their accuracy is reduced in urban settings due to interferences of buildings and electrical infrastructure with the Earth's magnetic field.
- Vehicle based systems frequently utilize other positioning sensors such as wheel-based or optical odometers to provide low noise relative position estimates in a single dimension.
- Laser Distance Measurement systems such as LiDAR (Light Detection and Ranging) are used frequently to augment the image data with corresponding scene depth information. In some applications, time synchronized depth measurements are also used to support the pose trajectory estimation in combination with the other sensors mentioned above such as via Monte Carlo Localization [326].

A range of integrated position and orientation systems for use in MMS are provided by manufacturers such as Applanix [252] or Topcon [327], which combine GNSS, INS and depth by means of sensor fusion. The specified positional accuracy of the Applanix POS LV 120 under optimal condition of continuous GPS visibility is better than **0.02 m** horizontally and **0.05 m** vertically, while the rotational accuracy is better than 0.1 degrees in all axes. Under worse conditions of GPS outages of 60 s, this degrades to **2 m** horizontal and **0.8 m** vertical accuracy and rotational errors of **0.2 degrees** (roll, pitch) and **0.65 degrees** (heading), which is still within the specification provided above.

The various sub-systems have to be time synchronized with each other and with the camera trigger signal, to facilitate the time alignment of their measurements. For this purpose, we have proposed a method for automatic synchronization of measurements from multiple sensor subsystems with **+/- 5 ms** accuracy, and aligning them using a common time base [36].

Measurements of these different components are typically integrated using sensor fusion such as by Kalman-Filtering - [328] - to achieve a trajectory which is optimal in terms of low positional and rotational errors in the fused output despite the errors in each of the individual sensor sub-systems. A method to combine measurements of a velocity sensor (such as an odometer) with an inertial acceleration sensor to correct for errors in both individual systems was proposed as part of the UltraCam-M streetside sensor development [35].

Alternative approaches exist which use the captured image data to estimate trajectory by interest-point based image matching using SFM or "Visual Odometry" [24], and combinations of such with the methods mentioned above.

4.3.6 Time Synchronization

Requirement:

All of the laser scanners and cameras used in the proposed system have to be time-synchronized to each other and to all other sensors in the system to within **+/- 5 ms**, to support easy data alignment during processing. This tolerance leads to a maximum relative platform motion of 14 mm at a typical velocity of 50 km/h which is below the image GSD of at 20 m distance.

While cameras can be triggered at variable trigger rates (up to 15 times per second), the laser scanners are usually operated at a fixed line frequency (e.g. 75 Hz). This needs to be considered during the time synchronization.

Proposed Solutions:

In [36] we have proposed a solution for synchronizing multiple sensor sources based on time stamps for individual signals determined by a PC system clock as well as the clock of a trigger circuit. We have verified that the synchronization errors with this solution are below **+/- 5 ms**.

4.3.7 Data Volumes and Redundancy

Requirement:

Due to the increased sub-cm image resolutions and short capture intervals of less than 5 m for human scale data, their data volumes can substantially exceed corresponding numbers for aerial imagery. This imposes significantly more stringent requirements during data capture, transmission to the processing facility, storage and processing.

A single stitched orthophoto image of a downtown area spanning **10·10 km^2** at **15 cm GSD** is approximately 4.1 Gigapixel in size. Assuming a redundancy factor of 10 to permit automated data processing the recorded raw imagery for this area adds up to **41 Gigapixel**. However, the same area captured with a **5 Megapixel** panoramic streetside sensor at **4 m** capture interval (assuming about 65·65 city blocks of 150 m spacing) consists of approximately **330,000** individual panoramas, resulting in about 1,700 Gigapixel net worth of data. Assuming a redundancy factor 2 due to duplications, this number doubles to **3,400 Gigapixel**, about **80** times as much as in the aerial case. At higher resolutions, and with denser coverage including off-street and building interiors, the data volumes for the same area can even reach tens of TB of un-compressed images.

Another problem requiring attention when capturing large areas with human scale sensors, is the need to minimize undesired duplications in the captured data. In order to minimize cost and time spent, specialized routing strategies have to be developed with the purpose of capturing a designated area within the shortest possible time, and / or following the shortest possible path through all required vertices and nodes of a graph.

In case of failures of system failures, or if data problems are detected after capturing, further logistical efforts are needed to facilitate re-capturing of affected regions.

Proposed Solution:

The proposed sensor system includes a replaceable data unit (see Section 4.4) containing a total of 14 hard drives which a combined storage volume of **4,500 GB**. This amount is sufficient for storing streetside data during a whole day of continuous data capture.

The capture software validates the recorded data by means of checksums, and also controls data export onto external drives, which can hence be shipped to the processing facility.

4.3.8 Data Processing

Requirement:

The preparation of human scale data for internet mapping requires substantial amount of data processing, including preprocessing of individual panoramic images, and the selection of a relevant subset for publishing.

Part of the individual panorama processing is done to obtain RGB color images from the raw data using Bayer-Demosaicking [329, 191], and correct for radiometric and geometric imperfections. These imperfections are either determined during an offline calibration procedure or as via self-calibration. Hence the individual sensor images have to be stitched together into a single panoramic image, which can be stored in various formats as explained in 3.4.5. Depth information collected by the laser scanners also has to be processed into a geometry model of the scene, as explained below in Section 4.5.5.

Another step required for individual panoramic image individually concerns the automatic detection and obfuscation of private image contents. The requirements for privacy protection as well as a proposed solution are illustrated in Section 4.5.

Altogether, the steps mentioned above take a significant amount of time for individual panorama images published to the mapping site. For the initial version of the streetside pipeline used for Bing Maps, this time was approximately **2 minutes** per published panorama image on a single CPU core. Considering that a downtown area including 330,000 individual panoramas would take 3 days to capture with one system, and **460 days** to process on a single CPU core, it becomes clear that on the order of **3,000 CPU cores** are required to cope with **1.2 Petabytes** of data **per month**, acquired by a fleet of **20** streetside capture **vehicles**.

Solution:

The data volumes and processing requirements related with streetside data demand a significant investment in processing infrastructure, including both hardware and software. For this purpose, Microsoft and other companies operate a number of data processing facilities with clusters of thousands of CPU cores [330], using distributed processing models such as Google MapReduce [331, 332, 333].

4.3.9 Camera Calibration

Requirement:

Similar to aerial cameras the geometric and radiometric properties of a terrestrial MMS have to be calibrated within a specified accuracy. We require the residual errors for a lab calibration to within **+/- 2 μm RMS** (~0.5 pixel) in the image plane, which be further improved by means of self-calibration from data captured in the field.

Proposed Solution:

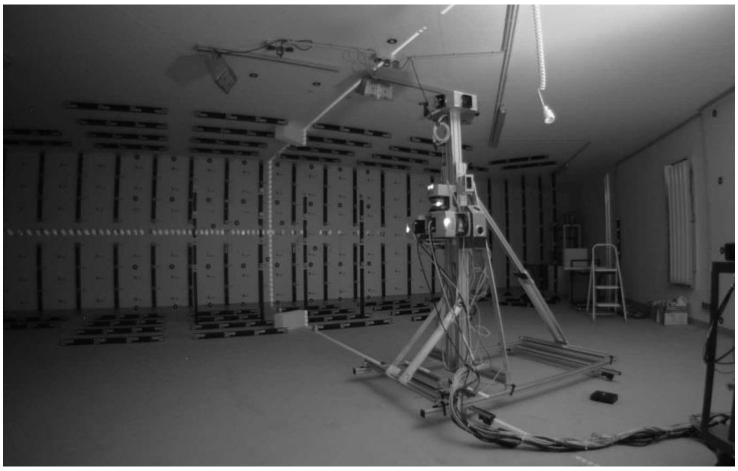


Figure 4.9 Infrared Laser Pattern Used for Determining Exterior Orientation of Laser Scanners

We use a similar method for calibrating camera geometry (extrinsic and intrinsic parameters) and radiometry as for the aerial camera systems (see 3.3.3). The geometric accuracy achieved per camera during the calibration for a variety of prototype and production systems was generally better than the specified limit of **+/- 2 μm RMS**.

In addition to the cameras, we also calibrate the geometric properties of the laser scanners. For this purpose we remove the infrared cut-off filters from part of the cameras, in order to capture the light pattern emitted by the laser scanners (see Figure 4.9). By combining the image measurements of the emitted laser beams with the corresponding depth measurements, we can determine rotation and translation parameters of the laser scanners relative to the cameras using bundle adjustment [334].

4.4 Overview of UltraCam-M Implementation

Similar to comparable aerial sensors, the UltraCam-M consists of several subsystems (see Figure 4.10), including a sensor unit (SM), a computing and data storage unit (CM / DM), several navigational components for sensor positioning (POS), as well as an interface panel for camera operation (IPM). We have previously described the general sensor concept in [33].

The computing unit (CM) contains 7 individual processing units for controlling the different sensors and transmitting the recorded data to the hard drives. The replaceable data unit (DM) holds 14 individual hard drives, each with a storage volume of 320 GB. The interface panel (IPM) provides a graphical user interface from which the system can be controlled by a single operator/driver. It also shows real-time quick view images and capture statistics. The positioning sub-system (POS) includes an integrated GPS/INS unit as well as a contact free odometer to measure the vehicle's velocity.

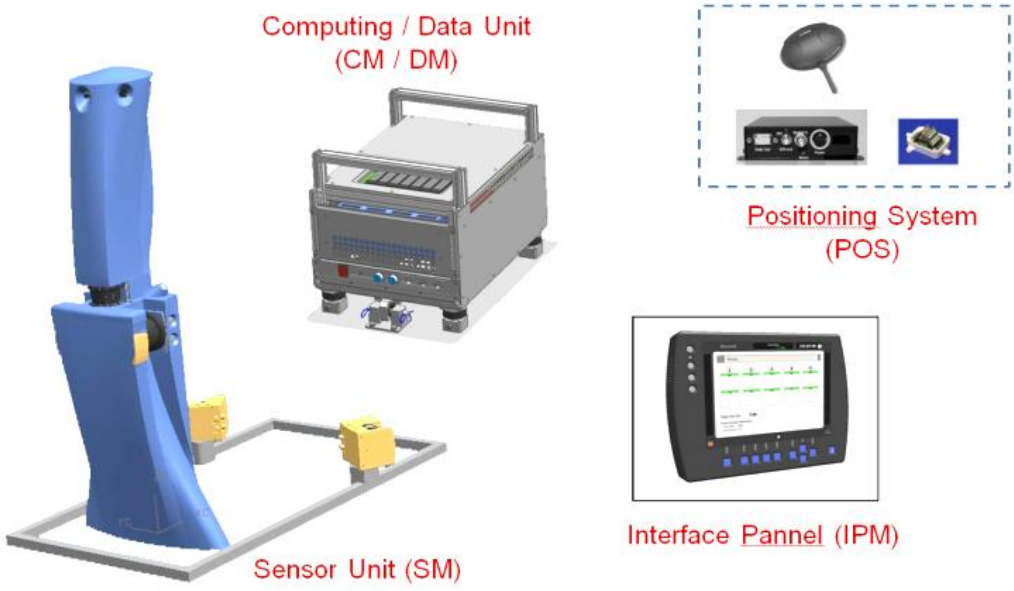


Figure 4.10 Overview of Components of UltraCam-M Mobile Mapping System

The sensor unit (Figure 4.11) consist of primarily the 12 image sensors, as well as 3 laser scanners (LMS) measuring the 3D geometry of the captured building façades.

A part of the image sensors (C0..C5) is arranged hexagonally, providing a 360 degree panoramic view around the vertical axis, with two cameras directly facing the expected direction of building façades (broadside cameras C1 & C4). Two sensors (C6 & C7) are placed at a vertical parallax of **0.85 m** to the former two, in order to support stereo reconstruction using a fixed baseline [24].

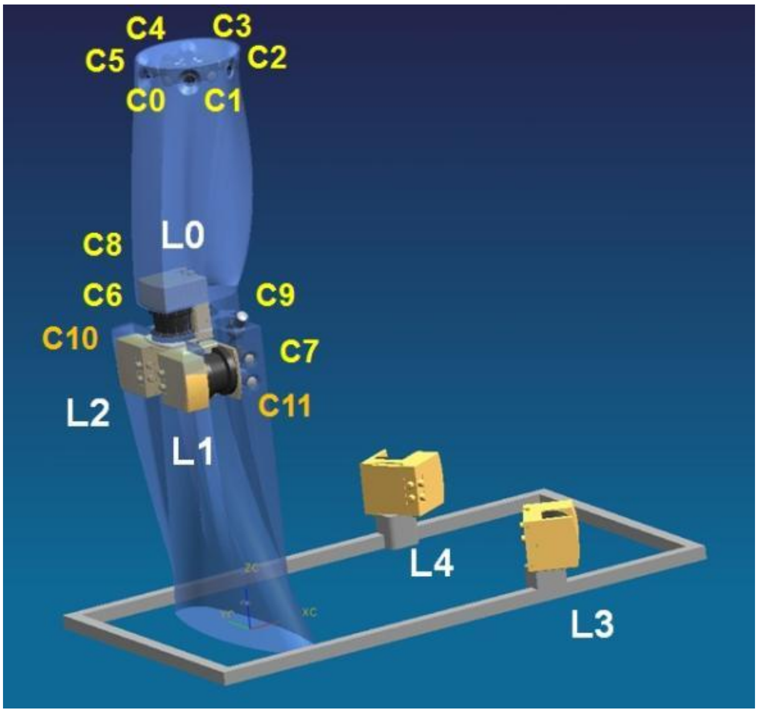


Figure 4.11 Arrangement of Cameras C0..C11 and Laser Scanners L0..L4 in UltraCam-M Mobile Mapping System

Two more sensors (C8 & C9) are pointed up at an angle of 50 degrees observing the top portion of building façades, and extending the 360 degree panorama coverage. While all previously mentioned cameras use color CCD sensors with Bayer-pattern in order to capture RGB color images, the final two cameras (C10 & C11) use monochrome CCDs equipped with near infrared (NIR) filters in order to support vegetation classification.

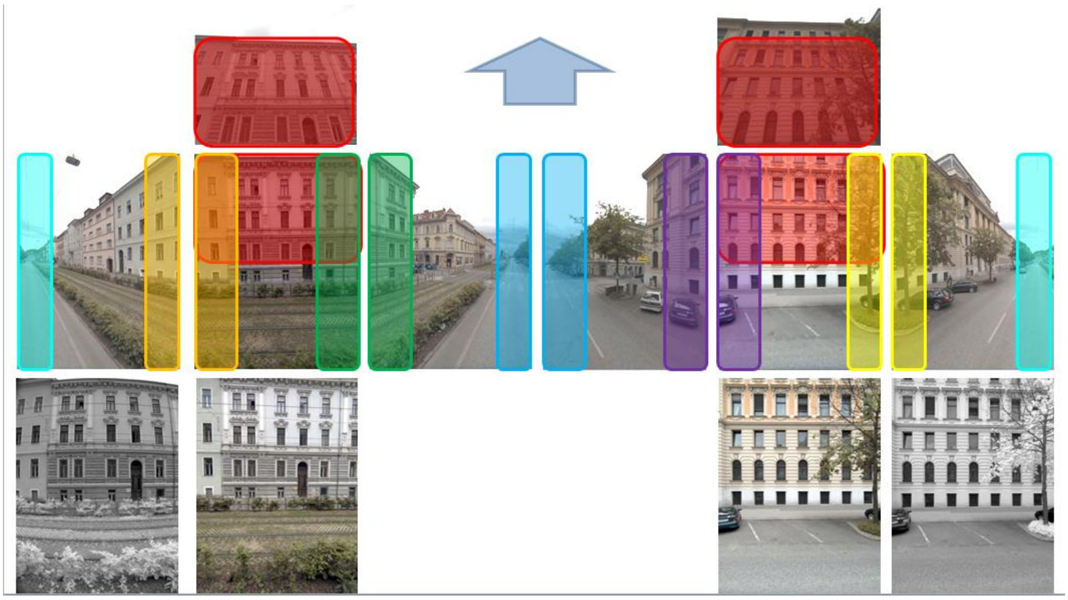


Figure 4.12 Overview of Images Recorded by All Cameras during One Trigger Event with Highlighted Overlap Areas

Each of the individual cameras is arranged in portrait orientation such that the angle of view is approximately 65° in horizontal direction and 90° in vertical direction. An overview of the images from all cameras for a certain trigger event is shown in Figure 4.12. The colored rectangles indicate overlap regions between adjacent images.

Figure 4.13 visualizes the number of views for each viewing direction by means of color coding onto a sphere. While the broadside direction is viewed by 3 cameras (2 visible + 1 NIR), all other directions, apart from the overlap regions, are only observed by a maximum of 1 camera. The overlap regions can also be recognized in this view. The combined field of view of the panorama rig is approximately **5 Megapixel** or **2 sr**, while each pixel frustum covers a solid angle of about **1.8 µsr**.

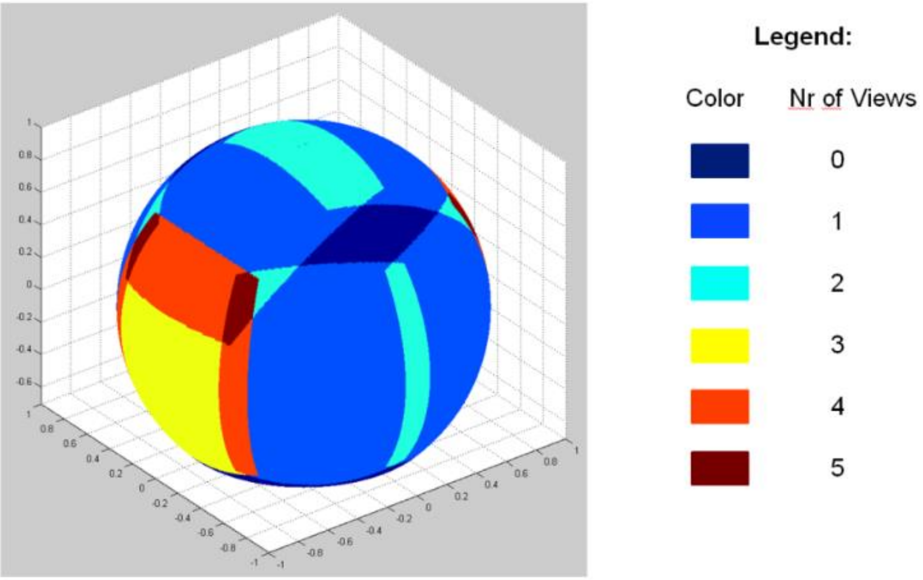


Figure 4.13 Illustration of the Number of Views for Each Viewing Direction at Infinity Projected Onto a Sphere; The panorama covers ~5 Million Pixels in Total @ 1.8 µsr Resolution;

In addition to cameras, the UltraCam-M system also includes a number of LMS which are used to directly measure the distance to objects along a 180° degree planar field of view by measuring the time of flight (TOF) for each scan angle. The configuration of the laser scanners in the UltraCam-M system is also laid out in Figure 4.11. One laser scanner (L0) is arranged horizontally and faces forward. Its purpose is to provide a horizontal depth profile along the scene at a certain height. Measurements from this LMS can be used during data capture to estimate distances to the left and right building façades, as part of the trigger control mechanism proposed in [37] (see Section 4.3.3). Two vertically arranged laser scanners (L1, S2) facing orthogonal to the driving direction (broadside), allow the scanning of depth profiles of building façades and other objects due to the forward motion of the vehicle. The resulting point cloud is visualized in Figure 4.8, superimposed onto an image recorded by one of the broadside looking cameras. Two more backward facing laser scanners (L3, L4) can be added optionally, with the aim of scanning the building façades even in case of occlusions. Occlusions of the broadside LMS can happened due to people, vegetation, parking cars or other objects.

Figure 4.14 shows the final version of the UltraCam-M system (version 1) installed on a vehicle used for data capture in front of the Microsoft office in Graz, Austria.

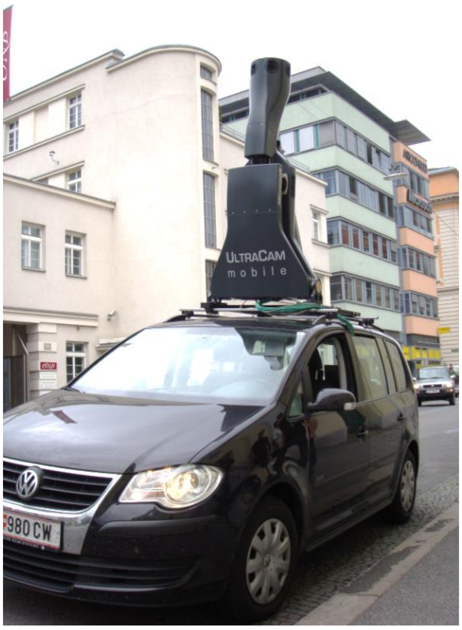


Figure 4.14 UltraCam-M Mobile Mapping System v1 Installed on Capture Vehicle

4.5 Privacy Protection for Streetside Imagery

Modern mapping services such as Bing Maps provide 360° images with panoramic views of streets in various cities throughout the world. These mapping applications display photos captured with multiple directional cameras mounted on a vehicle traveling along various streets. The vehicle uses GPS/INS units for positioning, LiDAR for measuring building geometries, and other components. The mapping service applications are provided to users with remote computing devices so they can experience a visually immersive perspective of streets. In effect, image capturing technologies provide users with the view of streets with buildings, streets, signs, object details etc. at a high level of detail.

A flipside of the increased image resolution is that the raw images initially captured by cameras may include private objects such as people or car license plates. This fact raises privacy concerns both by the affected people, as well as by government and nonprofit agencies across the world. While one may argue that people and objects if they are visible in a public place such as a street, are not really in a public environment, local regulations in many places prohibit photography and publication of such photography without prior consent of the affected person or owner of an affected object. In most locations, this concerns mostly people's faces and license plates of cars. In other locations such as Germany, local privacy laws also prohibit publication of house numbers visible on buildings. Additionally, people have to be enabled to flag remaining private objects for later removal by the map provider [165].

Altering the captured images to remove private objects is usually required. A common approach to privacy protection is to apply an algorithm that automatically detects private objects captured in each raw image. Identified private objects are then processed to remove or blur the private objects before publically releasing the images [28]. This problem is especially challenging as humans generally are very skilled at recognizing people and people's faces even under large variations, which means that automatic detection methods must be very well tuned in order to achieve sufficient detection of the majority of private objects.

Due to the sensitive nature of privacy, and the potential legal consequences of not complying with privacy rules and regulations in different countries, achieving a high detection rate close to **100%** is an important goal when developing methods for privacy protection. On the other hand, minimizing the damage to the non-private part of the recorded imagery incurred by false positives to below **2-3%** of the image area, is another important requirement.

Various algorithms have been developed to automatically detect faces and people in images, which seems relevant in the context of privacy protection for streetside imagery. Hence we decided to use a state of the art face detection method [335] proposed by Viola and Jones [294, 336, 337] which is based on Haar-like features [338] for initial testing. The implementation used was by Cha Zhang and Gang Hua at Microsoft Research, Nevertheless, experiments showed that although frontal faces that were clearly visible at a high enough resolution could be detected reliably, this was not true for the majority of faces in typical streetside imagery, due to significant differences in the appearance of faces compared to the training data used for such methods.

The main challenges are caused by the low resolutions of faces located at distances more than a few meters from the camera and large variations in the facial pose due to the raised camera position on the capture vehicle. Additional challenges arise from image noise due to exposure variations. Although face detection methods allow setting a sensitivity threshold to increase the detection rate, this also increases the false positive rate (FPR) significantly, while still not providing sufficient recall performance. That is, many non-private objects are identified as private objects, while there is still a large percentage (>50%) of private objects that cannot be detected successfully. Figure 4.15 features typical examples of faces and license plates as they are used for training and evaluation of existing detectors, as well as typical streetside examples of such objects.

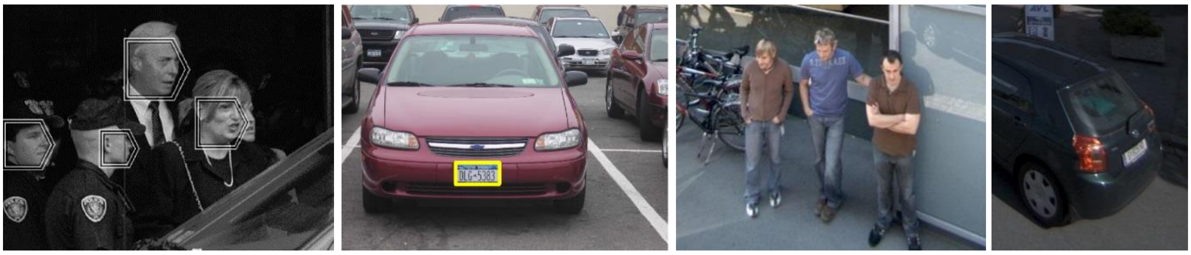


Figure 4.15 Typical Image Scales and Poses used for Evaluating "Traditional" Face and License Plate Detection Algorithms (Left) Compared to Streetside Example Faces and License Plates (Right)

The same tradeoff with respect to recall/precision performance applies to automatic detection methods for other kinds of private objects, such as license plates. Many of the existing methods for license plate detection are tuned for application such as automatic toll collection, where large image scales (50-100 pixel font height) with little geometric distortion are common [339, 340, 341]. They also report detection rates of 70-80% under such favorable conditions. As license plates in streetside images occur in much smaller scales and are affected by strong distortions most existing methods are unlikely to work reliably enough for privacy protection purposes.

We therefore propose a novel privacy protection workflow, which has been used in the initial release of streetside imagery by Bing Maps. It aims at achieving both a high recall rate in order to comply with legal requirements, as well as a low FPR. This is achieved by using a combination of detectors for identifying people and license plates, which by themselves would generate many false positives, and a second set of methods to identify areas in the images that are unlikely to contain private objects. Parts of this workflow have been presented in [39].

The detection algorithms include an implementation of Viola-Jones face detection [294], a set of low level features to detect areas of high contrast such as horizontal and vertical edges, and a specifically developed license plate detector, which was optimized for streetside imagery. The methods used for eliminating false positives and reducing their visual damage include a planar region detector, identifying building façades by comparing consecutive image events, hue based methods for segmenting skin tone and vegetation regions in images, height thresholding, as well as depth dependent adaptive image blurring. The individual methods are described in the following sections of the document (4.5.1 through 4.5.7). An illustration of their combination as well as a report of our experimentation results is provided in sections 4.5.8 and 4.5.9 respectively.

4.5.1 Face Detection

Detecting faces in images is a common problem in many different areas, such as consumer photography, surveillance applications, video conferencing or digital photography [335]. This presents us with an opportunity to use the same methods developed for face detection such as [294] for detecting people's faces in streetside imagery for privacy protection. However, most of the standard applications of face detection differ significantly in their requirements regarding the variations of faces to be recognized as well as the required recall rates. For example existing detection methods often are tuned for faces viewing the camera captured at dimensions of 50-100 pixels minimum.

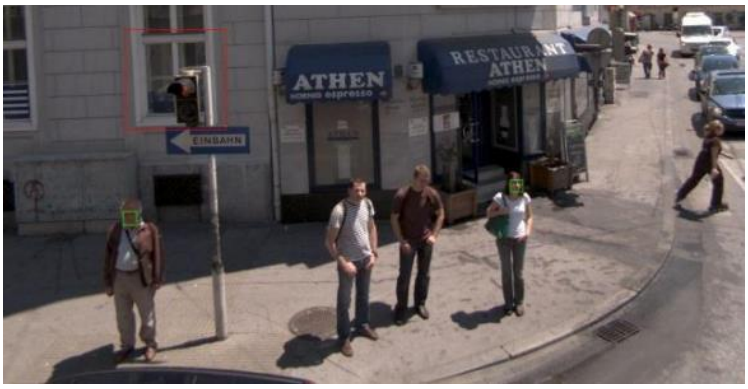


Figure 4.16 Example of Viola Jones Face Detection Applied on Streetside Image; Green Rectangles are True Positive Detections, Red Rectangles are False Positives;

However, streetside images often contain faces which are photographed from the top down, the side or even partly from the back. Therefore, additional challenges are posed on such methods due to large variations in the facial appearance, the pose, or the scale of the facial region in the image, which may cover only 10-20 pixels As a consequence we found that the recall performance of standard algorithms to be significantly worse (e.g. < 50%) for streetside imagery than for other applications. Figure 4.16 shows an example streetside image in which 2 out of 5 faces were detected correctly, while a window caused an incorrect detection.

In order to improve the recall of the Viola-Jones face detector, several options exist. For example, the sensitivity of the algorithm can be increased, such that more faces are detected correctly, at the cost of an increased FPR. Since false positives frequently occur at unrealistic scales, such as around a window of a building façade (See Figure 4.16), geometric reasoning based on the available depth information can be used to filter out detections at clearly incorrect scales (e.g. at scale ratios >2). For example the window in the above figure can automatically be removed from the detection. However this method depends on reliable depth information. The depth information obtained by a vertically mounted 1D laser scanner (see 4.3.4) becomes unreliable in case of moving people or objects, as illustrated below in Section 4.5.5. Hence the depth map representation of a person can be offset by arbitrary amounts from their image location. In case the thresholds for the region sizes are set too tightly, false negatives can occur, while setting the thresholds too loosely can increase the false positives beyond acceptable limits.

Another way of improving the recall performance is by training the respective face detection algorithm specifically for the typical appearance of faces as they occur in streetside imagery, including profile views, low resolution images, top down views etc. We attempted to retrain the face detection algorithm used for this specific purpose, which led to a noticeable improvement in the detection rate at a given FPR (by 10%), but was still not sufficient in many cases, such as due to profile views at small scale, people facing down or partly away from the camera.

4.5.2 Edge Based Saliency Detection

Both images of people's faces as well as license plates typically contain features in a specific frequency band, such as a person's mouth, nose and eyes in case of a face, or digits on license plate. As the algorithms specifically aimed at face detection often fail at achieving satisfactory recall rates for streetside privacy protection, we decided to use a simple, edge-based saliency detection algorithm tuned for high recall in this frequency band in addition to a dedicated face detection algorithm tuned for a low recall rate.

In particular we first compute magnitudes of the horizontal and vertical derivatives of an input image (See Figure 4.17 a). We use filer kernel [-1 0 1] and its transpose to compute the derivatives (b, c). Hence we apply a **9·9** pixel median filter on the derivatives to remove individual outlier edges (d, e). Finally we compute the output detection mask by means of thresholding the median images and combining them via a logical AND (f), followed by a morphological closing operation (g). The threshold value and the filter kernel size were determined heuristically based on a set of 40,000 streetside images over a variety of lighting conditions and geographies.

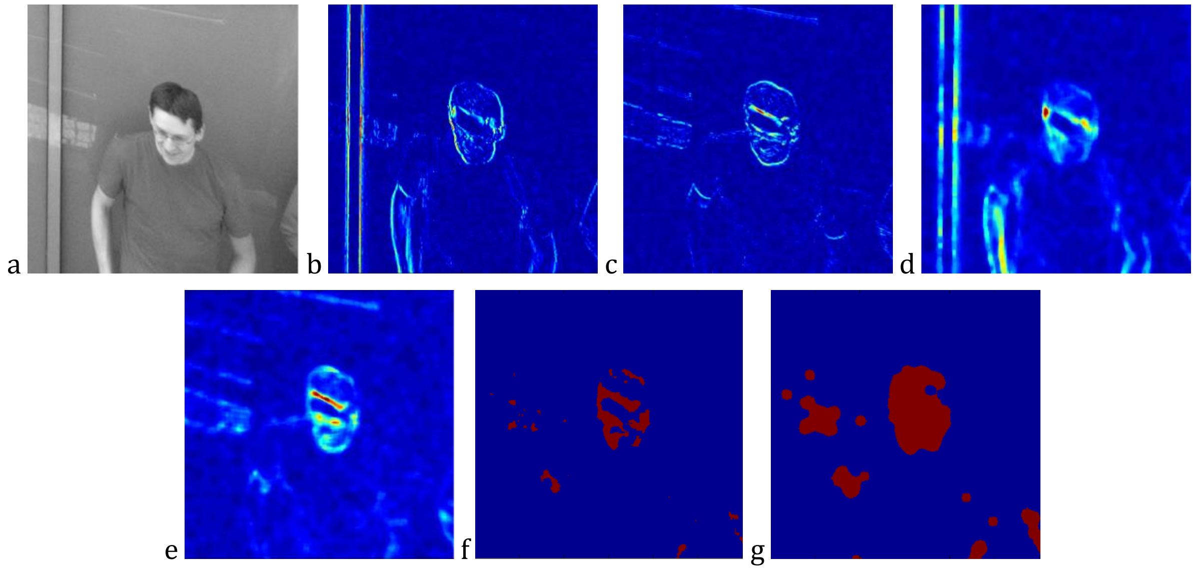


Figure 4.17 Visualization of the Different Steps during Edge Based Saliency Detection: a) Input Image; b) |Horizontal Derivative|; c) |Vertical Derivative|; d) Horizontal Median Filter Output; e) Vertical Median Filter Output; f) Binary Mask after Thresholding; g) Output Mask after Morphological Dilation;

Although this method is prone to finding many false positives in up to 10% of the other image areas, it is very reliable in finding close to 100% of all facial regions. The high FPR can be reduced to an acceptable range by combination with alternative methods. For example, detections can be limited to a certain height above ground (e.g. 2.2 m) where people and cars can mostly be expected. Additionally pixel based skin tone or vegetation detection, or filtering of detections on planar surfaces such as walls or the street surface can be used to further reduce false positives.

4.5.3 License Plate Detection

License plate detection for privacy protection in streetside images presents challenges similar to face detection due to the large appearance variations of license plates. Differences in scale and resolution, the viewing angle as well as the license plate design itself are evident (See Figure 4.18).

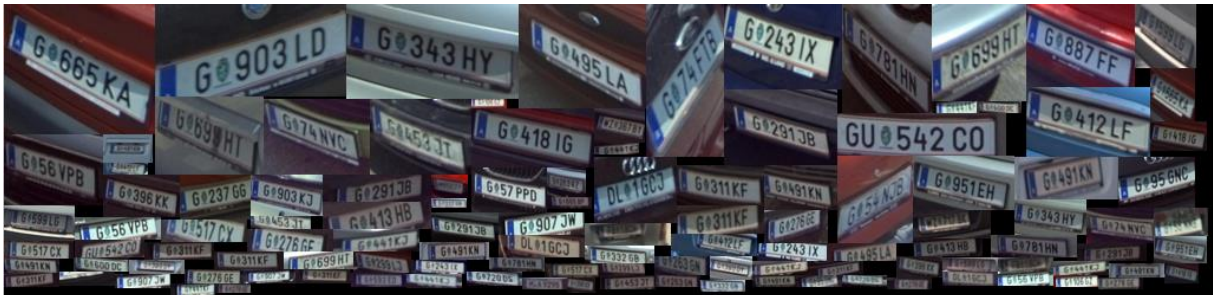


Figure 4.18 Samples of License Plates in Streetside Images

Therefore many of the standard methods used for license plate detection, which are designed for applications with more control such as surveillance cameras, road tolling, etc. achieve insufficient recall rates for streetside images captured for mapping. With the goal to achieve a sufficient recall performance, we therefore decided to use a combination of low level operations such as local sliding window filters or morphological operations for license plate detection. They are tuned for finding horizontally elongated regions of letters, at a variety of different viewing angles and scales. The following filters F_{W2}, F_{W4}, and F_D are used on the input image by means of a sliding windows operation (convolution), leading to outputs r_{W2}, r_{W4}, and r_D. For a particular input image (a) Figure 4.19 shows the intermediate results of the detection algorithm.

$$F_{W2} = \begin{bmatrix} 1 & 1 & -2 & -2 & 1 & 1 \\ 1 & 1 & -2 & -2 & 1 & 1 \\ 1 & 1 & -2 & -2 & 1 & 1 \end{bmatrix}; \quad F_{W4} = \begin{bmatrix} 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \end{bmatrix}; \qquad (4.1)$$

$$F_D = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \qquad (4.2)$$

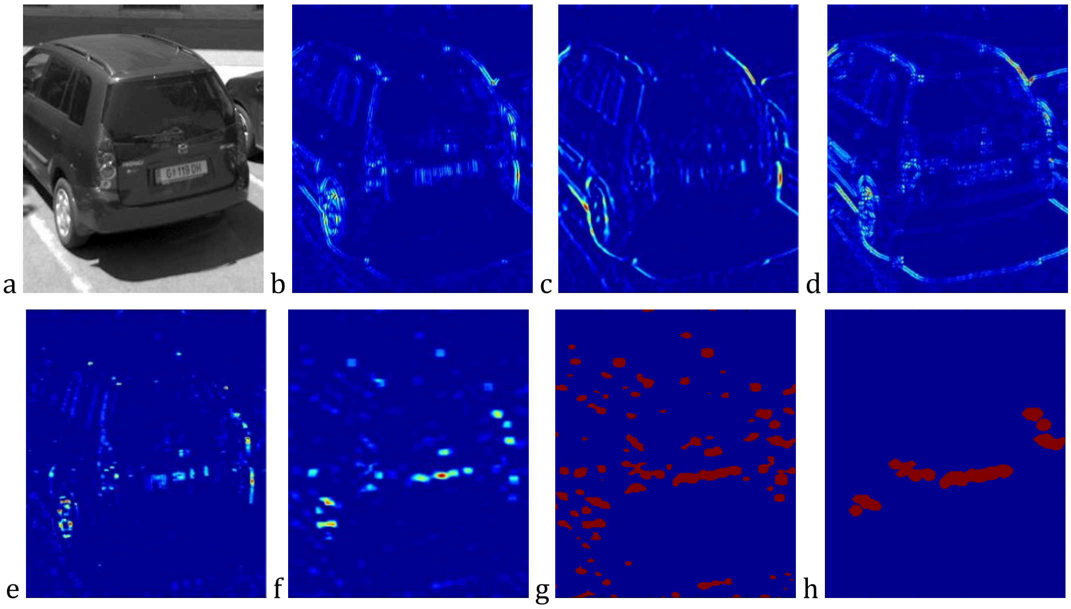


Figure 4.19 Visualization of the Different Steps during License Plate Detection: a) Input Image; b) Filter Output |rW2|; c) Filter Output|rW4|; e) Filter Output |rD|; f) Result of Equation 4.3; g) Binary Mask; h) Result after Applying Morphological Operations and Connected Components;

Filter F_{W2} is designed to generate high responses r_{W2} around vertical lines (letters on a license plate) with a width of 2 pixels (b). Filter F_{W4} also responds to vertical lines, with a width of 4 pixels and Filter F_D responds to diagonal lines of width 2 (c and d). From the different filter responses, we compute a detection score

$$R = 2 * |r_{W2}| - |r_{W4}| - 4 * |r_D| \qquad (4.3)$$

which favors vertical lines of width 2 but punishes vertical lines of width 4 and diagonal lines (e). This results from the assumption that letters on license plates tend to have a certain stroke width, and are arranged roughly vertically. Next we apply the two filters

$$F_V = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}^T; \qquad (4.4)$$

$$\text{and} \quad F_H = [1 \quad 1 \quad 1 \quad 1 \quad 1];$$

consecutively on the score R in order to first locate horizontal structures with an expected height of a license plate (10 pixels) and then connect them using horizontal smoothing (f).

The resulting score *S* of this filtering step is converted into a binary mask using thresholding (g). Finally we apply morphological opening and connected components on the mask in order to connect individual letters, and to remove components that do not have the expected (horizontally elongated) shape of a license plate. The result of this operation is the output license plate mask for the given scale (h). The same operation is repeated for several scales in order to find license places at different viewing distances.

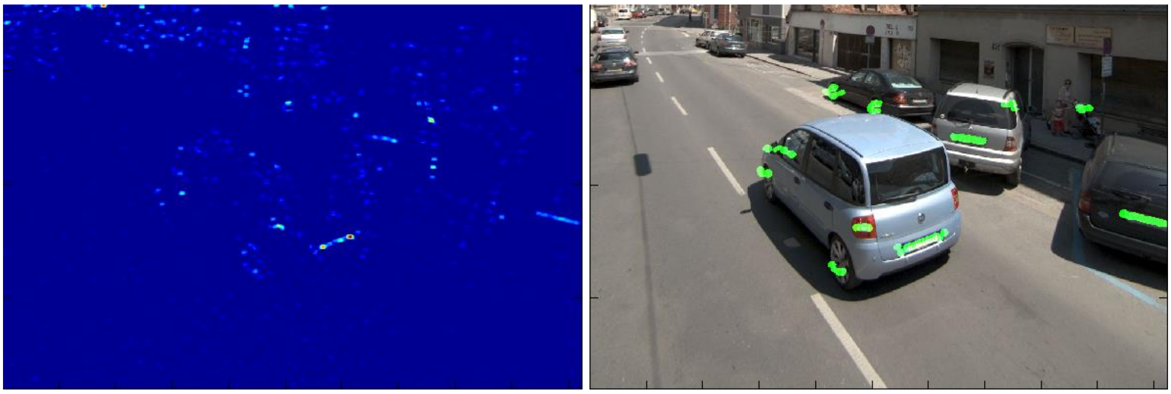


Figure 4.20 Sample Response S from License Plate Filter (Left); Input Image with Resulting Detection Masks from License Plate Detector (Right)

For a larger sample image, the filter output *S* as well as the resulting license plate mask are shown in Figure 4.20. The detection mask has been superimposed on the input streetside image. The three identifiable license plates in the image have been correctly detected, besides some amount of false positives. This example supports the expectation that false positives are an acceptable phenomenon with hardly any effect on the application.

4.5.4 Color based Segmentation

Since the various detectors for faces and license plates described in 4.5.1 through 4.5.3 have been designed or configured specifically to achieve high recall performance as required for privacy protection, they also generate a significant amount of false positive detections in other, non-private regions of the images. With the intention to limit the effect of false positives on the quality of the processed images, we remove detections in areas that classified as vegetation, or as non-skin in case of faces and other body parts.

As a preprocessing step for color based segmentation, the images first have to be color balanced. A simple way of doing so is by using presumably neutral gray parts of the images such as the street surface as a reference, and scaling the red and blue color channel linearly to align their intensity histograms to the green channel.

Skin Segmentation

For detecting skin regions in these color balanced images, we pursue a pixel based method involving a Bayes classifier [342, 343]. The method uses color histograms of skin and non-skin color values *c* in a particular color space as a means to compute the conditional probabilities *P(c|skin)* and *P(c|¬skin)*, as well as the non-conditional probabilities *P(skin)* and *P(¬skin)* as explained in [343]. Using Bayes' Theorem, we can hence compute the conditional probability that a color value c belongs to a skin region.

$$P(skin|c) = \frac{P(c|skin) * P(skin)}{P(c|skin) * P(skin) + P(c|\neg skin) * P(\neg skin)} \quad (4.5)$$

We use this probability as the score for skin classification, and compare it against a threshold *Θ* to decide whether a given input pixel color *c* likely belongs to the skin class.

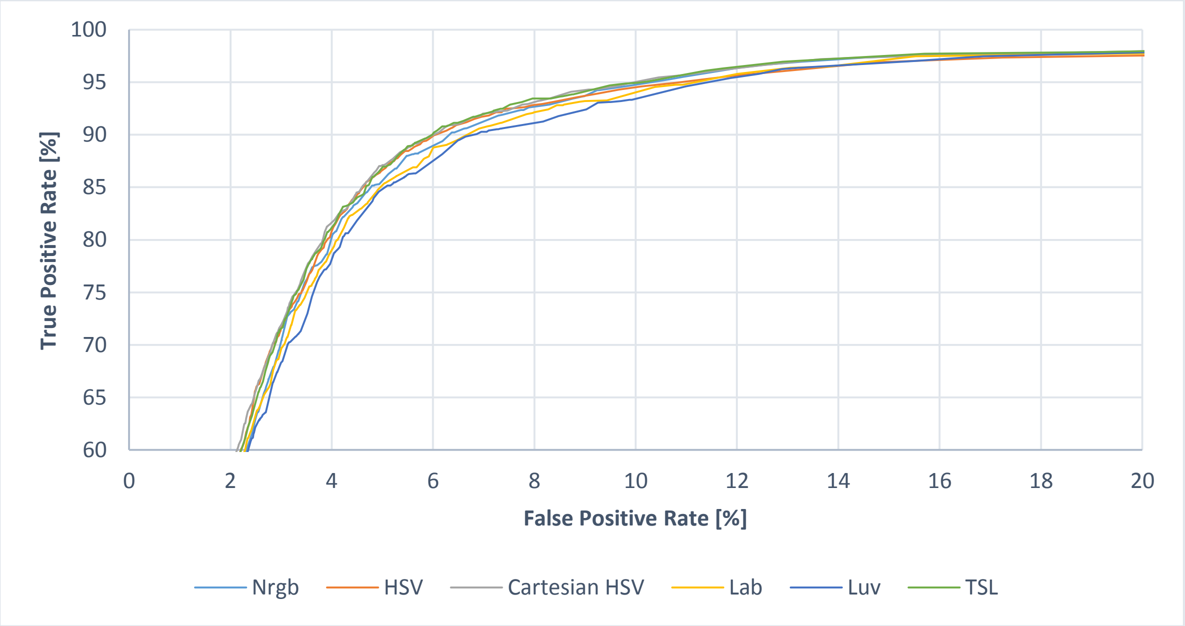


Figure 4.21 ROC Curves for Color-Histogram Based Skin Classification using Bayes Classifier – View is Zoomed in at Range 0..20% FPR and 60..100% TPR

For the training we selected a subset (50%) of color values from a labeled ground truth dataset of **3,500** skin pixels, as well as **300,000** non-skin pixels in order to compute color histograms for both classes in a particular color space. Using the remaining data samples, we evaluated a variety of color spaces suggested in [343] for the histogram computation and classification to determine which one is most discriminative for the two classes: Normalized RGB (Nrgb), Hue Saturation Value (HSV), Cartesian HSV, Lab, Luv and Tint Saturation Luminance (TSL);

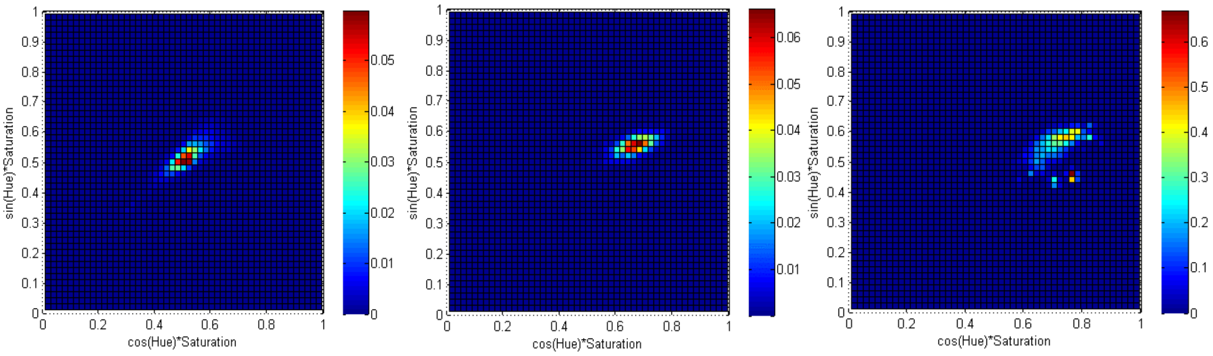


Figure 4.22 Conditional Probabilities: P(c|¬skin) - Left; P(c|skin) - Center; P(skin|c) - Right

The resulting ROC curves [52] in Figure 4.21 show that while there is generally little difference between the color spaces, Cartesian HSV as well as TSV achieve slightly better true positive rates for a given FPR, than the other color spaces. In consequence we decided to use the Cartesian HSV color space for the classification. The different conditional probabilities for this color space are visualized in Figure 4.22. Note that the non-skin cluster is offset to the bottom-left relative to the skin cluster, although there is some amount of overlap. The skin classification score (right) is maximal around the location of the skin-cluster.

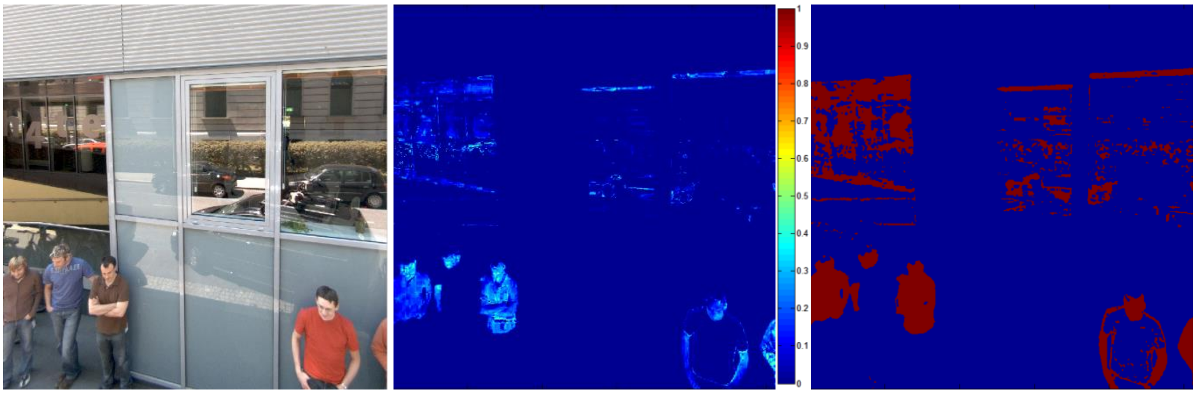


Figure 4.23 Skin Classification Sample Image (Left); Skin Probability (Center); Skin Mask (Right);

For a given input image (left), Figure 4.23 shows the conditional probabilities $P(skin|c)$ for individual pixel colors (center), as well as the resulting binary skin mask (right).

Vegetation Segmentation

Plants such as bushes and trees generally contain features (e.g. leaves) of a similar spatial frequency as facial features which would lead to many false positives of the edge based saliency detector (4.5.2). Therefore we want to specifically detect vegetation by means of per-pixel color segmentation. For this purpose we use boundaries in HSV color space for classification similar to [344]. These boundaries are defined as thresholds on the values of Hue, Saturation and Value (See Table 4-1) which were determined heuristically from sample images containing vegetation.

h_{min1}	h_{max1}	h_{min2}	h_{max1}	s_{min}	s_{max}	v_{min}	v_{max}
0.1	0.2	0.51	0.57	0.1	0.85	0.1	0.75

Table 4-1 Thresholds used for Pixel Based Vegetation Detection

An individual pixel is counted as vegetation if its color values (h,s,v) fulfill the condition

$$Cv = [(h_{min1} \leq h \leq h_{max1}) \cup (h_{min2} \leq h \leq h_{max2})] \cap (s_{min} \leq s \leq s_{max}) \cap (v_{min} \leq v \leq v_{max}) \qquad \textbf{(4.6)}$$

Figure 4.24 demonstrates the individual steps of the method based on a sample image. The binary mask resulting from the classification is filtered by morphological opening to remove noise. Then a density image of filtered binary mask (bottom left) is computed as the product of two convolution results using two different Gaussian kernels (σ_2=10 and σ_2=30 Pixels). The density image indicates whether an area of the image has a high density of pixels classified as vegetation.

The resulting vegetation mask (bottom right) is computed from the density image via thresholding (d_{min} = 0.04) and using connected components to remove regions smaller than a certain size (1000 pixel).

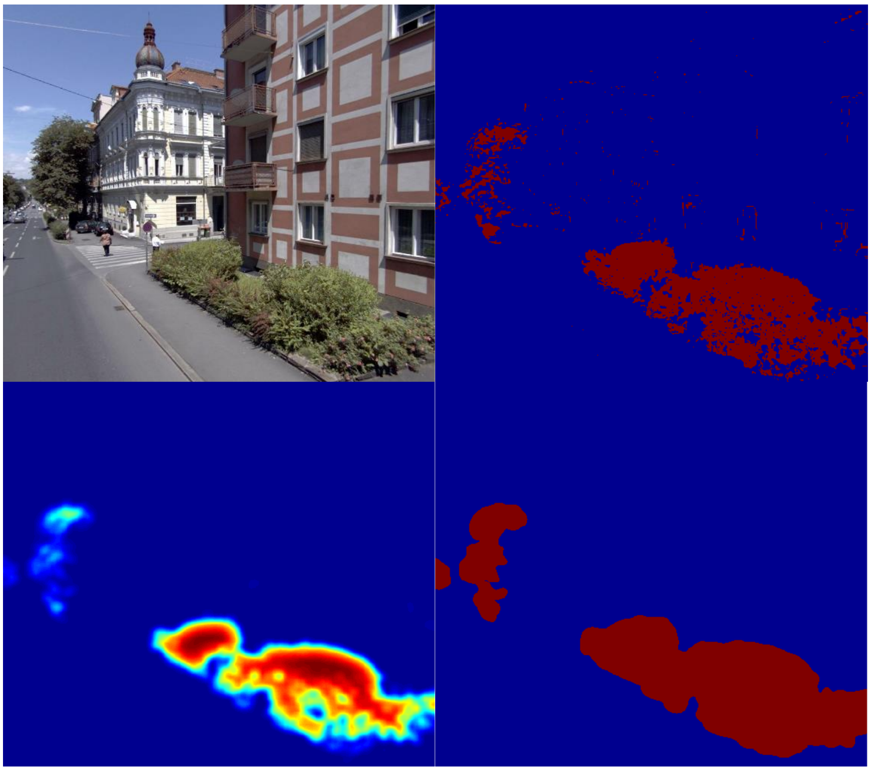


Figure 4.24 Vegetation Segmentation Example: Sample Image (Top Left); Raw Mask after Thresholding (Top Right); Density Image (Bottom Left); Final Vegetation Mask (Bottom Right);

4.5.5 Depth Map Creation

Several methods (4.5.6, 4.5.7) used for the described privacy protection workflow, require a per-pixel depth map image for individual camera views.

For this purpose we use the georeferenced laser point cloud such as the example shown in Figure 4.25 which has been computed by projecting the LiDAR based depth measurements relative to the motion trajectory captured by the positioning system. Note that red points in the 3D point cloud are outside of the cameras field of view, while green points are within. The blue lines indicate the camera frustum for the particular image event. The 3D points are hence projected into image coordinates (x/y) using the respective camera projection matrix P.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = P * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}; \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix} \tag{4.7}$$

Figure 4.25 shows the projected points which have been color coded based on the depth from the camera (center). A z-buffer is created at the same resolution as the camera image, and filled with default depth values (e.g. 100 m). For every projected point, the z-buffer image at location (x/y) is updated to the depth value of the respective point, only if this depth value is lower than the previous depth value stored at the same location. Eventually, missing values in the z-buffer are filled by means of morphological greyscale opening [345, 238] for small gaps between individual samples, as well as bicubic interpolation for larger gaps. We use a disk-shaped structuring (r=8) for the opening operation. As described in [346] greyscale opening removes bright features in images smaller than the structuring element, which means that in case of a z-buffer, closer points will dominate farther ones in their vicinity.

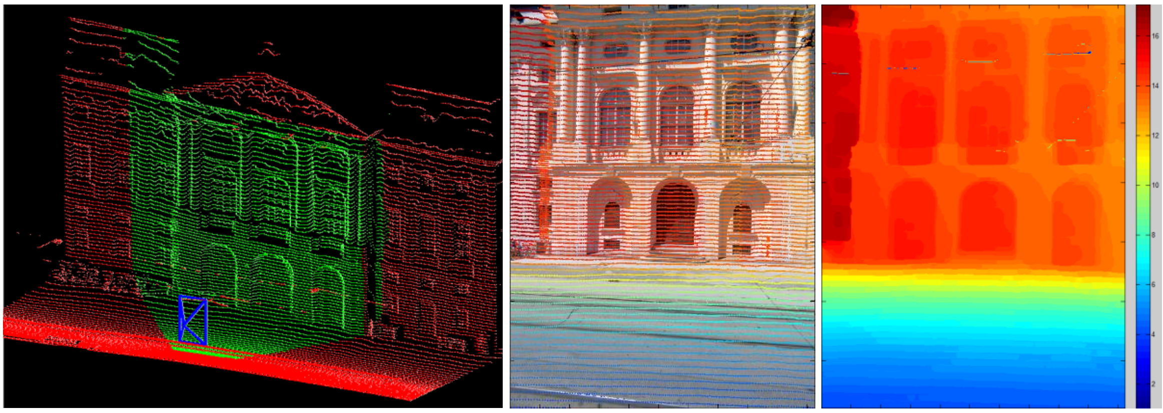


Figure 4.25 Example Depth Map Workflow (Opera House Graz, Austria): 3D LiDAR Point Cloud with Camera Frustum and Visible Points Highlighted in Green (Left); Image with Projected Points (Center); Interpolated Z-Buffer Image (Right);

Although LiDAR provides a computational advantage over stereo reconstruction of a scene, it also has several drawbacks.

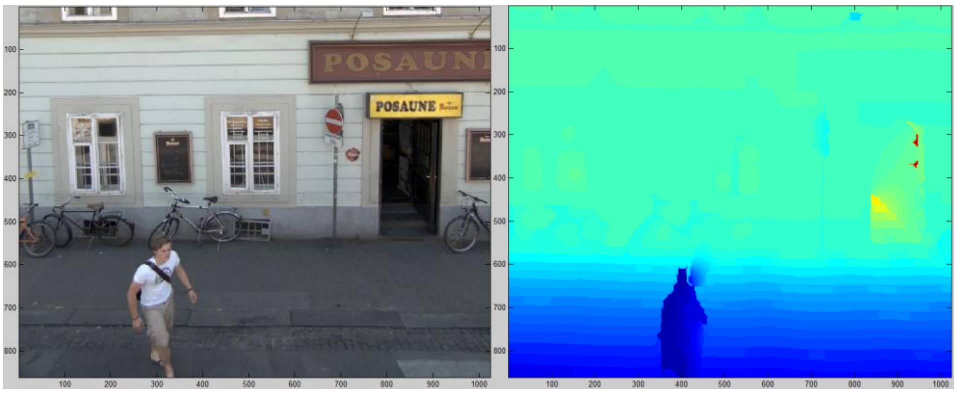


Figure 4.26 Streetside Image with Moving Person and Corresponding Depth Image; Note the Discrepancy Caused by the Motion of the Person between the LiDAR and Image Capture

For example, it comprises capturing events that must occur simultaneously with image capturing. Thus 3D data become unreliable in the presence of moving objects, as visualized in Figure 4.26. The location of the moving person in the depth map (right) is offset relative to the corresponding image (left), due to the time offset between the two captures

4.5.6 Detection of Objects on Major Planes

As an additional measure to remove false detections, we use a method to identify major planar surfaces in urban scenes, which was previously described in [26]. Images of urban environments typically include several planar surfaces occupying portions of the images. These planes may include building façades, streets, sidewalks, and the like. The identified planes are used to create a mask that prevents those areas to be included in subsequent private processing.

The algorithm identifies planar surfaces by comparing the gray values of multiple exposures of the same scene. Figure 4.27 is an example of a triplet of images recorded by a streetside capturing system, with indices N-1, N and N+1 (from left to right), where a camera has transitioned along a street between each of the images. The three images overlap, and include both static objects belonging to major planes (street, façades) as well as dynamic objects (car, people, signs, and the like) not lying on the major planes. Although this example uses three images, the present solution is not limited to three because any number of images can be utilized to generate similar results.

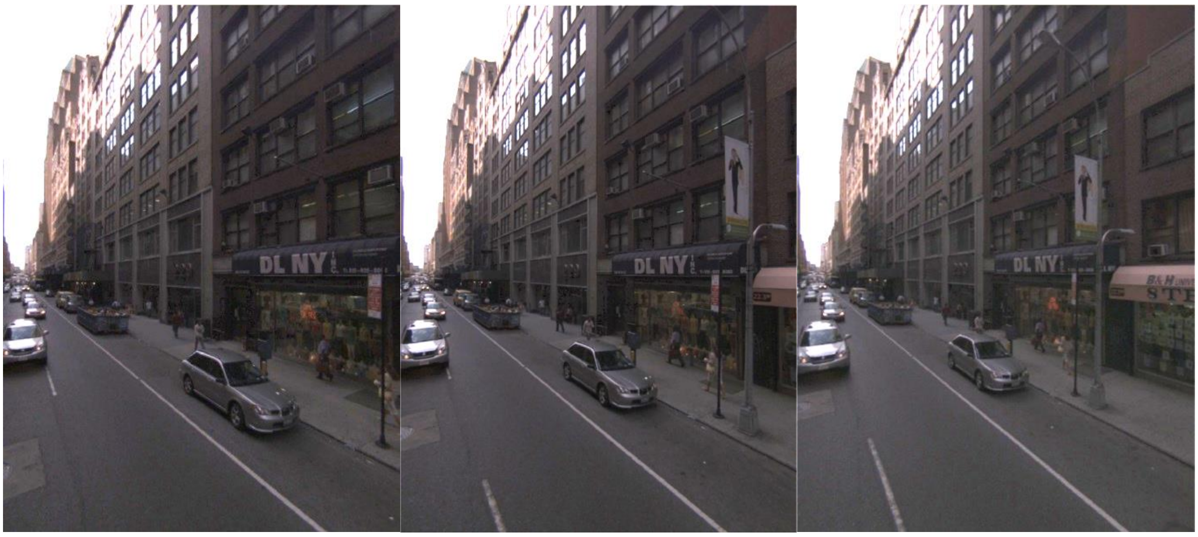


Figure 4.27 Samples of Consecutive Streetside Image Trigger Events N-1, N, N+1

In order to determine the geometric relationship between identical scene points in the multiple views, we use a depth representation such as a depth map image, which has been generated using the method described in Section 4.5.5. However, rather than including all of the LiDAR measurements obtained, we only use those points located on major planes to generate the depth map, excluding points on other structures.

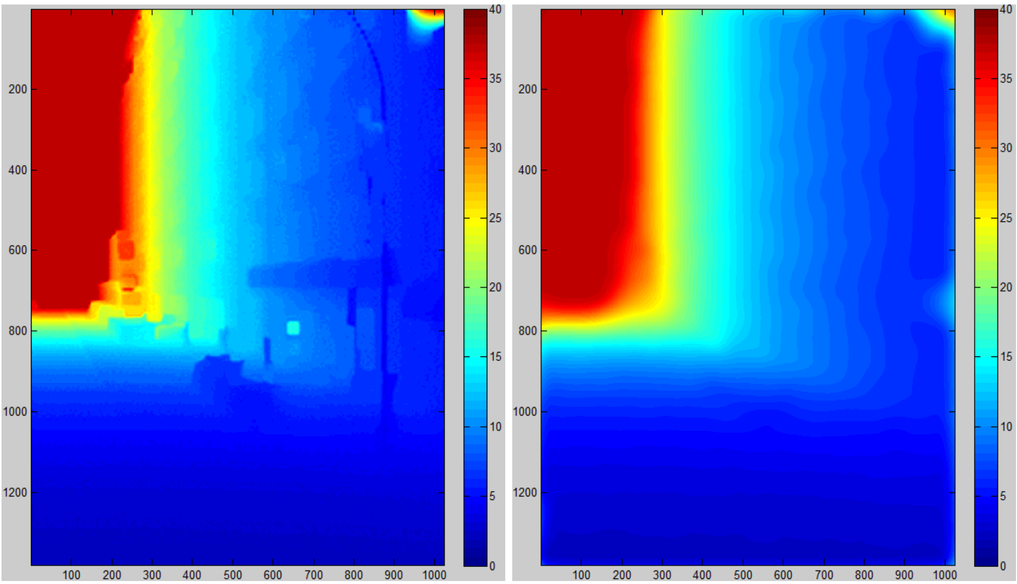


Figure 4.28 Depth Map Image for Trigger Event N (Left); Depth Map Image Containing Only Planar Surfaces (Right);

Major planes may be extracted from the 3D LiDAR data through different methods, such as "RANdom SAmple Consensus," (RANSAC) fitting of planes to the point cloud [45]. RANSAC is an iterative method utilized to robustly estimate parameters of a mathematical model from a set of observed data containing outliers. Figure 4.28 shows the original depth map image for event *N* including all scene points (left) as well as the depth map image computed only from points on major planes (within 0.2 m tolerance). For each of the three images in Figure 4.27, geometric information of the scene has been collected by the LiDAR scanner included with the streetside capturing system.

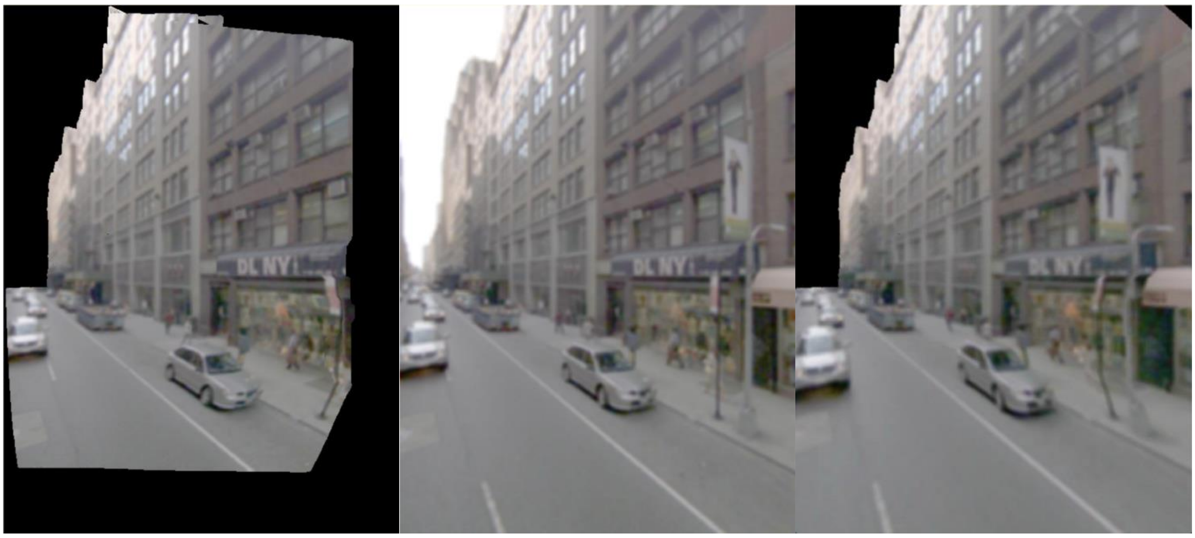


Figure 4.29 Reprojection of Blurred Image N-1 (Left); Blurred Image N (Center); Reprojection of Blurred Image N+1 (Right)

By using this depth map, the pixels from one image are mapped to another (re-projected) in accordance with the scene geometry (depth map) using a method similar to [347]. Based on the depth dependent transformation for each pixel location, image N-1 can be re-projected (warped) onto image N. The same can be done for image N+1. The result of such re-projections is displayed in Figure 4.29, compared to the original image N. Note that all three images have been blurred using a Gaussian filter [238] prior to re-projection to smooth out noise and small variances of grey values close to edges. Specifically, Figure 4.29 includes a blurred version of image N in the center, as well as images N-1 on the left and N+1 on the right, warped onto image N using depth information from major surfaces.

After the re-projection, grey values from the re-projected image N-1 can be compared to grey values of the corresponding pixel locations of image N. For each pixel location, a grey value difference can be computed as the Euclidean distance in RGB space between two RGB vectors. Wherever a pixel location contains an object that is part of one of the major surfaces (street, façades), the re-projection error is low, and hence the two grey values should be nearly the same. If a pixel location contains an object that is either not on one of the major surfaces or has moved between the two image events or both, the grey values for the same location in the two images may significantly differ from each other (depending on an object and background). By using a threshold, a binary decision can be made for each pixel to make a distinction between portions that are surfaces and outliers, which leads to the creation of a binary mask for the whole image.

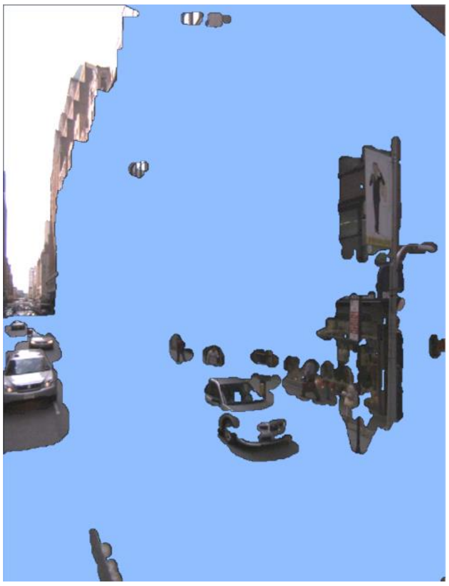


Figure 4.30 Major Plane Detection Mask

The same computation of a grey value difference and use of a threshold may be performed for image N+1 as well, resulting in a second binary mask. Combining the two binary masks by using Boolean AND generates a final outlier detection mask. An example of such a mask is shown in Figure 4.30, superimposed onto the original image, where portions corresponding to a logical "FALSE" are shaded solid to indicate static objects on major planes, and locations corresponding to logical "TRUE" are all transparent portions. The present private object detection workflow uses this mask to identify private objects in portions of the image that are designated as outliers. The various detection algorithms detailed in 4.5.1 through 4.5.3 may hence be tuned to a high sensitivity (>95%) for detecting private objects while generating low FPR (<2%) because portions of the image designated as surfaces are excluded from detection.

4.5.7 Depth Dependent Adaptive Blurring

The final step of the presented privacy protection method involves the blurring of image regions that were identified as containing private content during the previous steps. Several goals exist for this blurring step. The primary goal is that the blurred image content can no longer be personally identified after the blurring step, and that the blurring method is irreversible, such that no algorithms can be used on the blurred image to recover the original image content. Additionally, it is important that while it should be made obvious for the community that the image has been intentionally redacted for privacy reasons, the amount of blurring should not affect the image quality more than required, especially in regions containing false positive detections.

In order to fulfill the above requirements, we first add noise to the input image, and then apply Gaussian blurring by means of convolution. We further use the scene geometry as a guidance to how much blurring needs to be applied at a specific location of the image. Objects such as faces, which are farther away from the camera, appear in the image at a smaller scale than nearby objects and thus require less blurring. Hence we use the scene depth images gathered by the laser scanners (Section 4.5.5) as a means to estimate the dimensions of private features in the image. Assuming a typical natural feature size s_{nat} of 1" (2.54 cm), the projected feature size s_{proj} in pixels

$$s_{proj} = \frac{s_{nat} * f}{d} \qquad (4.8)$$

can be computed for each pixel in the image, given that depth information d is available. The projected feature size in pixels is to decide which σ value to use used for the Gaussian blur at the specific location. We choose a sigma value proportional to the projected feature size.

Due to the fact that the uncertainty of laser scan measurements for a given image grows as a function of the time between the image capture and the time when the actual laser line was captured $|t_{cam} - t_{lidar}|$, we reduce the depth estimate by an uncertainty offset $\boldsymbol{\delta_d}$. This leads to a larger amount of blur for objects scanned at a larger time offset from the camera trigger event. We use a linear uncertainty model, decreasing the depth estimate proportional to the time offset, up to an offset of -2 m according to

$$\boldsymbol{\delta_d} = \max\left(2, \frac{|t_{cam} - t_{lidar}|}{2}\right) \qquad (4.9)$$

An example demonstrates this behavior in Figure 4.31. The input depth map image (a) and depth offsets to compensate for the time difference (b) are combined into a corrected depth map image by subtraction (c), in order to compute the sigma values (d) used for blurring the input image in those regions detected by the privacy protection algorithm. The input image for this example, with highlights indicating detected regions, as well as the blurred output image are shown in Figure 4.32 (a) and (b). All three detected license plates have been blurred sufficiently to render the license plate illegible. Note that the amount of blur applied to obfuscate the person in the back of the scene is smaller than for the license plates, due to the higher distance.

While false positive detections around other parts of the car (rims and lights) have caused some notable blurring artifacts, the amount of blur is less than if a constant blur amount had been used throughout the image.

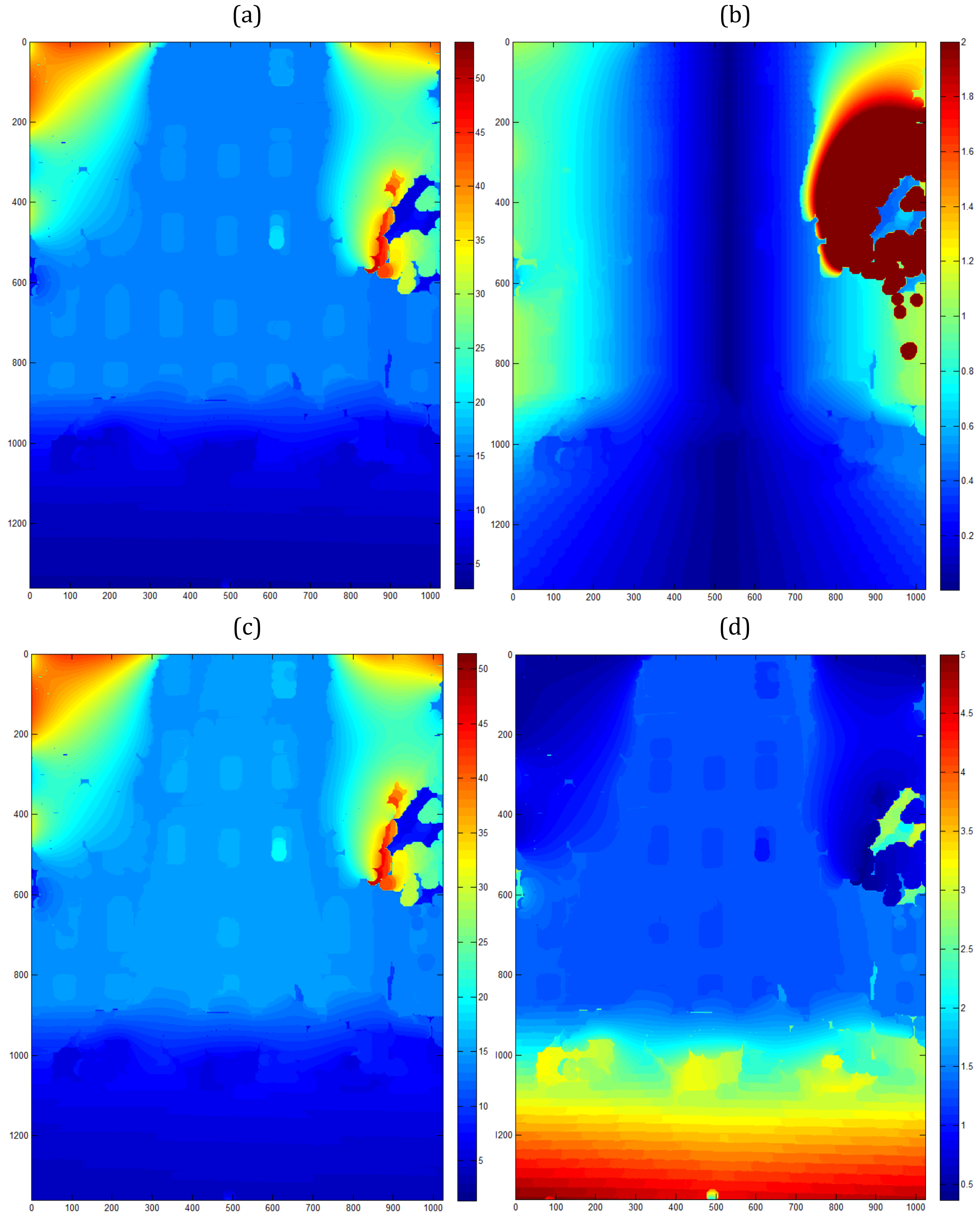


Figure 4.31 Depth Maps used for Depth Dependent Blurring: (a) Input Depth Map; (b) Depth Offset due to Time Uncertainty; (c) Updated Depth Map; (d) Sigma Values used for Gaussian Blur;

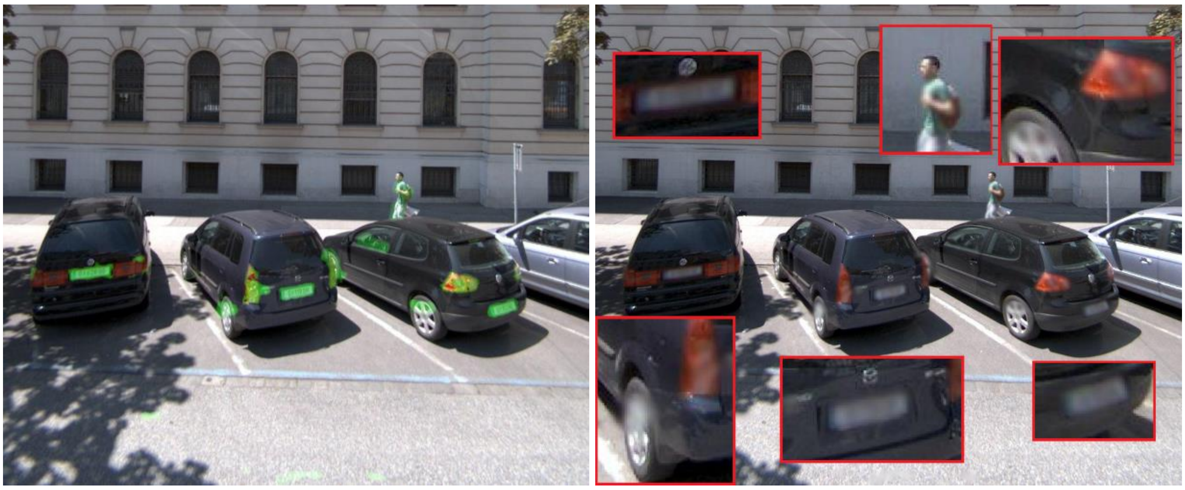


Figure 4.32 Input Image with Highlighted Regions Detected by Privacy Detector (Left); Output Image after Applying Depth Dependent Blurring – Details shown Enlarged (Right);

4.5.8 Combination of Different Methods

The proposed workflow for privacy protection in streetside imagery uses several detection algorithms providing high recall / low precision detection of private objects (faces, people and license plates) in combination with different ways of reducing false positives. Each detector generates binary pixel masks of the same size as the input image, which are combined using Boolean operations to determine the final blur mask used to generate the blurred output image.

Figure 4.33 shows a sample streetside image captured with one of the broadside looking cameras of the UltraCam-M mobile mapping system, containing private objects such as vehicle license plates as well as a person. Although the person as well as the smaller of the two license plates appear at a relatively small scale, they may still be identifiable by someone familiar, and therefore should be detected by the privacy algorithm. The image further contains objects such as buildings, street markings, or vegetation which should be preserved as much as possible during privacy processing.

The detection result of the individual detectors computed on the input image in are shown in Figure 4.34. As one can see the result of the edge based saliency detector (a) has been tuned for high recall / low precision in order to detect most parts of the image containing features of a certain dimension typical to faces, hands, or license plates. Detections have been restricted to a height above ground of 2.2 m as derived from the laser geometry measurements.

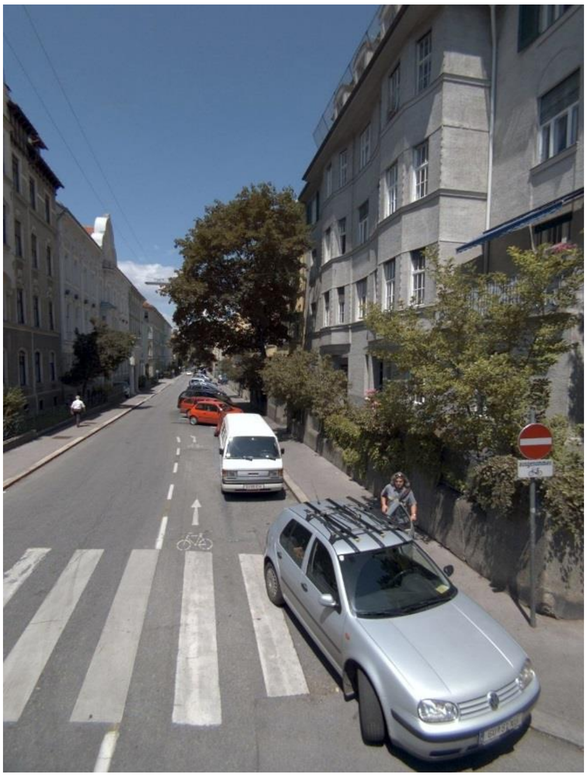


Figure 4.33 Sample Streetside Image with Private Content

Similarly, the license plate detection result (d) is tuned for high recall / low precision, therefore causing many false positive detections. The detected regions generally have rather elongated shapes at an angle compared to the horizontal direction, as expected for license plates.

The result of the color based skin (c) and vegetation masks (d) similarly have been tuned for a high recall and low precision, to avoid false negatives of the privacy detection workflow. Therefore the skin mask also includes other regions with similar colors, such as parts of the vegetation or the red car. The threshold for the vegetation detector are set for high precision / medium recall, thus causing some false negatives, and only few false positive detections.

Finally, the detection result for objects on major planes in the scene are shown in Figure 4.36 (a), covering part of the street as well as parts of the bulding façades on the right. The building façades on the left were not detected as no depth information could be obtained due to the nearly parallel viewing direction of the LiDAR. This fact can also be seen in the depth image computed for the respective camera (b).

Figure 4.35 describes schematically the boolean combinations used to combine the individual detectors and to compute the final detection mask used to control the image blurring algorithm. In order for the final mask to be logically TRUE, the edge based saliency detector AND either the skin OR license plate detectors have to be TRUE, AND NONE of the false positive removal methods such as major plane detection OR the vegetation detection may be true. Note, that a region detected as skin overrules regions also detected as vegetation, therefore the difference of the two is used for removing false positives.

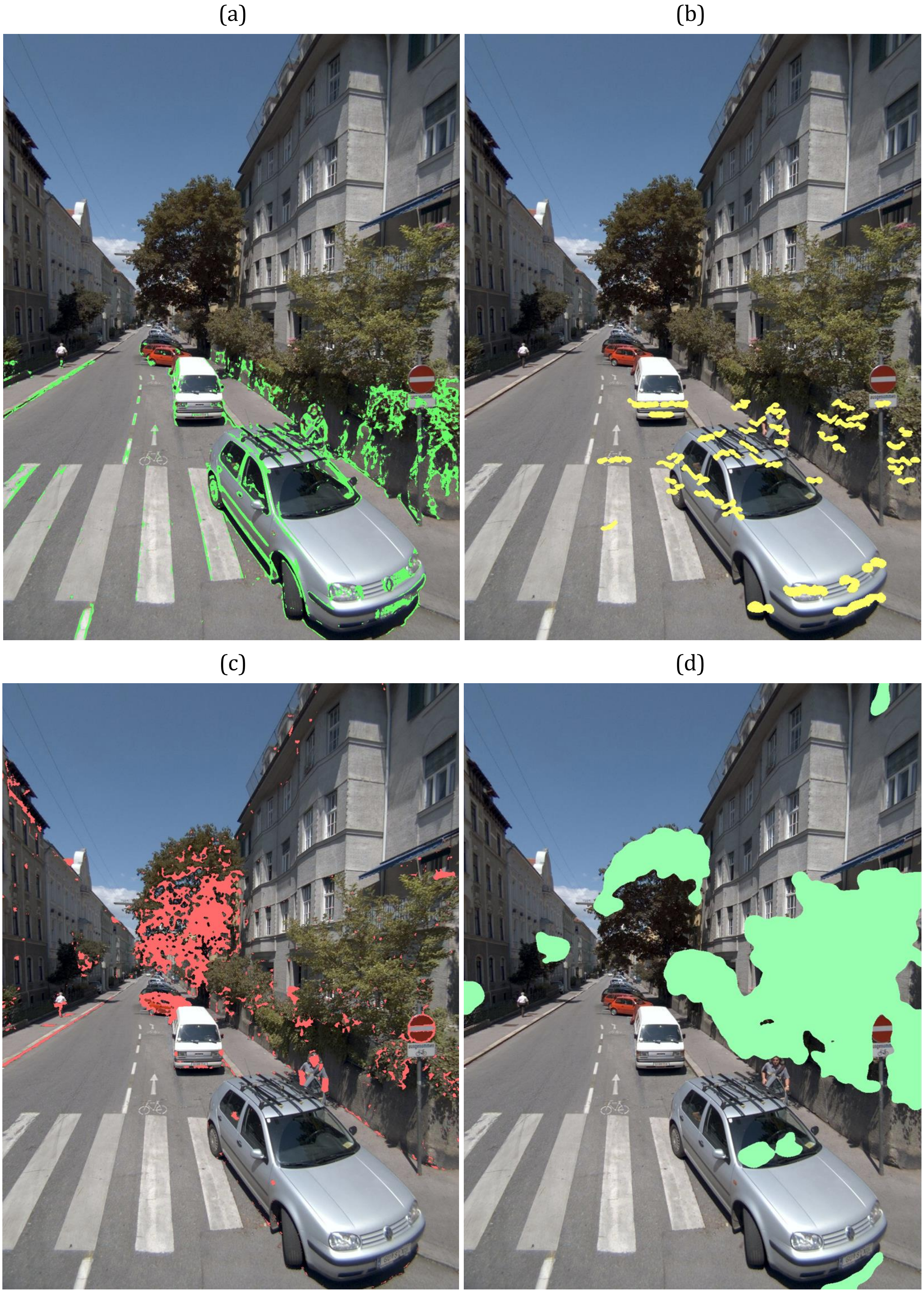


Figure 4.34 Results of Individual Detectors: (a) Edge Based Saliency Mask; (d) License Plate Detector Result; (b) Skin Tone Detector Result; (c) Vegetation Detector Result

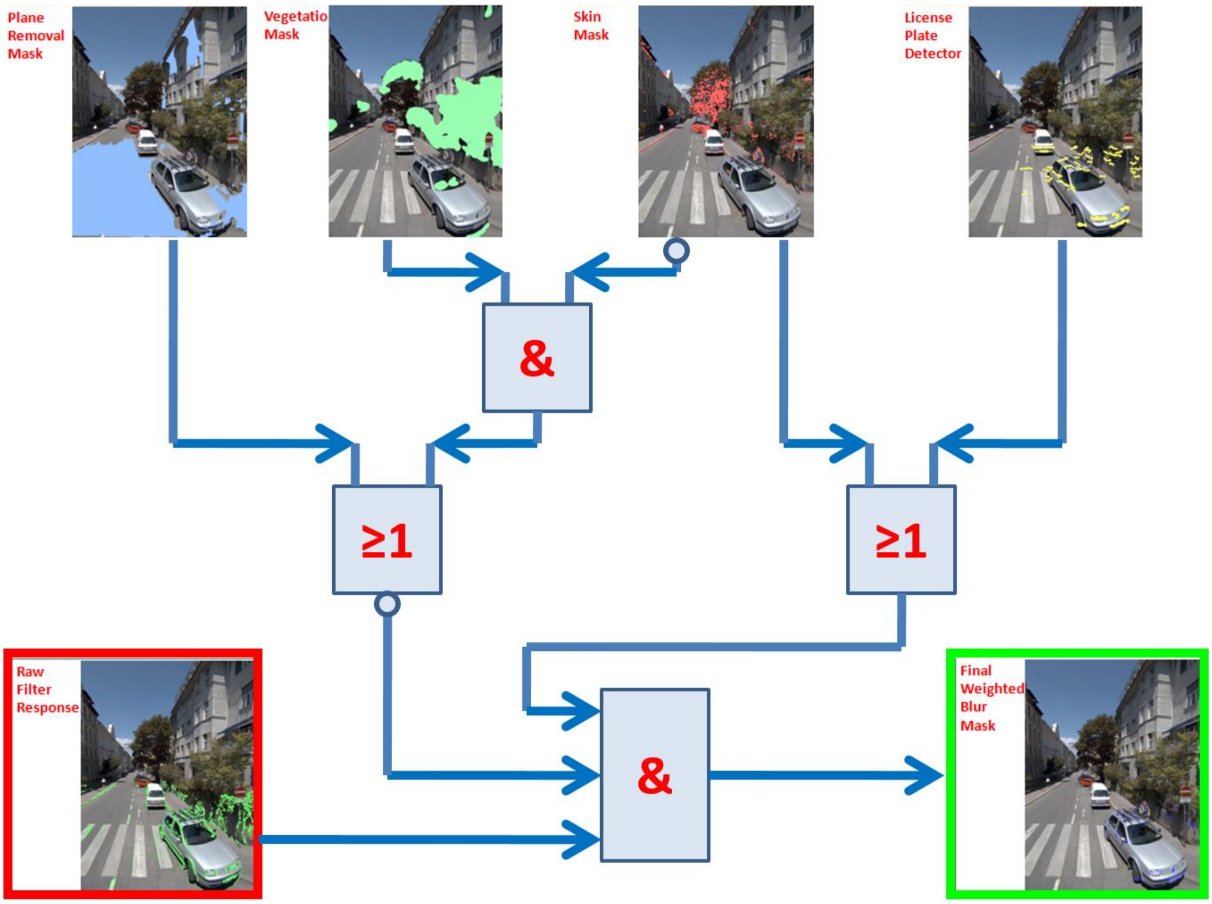


Figure 4.35 Boolean Combination of Individual Binary Detection Masks to Generate Final Detection Mask used for Output Image Blurring

The resulting mask from this operation is depicted in Figure 4.36 (c), with colors indicating whether an object has been detected by the skin detector (green), the license plate detector (blue), or by both detectors (red). Finally, the resulting output image after applying the depth dependent blurring algorithm (d) shows that both license plates as well as the person facing the camera have been correctly detected and blurred by the algorithm. While false positives caused some amount of blurring of non-private regions, the majority of the image such as the building façades remain unaffected. Also, due to the depth dependent blurring algorithm, objects further away from the camera are blurred relatively less, than close-by objects. This can be seen specifically on the example of the farther of the two license plates.

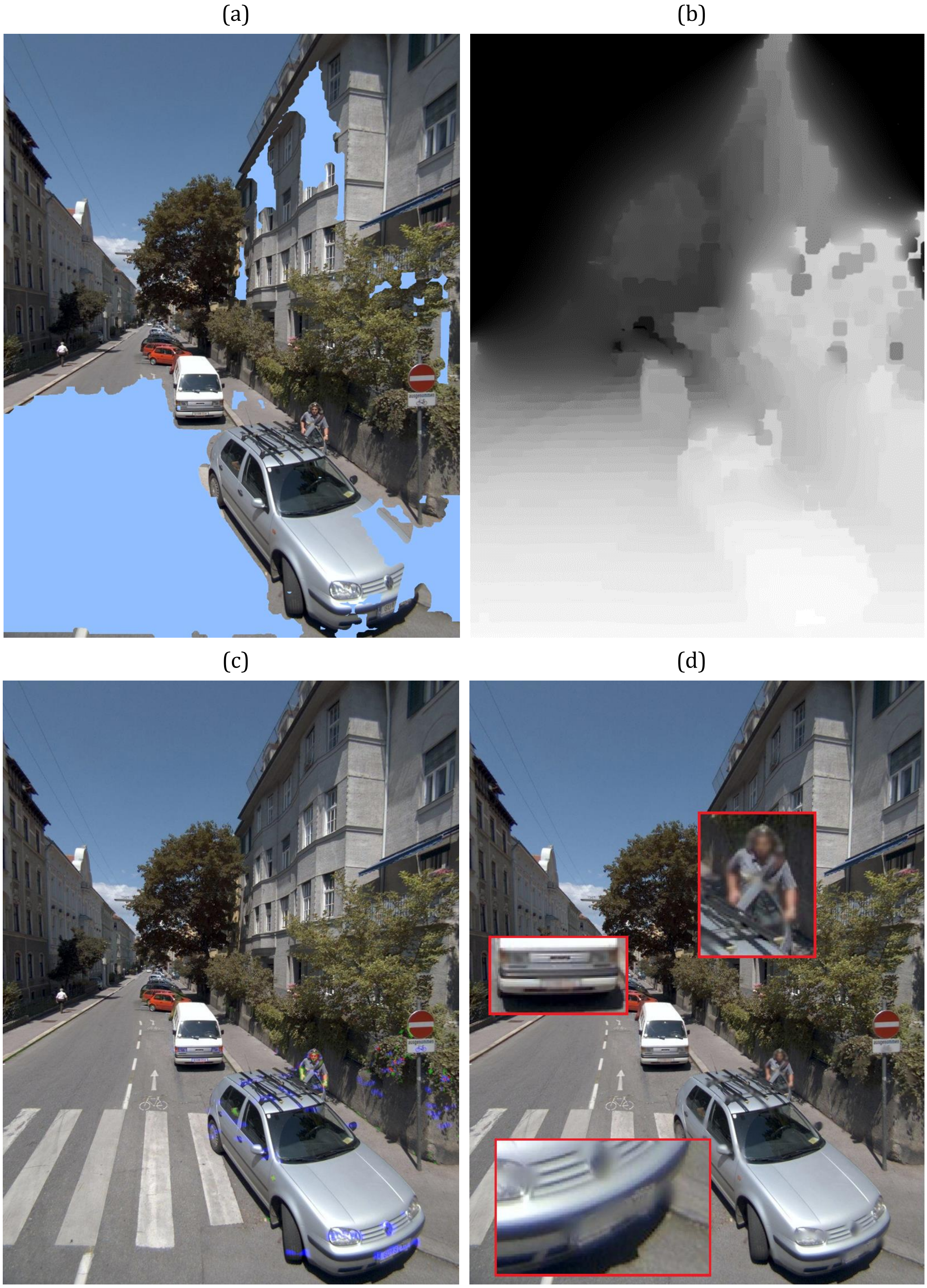


Figure 4.36 (a) Detection of Regions on Major Planes; (b) Depth Image Computed from LiDAR Data (c) Combined Detection Mask (d) Resulting Output Image after Privacy Blurring

4.5.9 Experiments and Results

In order to evaluate the performance of the streetside privacy protection algorithm described in 4.5, we used two different test datasets. The first dataset was captured in downtown Manhattan, NY, and contains a total of 1,472 images. In this dataset, 1,404 faces were manually labelled by drawing ground truth (GT) bounding boxes around people's faces. This dataset does not contain GT labels for license plates. The second dataset, captured in Denver, CO contains 553 streetside images. In this case, 2,488 faces and 43 license plates have been manually labelled as GT. In the second dataset, the face labels were further categorized depending on whether they were frontal views or side views of faces, and whether the face was clearly recognizable or vaguely recognizable. Instructions were given to the labelers, not to label any faces on printed posters, as they don't reflect private content.

The metrics we decided to use for evaluating the privacy protection algorithm are the per bounding-box recall for faces and license plates, as well as the per-pixel false positive rate (FPR) for all pixels outside of any labeled bounding box. The recall is computed as the fraction of bounding boxes which were sufficiently blurred by the algorithm. This number is computed in two steps. First the bounding boxes for which the ratio of detected pixels within the bounding box exceeds a threshold of 10% were automatically detected. This value was chosen, as the bounding boxes were often significantly larger than the actual face or license plate, such that for many correct detections, the actual facial region detected covered only a small part of the bounding box. The blurred output images were also manually inspected to assure that they were sufficiently blurred, and false positives were removed from the set. The per-pixel FPR is computed as the fraction of all pixels outside of any ground truth bounding box, which was erroneously detected.

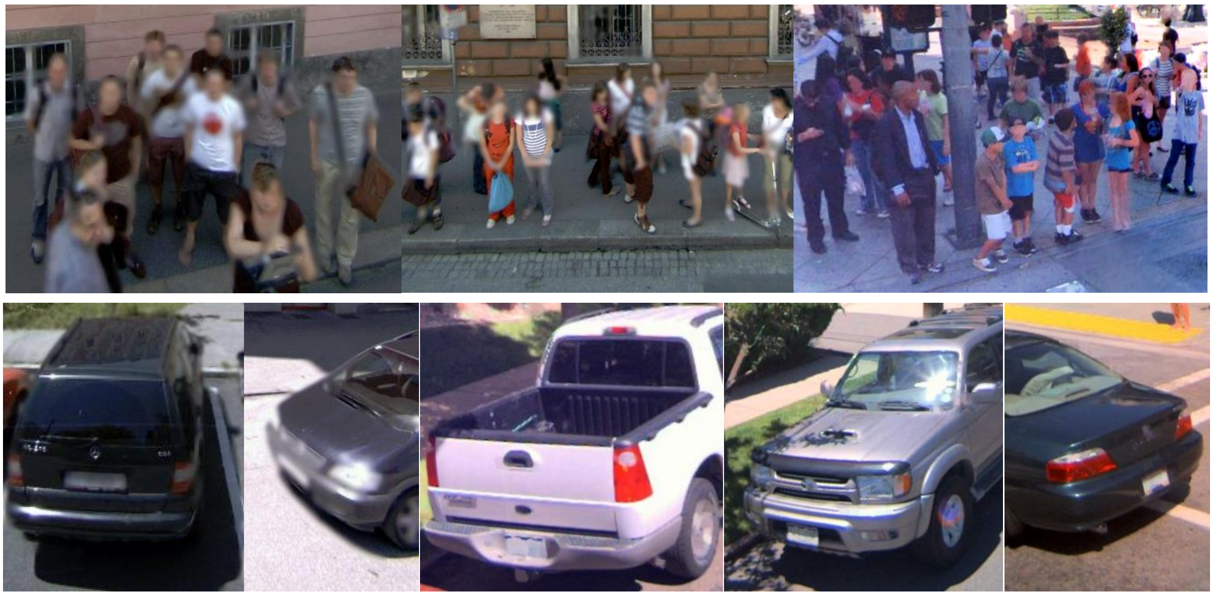


Figure 4.37 Examples of Correctly Identified and Blurred People and License Plates

For the Manhattan dataset, which was used to evaluate the per-face recall standalone without license plate detection, the proposed workflow achieved a per-box recall of 95.4% (1335 detected faces). The per-pixel FPR for this dataset was 0.79%.

In case of the Denver dataset, the results are broken down based on the label type, and whether the license plate detection was used in addition to face detection in Table 4-2 and Table 4-3. The recall for all clearly visible faces (97.0%) is notably higher than for vaguely visible faces (83.5%), while it can be argued that the latter are less important to be detected. The overall recall for all face categories combined is 90.9%, compared to 95.3% for license plates.

GT Label Type	Total GT Labels	Detected Labels	Box Recall	Pixel FP Rate
Frontal Face Clear	596	575	96.5%	1.5%
Side Face Clear	679	662	97.5%	1.5%
Frontal Face Vague	566	459	81.1%	1.5%
Side Face Vague	647	554	85.6%	1.5%

Table 4-2 Privacy Metrics for Denver Dataset without License Plate Detection

The observed per-pixel FPR for the Denver dataset is 1.5% without using the license plate detector, which increases to 2.3% with the license plate detector enabled. Although FPR is relatively high, their is offset by using the depth dependent blurring algorithm, as many false positive regions are located at such a distance from the camera, that the amount of blur caused in the images is marginal.

GT Label Type	Total GT Labels	Detected Labels	Box Recall	Pixel FP Rate
Frontal Face Clear	596	576	96.6%	2.3%
Side Face Clear	679	664	97.8%	2.3%
Frontal Face Vague	566	462	81.6%	2.3%
Side Face Vague	647	560	86.6%	2.3%
License Plate	43	41	95.3%	2.3%

Table 4-3 Privacy Metrics for Denver Dataset with License Plate Detection

After the completion of the described work, Frome et al [27] have reported similar recall metrics (89% faces, 94-96% license plates) for the privacy protection method used for Google Maps, while the FPR metrics reported (0.2% for face and LP combined) are an order of magnitude better than our results. However, the depth dependent blurring algorithm compensates for a substantial part of the FP generated by our system. A visual inspection on Google Maps we performed in April 2009 confirmed that the percentage of faces that were blurred was about 90%, and for license plate it was 95%, which corresponds to the published numbers in the paper. On the other hand, an estimate of the FPR based on visual inspection suggested significantly higher numbers (5%) than those published in the paper. This may be because the data that had been published at that time had still been processed with a prior version of Google's privacy protection system than described by Frome et al.

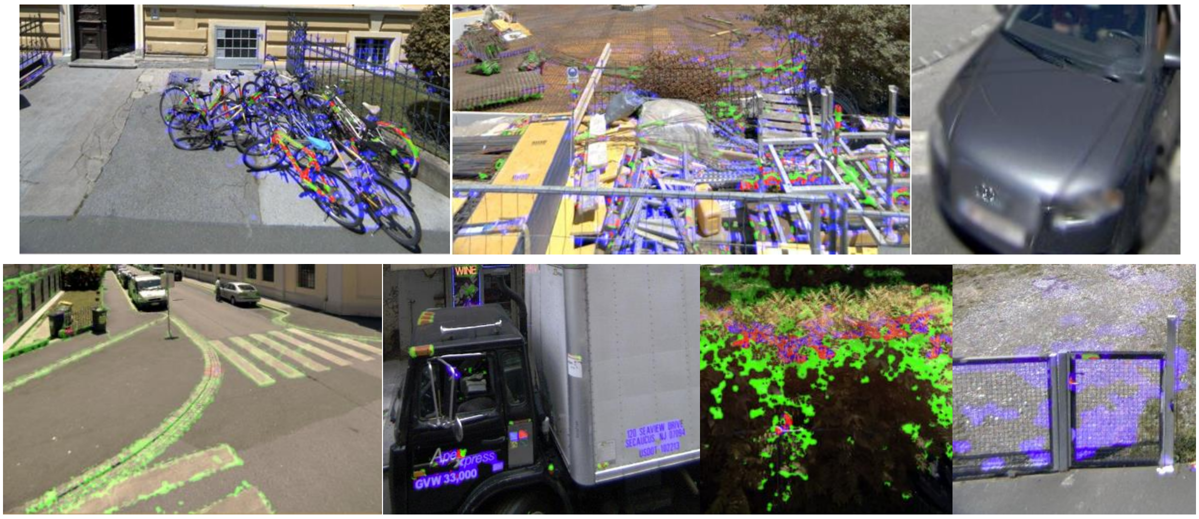


Figure 4.38 Examples of Remaining False Positives

While we believe that the recall numbers for faces and license plates are promising considering the large variability in the appearance of such objects in streetside imagery, there is still a large potential to improve both the recall as well as the FPR. The examples of remaining false positives shown in Figure 4.38 indicate problematic areas, such as highly textured structures that were neither removed by the skin-tone or vegetation detectors, nor were they found to be on a major plane. Particularly bicycles, car lights or rims frequently cause FP detections. Such errors could potentially be reduced by training a learning based method such as [294] on particular instances of alike objects. Other problems occur in cases where one or more of the methods to avoid false positives failed, such as the major plane detection as well as the skin tone detection. One example in Figure 4.38 shows an area where part of a cross-walk was detected due to its high contrast. In this case the LiDAR point cloud did not align precisely enough with the image, which caused misalignments between the consecutive images. In addition the gray balancing had failed, therefore the street was within the color range classified as skin.

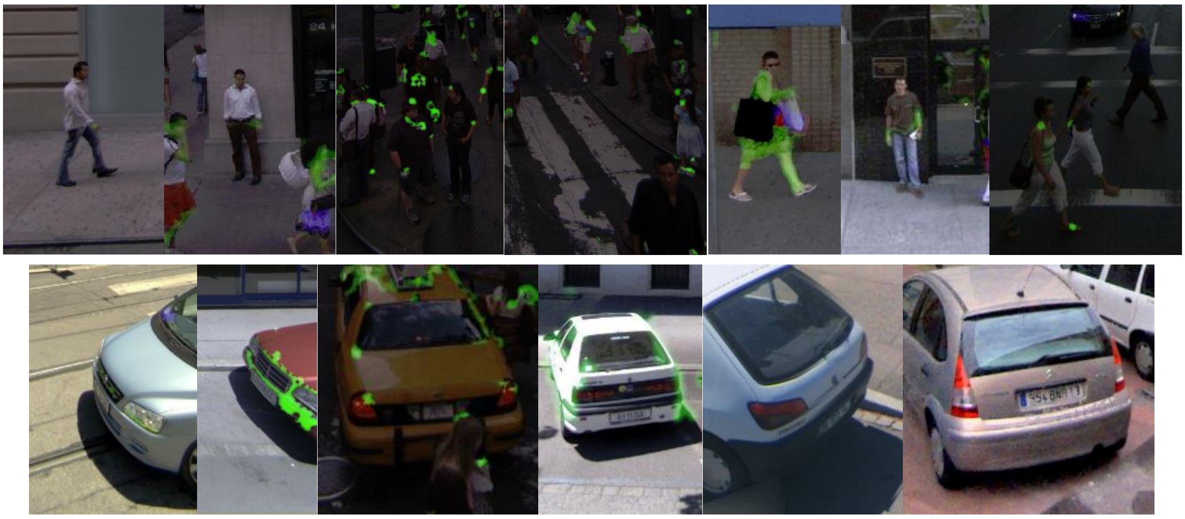


Figure 4.39 Examples of False Negatives

Similar to false positives, false negatives are also occur frequently (>80% of all cases) due to miss-detections of major planes. This is often caused by are dark image regions resulting in miss-matches between private objects and background planes. Using a better, luminosity independent method of comparing gray values such as comparing only the chrominance or using relative instead of absolute thresholds may improve this behavior. In other cases, missed private content includes people standing close to walls or license plates being missed due to their extreme viewing angle, as shown in Figure 4.39.

4.6 Summary and Outlook

In this chapter we have described two of the main contributions to human scale image data in internet mapping, which matured into practical applications in Bing Maps. This includes a capture system for efficient capture of millions of streetside images fulfilling a series of requirements, as well as a workflow for automatic privacy protection for the captured imagery.

4.6.1 Streetside Data Capture

While the system described above (UltraCam-M version 1) reflects the requirements in internet mapping in 2007 and 2008, progress in various domains such as spherical image sensors [348], integrated GNSS/IMU systems [252], laser scanning devices [249, 250] as well as algorithms for data processing [314, 315, 316] has since pushed the limits in this domain.

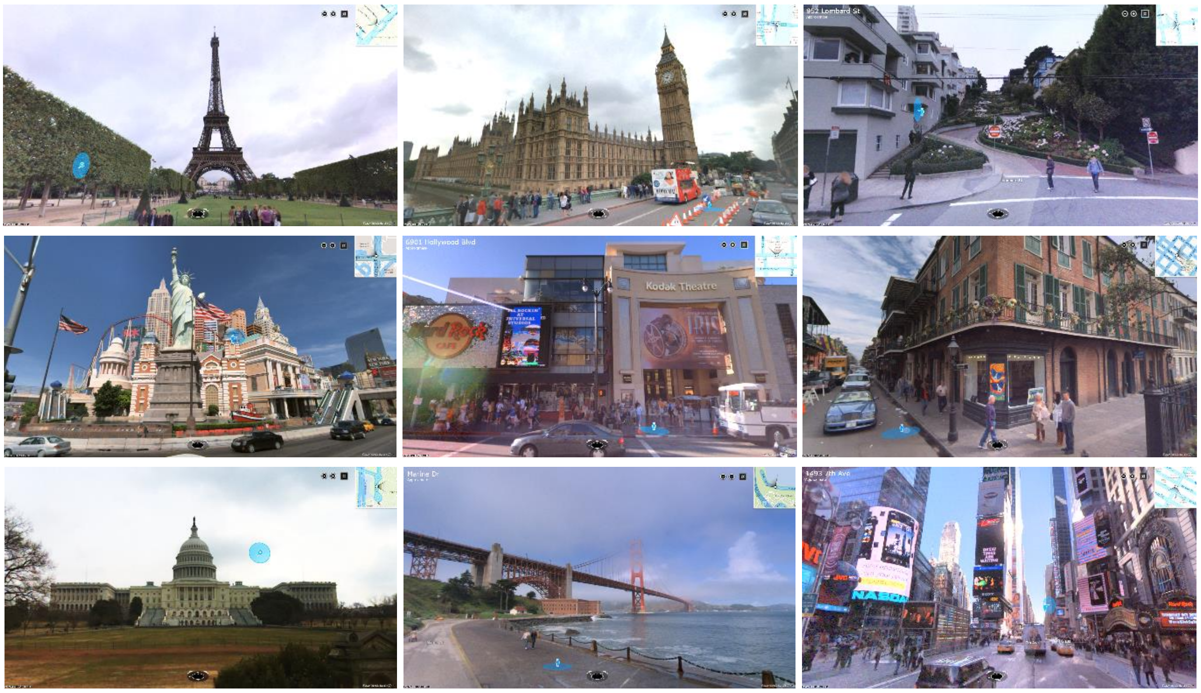


Figure 4.40 Locations Captured with UltraCam-M Streetside System (From Top Left): Eiffel Tower, Paris; Big Ben, London; Lombard Street, San Francisco; Excalibur Hotel, Las Vegas; Hollywood Boulevard, Los Angeles; French Quarter, New Orleans; US Capitol, Washington DC; Golden Gate Bridge, San Francisco; Times Square, New York;

Since the development of the UltraCam-M system, streetside imagery have become an essential component of internet mapping sites such as Bing Maps and Google Maps. Both companies have captured and shared hundreds of millions of images in many locations around the world. While Microsoft has focused mostly on urban areas in the US and Europe, Google has captured data even in remote areas on all 7 continents [349]. Both companies had to face both logistical challenges as well as privacy concerns by communities around the world. The following section defines the requirements for streetside privacy protection in more detail, as well as a proposed workflow to address the problem.

4.6.2 Privacy Protection

We have presented and evaluated a workflow for automatic detection and obfuscation of private objects such as people or car license plates in streetside imagery. We have shown that the presented method successfully detects >90% of people's faces and >95% of license plates, which is comparable to the results reported by Frome et al. for the workflow used by Google Maps [27]. The higher FPR (2.3% vs. 0.2%) is partially offset by the proposed depth dependent blurring.

Note that while automatic privacy processing of streetside imagery is an important and cost-effective step to protect people's rights for privacy, no algorithm can currently guarantee 100% recall rate without significantly affecting the image quality and thus rendering the data usefulness. Therefore, it is essential for internet mapping providers to establish functionality for manual flagging of private objects besides the automatic detection.

With increasing sensor resolutions, both for human scale as well as aerial (oblique) images and the emergence of indoor imagery, privacy protection is gaining more significance, as a larger number of private objects is exposed. Therefore new methods for detecting a broader set of private objects (e.g. house numbers) in a larger range of image scales need to be developed.

In order to retain optimal image quality despite artifacts due to false positives, it may be preferable to remove people and cars from the respective images or even replace them with anonymized content, rather than blurring the affected regions. [350] and [351] have presented interactive and automatic solutions for removing private objects, while [352] provides a method for replacing pedestrians with another one selected from a controlled and authorized dataset.

5 Geospatial Image Matching Within and Across Domains

We have previously explained the varieties of geospatial imagery with a different choice of image scales and resolutions, capture processes used, their associated metadata and specifics of their ownership. Data often complement each other in that strengths of one type are weaknesses of another, and vice versa. While for example systematically collected streetside images have sub-meter accurate geocoding and a well-defined coverage of large areas, they may often be several months or years old and do not contain information about which places are most popular or of user-interest. Community photo collections (CPC), on the other hand, can provide the missing popularity information and freshness, as well as a larger variability of the scene appearance (see nighttime image Figure 5.1). On the other hand they are lacking the geocoding accuracy often required for mapping sites. Additionally CPC data may contain user tags describing features of the recorded scene, which systematic sources can hardly provide. Therefore it would be generally desirable if complementary information from multiple image sources could be combined to improve the overall usefulness on the web.

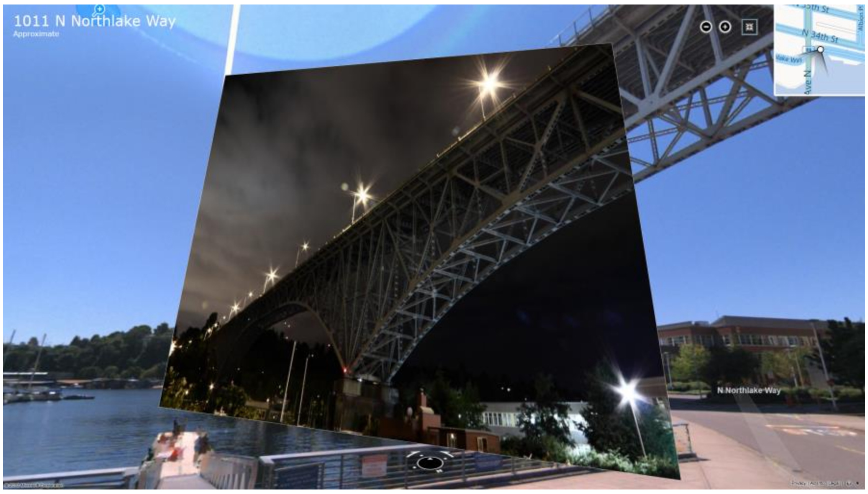


Figure 5.1 Nighttime User Photo from Flickr Superimposed on Bing Maps Streetside Panorama Image by Means of Image Matching (Aurora Bridge, Seattle)

One way of achieving this synergy is by means of image based location recognition or location search, to connect images from multiple sources in order to allow propagation of different kinds of information across data types. Location recognition means in the simplest case recognizing that two or more images cover the same physical location. In a broader setting location search extends to the problem of finding the best matching location for a given query image within a search scope such as a building, street, city block, city or the world.

5.1 Image-Based Location Search

The terms “location recognition” and “localization” using images have been broadly discussed in robotics research since about 1988 [353, 354, 355, 356]. They describe the automatic recognition of a previously visited or recorded location, which is a common problem in robotic navigation and more broadly in the area of “simultaneous localization and mapping” (SLAM) [357, 358, 359]. Other related research areas such as “landmark recognition” [16, 360], “personal navigation“ [361, 362], or “augmented reality” (AR) [41, 359] have an interest in solving essentially the same problem of visually recognizing a location [363].

In the following chapter we describe an innovative workflow to solve the problem of combining image data from different sources in order to maximize their usability for online mapping systems and other geospatial applications. Specifically we propose a scalable geo/spatial image index capable of efficiently matching geocoded and non-geocoded query images to an index containing tens of millions of geospatial images. The precision and recall of the query results are optimized by using priors for the location estimates as well as the image rotation. Both image query as well as image ingestion to the index are in real time without the need to rebuild the index.

The presented solution has applications in several of the areas mentioned above, although it is primarily intended for city scale location recognition by means of matching new image data to previously available and geographically referenced images. City scale in this case refers to a database of several hundreds of thousands up to millions of images. Apart from the large search space, an automatic image matching system has to cope with large dissimilarities between such images. These are caused by differences in the capture system and the environment such as the following:

- Differences in image quality (resolution, lens distortions, contrast, noise, …)
- Differences in scale and perspective (human scale vs. aerial, small overlap regions, …)
- Differences in illumination (day, night, dawn, sunshine, shadows, artificial lighting, …)
- Dynamic scene contents and occlusions (people, cars, clouds, trees, …)
- Major scene changes (new constructions, renovations, …)
- Large time differences leading to a combination of the above.

One of the goals of this work is to achieve a high matching rate, ideally **100%**, while keeping the rate of false matches below an assumed user’s error tolerance of **1%**. Another goal is to allow both batch processing of **tens** of query images **per second**, such as from a CPC like Flickr, as well as real time queries from mobile devices with end-to-end latencies below **5-10 seconds**.

5.2 Related Work

The task of matching multiple images according to a set of features is one of the most important tasks of computer vision [364]. This has already been a topic for several decades, using a great variety of features and algorithmic approaches. This task becomes especially challenging if the image contents differ significantly in their radiometry, geometry, resolution, perspective or other parameters. The task also becomes more computationally expensive if a large number of index images needs to be considered for a single query image.

Let us consider some examples of using image matching to determine the location of the contents present in a query image within a series of index images, or an estimate of the camera pose relative to some world coordinate system.

Structure from Motion

While they are not the main focus of this work, structure from motion (SFM) methods are related to location recognition in that they also require the association of multiple images captured in the same location by identifying commonly visible scene elements. SFM methods proposed by Snavely et al. [15], Goesele et al. [2] or Agarwal et al. [11] match local image features such as SIFT [364] across images to identify common scene elements. The matches can hence be used to reconstruct the 3D geometry of the scene. SFM has been used in internet based applications such as Microsoft Photosynth [163] or Google Photo Tours [266]. While [15] and [2] aimed at small collections of tens to hundreds of images, [11] expanded the scope to a large collection of 150,000 images of the city of Rome downloaded from CPC. Additionally several location recognition methods [365, 366] use the resulting 3D structure and features as a basis model for “6 degree of freedom” (6DOF) localization.

Global Landmark Recognition

One application which often does not require 6DOF poses is the visual recognition of a particular instance in a global database of landmarks. This may be useful for organizing and auto-labeling a photography collection, e.g. after returning from a vacation. Databases are often created based on data from CPC which are either manually or automatically (based on GPS or user tags) grouped into individual landmarks. Li et al. [16] propose a method for automatically selecting a few hundred iconic CPC images from tens of thousands of images by means of clustering and creating an “iconic scene graph” between clusters. SFM scene models are hence computed from the iconic data to provide a compact summary of individual landmarks. Queries against the model using both visual appearance as well as geometric verification lead to recall (=TPR) numbers of up to **40%** for **3** landmarks evaluated.

Alternative, classification based approaches by Li et al. [17] and Zheng et al. [360] and have shown that significantly more landmarks can be modeled and recognized automatically at even higher recall. Li et al. achieve **45%** recall by clustering 30 million CPC images into **500** distinct landmarks using their geocoding, and training a support vector machine (SVM) [367] to perform classification based on SIFT based bag-of-word and textual features. Zheng et al. use similar methods to obtain 21.4 million landmark images from CPC and tour guide web pages, from which **5312** individual landmark clusters are identified. Queries are done using KD-tree [44] nearest neighbor search in a database of image features. Zheng et al. report recall performance of **46%** with an FPR of **11%**.

City Scale Location Recognition

Location recognition differs from landmark recognition in that its aim is to provide a precise (meter to sub-meter) location or 6DOF pose for a query image within a larger (e.g. city-wide) region, rather than simply the association with a particular landmark. Hence a significantly larger corpus (e.g. several hundred thousand) of geo-positioned index images per city may be required for continuous coverage. Research by Johansson et al. [368], Robertson et al. [369] and Le Bris et al. [370] made use of the fact that many urban scenes primarily consist of planar building façades with linear features, by automatically extracting vanishing point directions [371, 372] from images and rectifying them into canonical views. Hence in order to confirm a match between two rectified images, the search space is reduced to the parameters of a similarity transform - offset, scale and (optionally) rotation. While [368] used summations of difference images to align image-edges in X and Y independently, [369] instead applied wide baseline matching using local color statistics as features. Later, [370] followed the same idea, but used SIFT features to replace color features. After confirming the correspondence between a query image and an index image, a 6DOF pose can be determined based on the vanishing point directions and known index image orientations. All three methods achieved satisfactory results on “several” images, but were not evaluated on city scale datasets.

Zhang et al. [362] proposed an approach based primarily on SIFT features for correspondence search, combined with a robust geometric verification using RANSAC [45] with a homography or fundamental matrix model. Finally the camera pose is estimated by triangulation using the point correspondences with two of the matched index images. [362] reported similarly good results as the above authors for a small set of 22 query samples.

In contrast to the work reported above, Schindler et al. [363] aimed at location recognition within a much larger corpus of **30,000** systematically captured index images. This could be achieved by employing an image retrieval scheme based on [40] and [30] for efficient image ranking of index images. For ranking, a selection of the most location-distinctive quantized image features was used, leading to a better ranking performance than the original method [30]. This approach led to a recall rate of up to **73%** for the first image in the ranked list. However the authors did not perform any verification of the matches, hence the FPR is expectedly high (**27%** = 1 - TPR).

Other more recent work on location recognition by Irschara et al. [365] and Yunpeng et al. [366] use SFM for creating a “world model” used for localization. While [365] also uses image retrieval in a database of real and synthetic views of the world model to obtain valid matches, [366] rather compresses the model into a subset of “prioritized” features, which are matched to query features using nearest neighbor search and reasoning based on visibility. In both cases the resulting matches can be used to estimate 6DOF poses. [365] was evaluated on several landmark models ranging between roughly **100** and **1,000** images from CPC and targeted captures. The achieved recall rates ranged between **96%** for query images captured close to the index images, and **43%** for query images following a different path. [366] used an SFM method based on prior work in [11]. Hence significantly more index images (**1,300** to **16,000**) could be used to generate the compressed SFM models. While no FPR numbers were reported, the achieved **94%** are remarkable.

Simultaneous Localization and Mapping

A specific application of location recognition is the recovery and loop closing problem of SLAM systems for robotics and AR [359, 373]. In order to connect to a previously created map either after initialization or to recover after tracking failures, visual localization is commonly performed [358]. While tracking in large spaces with loops, it is also common practice to connect to previously observed frames to create additional constrains about the map geometry [374]. Similar methods as the ones described above, which are able to compute 6DOF poses relative to an existing map, may serve this purpose besides specific methods developed for SLAM applications.

Local Image Features

Most of the recent research on location recognition [375, 369, 376, 377, 362, 363, 378, 17, 360, 16] makes use of local image features, which, in contrast to global features, describe properties of smaller regions within the image. The advantage of this approach is that when correct correspondences between regions of an image pair can be made, they can be used to compute a more precise geometric relation between the images. On the other hand, local image features usually require more processing steps and so are more expensive computationally. Matching local features usually requires four steps. First, salient image regions need to be found by an interest point detector, second the feature descriptors from these image regions get extracted. These feature descriptors of multiple images are compared and matched and the matches are verified geometrically.

Typically, interest point detectors are designed to find salient local image regions such as corners or blobs in a scale space [379], by using a mathematical definition (e.g. Harris corners [380], Laplacian corners, Laplacian of Gaussian, Determinant of Hessian [381] etc.). Research has been performed to develop interest point detectors that are possibly invariant to changes in offset and scale [382, 376, 42], view point and illumination [383], and ideally detect the same interest point at the same scene location repeatedly. Since 3D viewpoint changes usually cause more or less large local deformations of image regions, invariance to affine [384, 385] or perspective [386] distortions can contribute significantly to the matching performance. Other research aims at computing interest points very rapidly for applications running on mobile hardware, e.g. FAST interest points [387]. More research has been done on evaluating and comparing the performance of different interest point detectors [388, 50, 389].

After the interest point detection, image patches are extracted around each point, often considering scale and orientation parameters determined by the interest point detector, from which feature descriptors can be computed. A primary motivation for feature descriptors is to compress the information contained within an image patch into a much smaller vector, to simplify the feature correspondence search across multiple images compared to simple correlation. In addition, features invariance to changes in scale, position, lighting and orientation is desired. A common approach to this is by computing statistical information about the distribution of gradients in an image region. The most frequently used feature descriptor is SIFT [390, 364], which sub-divides the square image patch into 4·4 equally sized regions, and computes for each region a histogram of image gradients, which is quantized into 8 bins each. This leads to a 128-dimensional descriptor for the image region. Derivatives of SIFT are SURF features [391], Viewpoint Invariant Patches (VIP) [386] as well as DAISY features [392, 393, 394]. Several comparisons have evaluated the quality of the different feature descriptors [395, 43, 389, 50] which find SIFT generally to be superior compared to other methods.

Correspondence Search and Verification

Once the interest points are found, they need to be matched to the database images. As exhaustive search through all indexed descriptors is computationally expensive, a common solution is to organize the *n* descriptors of single or multiple images in a KD-tree structure [44], for efficient nearest-neighbor search in *O(log n)* complexity for a given query feature [376, 377, 15, 11].

Even much faster feature matching can be achieved by using quantized image features, also referred to as “visual words” [40, 30, 363]. A large number of image feature descriptors are clustered into a visual vocabulary, each of which is basically represented by a single integer number. For each index image, a list of the included words is saved, and an inverted file index can be generated which contains for each visual word a list of images in which it appears. Matching of a query image involves feature quantization and then using the inverted file table to find possible image matches. Index images can be quickly ranked by the number of words overlapping with the query image, which is usually weighted by some a-priory likelihood for each word. Using this method, millions of images can be searched within less than a second, which makes it very attractive to large scale image search problems, such as location matching. Examples of location matching based on visual words are [363] and [365].

The last step is typically a geometric verification of the point matches, to filter out mismatches, which occur frequently. This often uses the RANSAC algorithm [45], for robust model estimation even in the case of many outliers. A variety of models, such as Fundamental Matrix, Homography, 6-DOF pose estimation etc. can be used to verify the geometry.

Alternative Approaches

While variants of the above workflow are common, other methods are also worth mentioning. Lilja [396] proposed a seed and spawn algorithm that tries to grow the matches starting from some strong seeds, by using geometric reasoning through the image space and scale space [379].

An alternative to using patch based feature descriptors for matching is edge information (edgels) either to support the location matching effort, or for later pose tracking within the world model, such as shown in [397]. Global image features, which contain a global description of the essence of an image (gist), comprise another alternative to local features. They are typically derived by simple statistical analysis or by image understanding methods, such as color histogram information, image texture statistics or statistical descriptions of the image content, and contain a much compressed representation allowing more efficient retrieval of related images.

Jacobs et al. [398] have presented a different approach for approximate geocoding of images, by correlating the time-modulation of image intensity in videos recorded by stationary web-cams to the pattern of cloud-motion derived from satellite images. The major advantage of global features is the relatively high speed of matching, even in the presence of a very large number of index images (>> 1 Million). Nevertheless, this is often outweighed by the disadvantage that positioning can usually only be done roughly, and with a large remaining uncertainty, which renders this method inappropriate for applications such as AR.

Furthermore, while we focus on matching by using natural image features, frequently used tools for camera localization in AR are artificial markers such as those provided by ARToolkit [399]. Artificial markers are designed to be easily detectable, even on mobile devices [41, 400], but they impose the disadvantage that localization and tracking can only work in very limited areas where markers are located.

5.3 One Query Image in a Sea of Index Images

While a rough geo-location of user photographs already simplifies the task of exploring images by their location from a top-down view, it may not be as pleasant an experience when viewed from a "human-scale" perspective, such as within a streetside- or indoor- scene. In this case it would be desirable to have a more accurate alignment of the photograph with the underlying model such as shown in Figure 5.1 - ideally pixel-accurate.

Not only could the image be observed from a perspective similar to the one from where it was taken (putting the observed scene in the context of its surrounding) [49], but it would also be possible to augment the image by relating to it known information about the world (such as the names of streets, buildings, shops, etc.). Knowledge of a photo's position and orientation may also help organizing photos into groups based on scene semantics, offering a better browsing experience of the photos [159]. This could be achieved in an offline process, to more accurately geo-position a set of images, and augment them with the desired meta-information. If the process of aligning the image is fast enough, and computationally cheap, this could also be done in close to real-time, ideally on a mobile device, and be the basis for certain augmented reality (AR) scenarios.

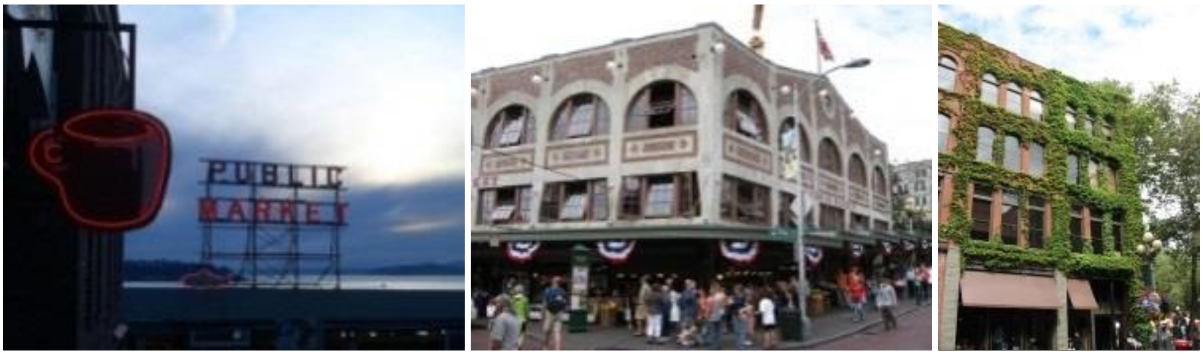


Figure 5.2 Samples of Geocoded Query Images from Community Photo Collection (Flickr)

The primary problem we want to solve is hence the matching of an arbitrary query image to an existing set of geospatial images present in an index. A secondary problem is the dynamic ingestion of a new query image into the image. The index may contain a mix of both human scale panorama images as well as user images previously matched.

For offline-batch processing, as well as for real-time applications, the input data consist of one or more query images (See Figure 5.2). These are often associated with a geo-location (latitude, and longitude), as well as some estimate of the error radius r of the used geo-tagging method. Some image may have additional information about the orientation of the camera, while other images may only have a coarse definition of the location (e.g. city name). If a-priori location and search radius are known for the query image, all indexed images within this region are within the scope. If no scope is defined, the whole index is considered to be within the query scope.

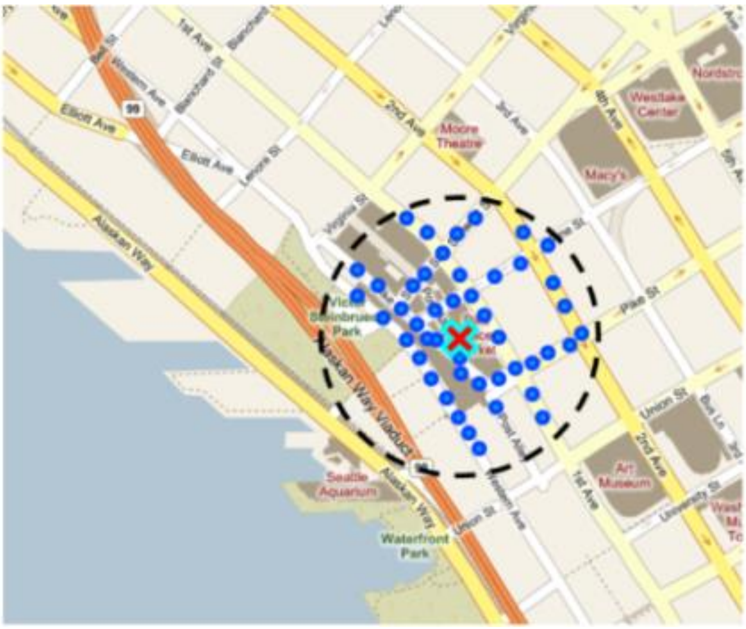


Figure 5.3 Overview of Streetside Panoramas within Search Range (Blue Circles) as well as the Query Image Prior Location (X)

One assumption made henceforth is that the images are always oriented nearly horizontally, which applies to most pictures available on photo sharing sites, since users presumably rotate the images before uploading them. In addition, newer point and shoot digital cameras as well as some cell phone cameras, contain accelerometric sensors, for estimating of the gravity vector with respect to the image [401]. Not all images in the set necessarily have to be outdoor images, or are taken in an area where index images are available. Therefore an algorithm must evaluate whether a match is correct, based on some quality criteria.

The accuracy radius *r* can often be extracted from image metadata, otherwise a default setting (e.g. $\boldsymbol{100\,m}$) may be used. According to this, a search scope can be defined, restricting the search to images within the area (See Figure 5.3). As one can see, a radius *r* of 100 m can span multiple city blocks. Therefore, in some cases a large number of human scale panorama images ($\boldsymbol{300}$ to $\boldsymbol{1{,}000}$) and even more user images need to be taken into account during the matching process.

Another example of a query photo is shown in Figure 5.4 together with a panorama image and a user photograph in the index, taken at the same location. The 360° view of the panorama has been warped into a continuous two-dimensional image using spherical projection (see Section 3.4.5). The x-axis corresponds to the angle around the vertical panorama axis ("Panorama-Longitude"), and the y-axis corresponds to the angle from a horizontal plane in the panorama ("Panorama-Latitude"). We assume that the index images have been oriented nearly horizontally which restricts valid transformations between query and index images to a certain extent (+/- 45 degrees). While we support matches between two central perspective images or between a central perspective image and a panorama, we currently exclude matches between multiple panorama images for simplicity.

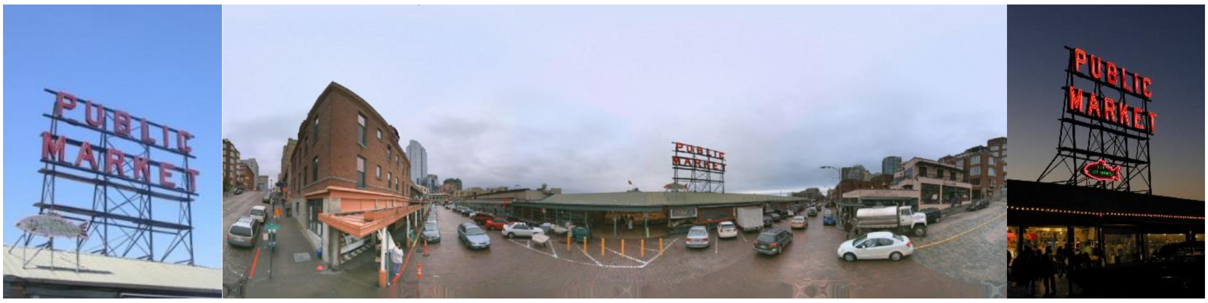


Figure 5.4 Query Image to be Matched (Left); Sample of Bing Maps Streetside Panorama Image in Index (Center); Sample of Previously Added User Image in Index (Right);

Note that while the examples given here contain only outdoor images, the applicability of the method described is not intentionally limited to this scenario. The same method could potentially be applied for indoor images. Evidently, the chances of matching depend largely on the contents of the captured images and whether they contain sufficient overlap of recognizable scene content.

5.3.1 Bing Maps Streetside Photos

In our previous work [25], we had presented a workflow for the location-search problem described above, allowing reliable geo-positioning of query images by means of image-matching using local image features. However the compute time of several minutes per query image, due to the exhaustive search within a search radius *r* restricted the approach to offline batch-processing rather than real-time matching. Additionally, no dynamic ingestion of new images for future queries was supported and the index was restricted to exclusively contain panorama images.

The workflow was able to match roughly geo-coded query images from photo sharing sites like Flickr to a trellis of precisely located 360° streetside panorama images, hence providing **pixel-accurate** 5-DOF pose (camera position and orientation without scale). It was optimized to handle input images even in the presence of significant changes to the camera pose, radiometry and scene content such as images taken at night, or old historic images.

We demonstrated the algorithm on a database of **300,000** streetside images covering a whole city in order to show its usefulness for a vision-based augmented reality system. The algorithm successfully matched **59.5%** of the verified test dataset in combination with a low false positive rate of **0.5%**. This performance could be achieved even though the test data included a subset of very challenging images, including night-shots, images that were very blurry, or had only a small overlap with the panorama images. Problems occurred mostly when query images had a large number of features due to repetitive structures on textured objects, or if the field of view of the query image was too limited and didn't contain enough unique features to allow reliable matching.

The workflow combined several key elements. First, we used descriptor based correspondence search, which was constrained by the feature orientation using orientation prior information. Since urban scenes often consist of partially planar objects, we used a homography model for geometric verification, reducing the chances of mismatches [293]. The method further involved a second guided matching phase after an initial homography estimation to increase the match density. Additionally match hypotheses were verified by image correlation.

In order to distinguish true from false matches, we used a matching confidence score, which was based on various metrics from the matching process, such as the inlier count, the distribution of the matched points in both the query and index image, the mean reprojection-error, the mean Euclidean feature distance between all feature pairs, as well as a correlation coefficient between the two images.

Results from this work were further demonstrated to the public as part of the Bing Maps Streetside Photos Community Tech Preview [46] presented at TED in 2010 [402]. The way of showing user photos superimposed onto streetside imagery has been described in [49].

5.3.2 From Offline to Real-time

Based on the prior work [25] described above, we propose several major improvements related to the scope of the work as well as the algorithmic performance. While the previous system was designed to match user photographs in an offline process to a static set of panoramic images within a search radius, the proposed system extends the scope in several ways. Though determining a **pixel-accurate** alignment between query images and panoramic images is still a goal, the proposed method further supports image matching to other user contributed images, dynamic insertion of new images into the index, and reducing the restriction to search in small search radii. Several major performance improvements were required for the transition from an offline batch-process taking minutes per query to a real-time system capable of matching new photos to the existing index within **5-10** seconds.

A drawback of the purely pairwise matching used in the prior system was the fact that the compute cost strongly depended on the search radius r and the density of existing images d in the area as per the complexity $O(r^2 \cdot d)$. Therefore, it was limited to a small radius (e.g. 100 meters) to avoid excessive search times. Furthermore exhaustive search is only feasible if geocoding information is available, which applies only to a **3.6%** subset of user photos available in CPC. As a means to avoid these limitations, we employ image ranking prior to pairwise matching, based on the "bag of features" approach described in [40] and [30]. Thus only a subset of the ranked list of images returned by ranking needs to be matched (e.g. 200) leading to a significant performance improvement for search radii above 100 m. Thus even images without geocoding information can be matched with this approach if the visual ranking succeeds. Optionally we employ rotational scoping during image ranking, such that only features with roughly the same orientation are assigned the same visual word index.

With an aim to achieve better performance of the system for real-time applications, we further propose a dynamic version of the visual word based index, allowing insertion of new image documents in quasi real-time of less than 5 seconds. While typical retrieval methods based on the "Term Frequency – Inverse Document Frequency" (TF-IDF) scoring function [403] allow retrieval of image documents in a static set of images, the proposed method uses a simpler scoring function allowing easy dynamic document insertion and removal while sacrificing little retrieval performance. Thus query images can be matched to other user images which were inserted into the index only seconds earlier.

The processing cost for pairwise matching in our prior work was relatively high, due to the way orientation constrained descriptor matching was implemented, and due the need for a second matching phase with guided feature matching and correlating the actual images. We therefore want to eliminate or simplify several steps. These proposed changes include a single pass version of rotationally scoped approximate nearest neighbor search using a modified KD-tree [44], a fast similarity based RANSAC for geometric verification using hypotheses derived from individual point correspondences, and an optimization step for the geometric alignment using a similarity or homography model.

The combination of these changes leads to significant speed improvements from several CPU minutes to **17.2 CPU seconds**, while at the same time improvements to recall from **59.5%** to **73.3%** can be achieved.

Further substantial quality and speed enhancements are made to the local image features used for retrieval and pairwise matching. As an alternative to the Laplacian detector used in our prior work, we use both local maxima and minima of the "determinant of hessian" (DOH) [381] function, as also proposed by [42]. With an aim to enable feature extraction on mobile devices we use a recursive Gaussian blurring method [238] before computing the DOH score, proposed by Young and Van Vliet [404].

5.4 Extensible Real-Time Geospatial Image Retrieval Workflow

As a basis for image matching we use local image features, extracted around salient image regions determined by an interest point detector in different levels of an image scale space [379]. For each interest point a high dimensional descriptor vector is computed, describing the gradient statistics within a patch around the detected interest point. Features are detected for each query image as well as each of the candidate images in the index in the same manner. Details of the interest point detection and feature extraction step for one variant of feature are provided below (Section 5.5).

Feature extraction is followed by a query in two phases. Initially a bag of features (BOF) [30, 40, 405] based image retrieval method is used to efficiently find the most likely match candidates from the entire index based on a ranking (5.6) Hence a more expensive and more reliable post-verification (5.7) of the top ranked images is performed in order to classify them into correct and incorrect matches. Both steps use the same local image features.

Ranking initially uses a quantization method to find a representative scalar visual word index for each of the detected image descriptors. The set of visual words is hence used to rank all indexed images based on the co-occurrence of the same visual words as in the query image. To improve the reliability of the ranking results, we propose a rotational scoping method to assign only features of similar orientation to the same visual word index. For dynamic ingestion of new features into the index, we further propose a dynamic version of the index using a simplified ranking method.

Post-verification is pairwise between the query image and each of the top ranked candidate images. It uses descriptor based nearest neighbor search by means of a KD-tree algorithm [44] to solve the correspondence problem between the query and candidate images followed by a geometric verification using a fast RANSAC [45] method. For improved reliability of descriptor based matching rotational prior information about the image orientation to constrain possible matches. The geometric verification initially estimates a similarity transform using a fast similarity RANSAC algorithm, followed by optimization using the Levenberg Marquardt method to find optimal solutions for the similarity and homography transforms between the images. The inlier count for the geometric verification is used as a metric to decide whether or not to accept a match, and to rank all matched images.

Since the system is designed to handle both central perspective image data as well as panoramic images, certain distinctions have to be made during the different steps based on the image type. For example more features have to be detected to cover the larger viewing angle of panoramic images. Additionally the post-verification includes a panorama window selection step in order to determine which section of the panorama overlaps with the respective central perspective image. While we assume in the following sections the query to be a central perspective image and the candidate image to be either central perspective or panoramic, this should not limit the scope of the algorithm. In case of a panoramic query image and a central perspective index image, the image features are swapped and the same methods can be applied in the reversed direction.

5.5 Feature Extraction[3]

After the initial preprocessing of the images (which are resampled to be ≤ 640 Pixel in dimension and converted into grey-scale), interest-points are detected and corresponding feature

[3] The methods for interest point detection and descriptor computation described here have been developed by a team of people at Bing Mobile led by David Nistér.

descriptors are extracted from both the query and index images. For our prior work described in [25], we had used a Laplacian interest point detector to detect a similarity reference frame around each location (Offset, Scale and Orientation) in combination with a version of a Daisy feature descriptor with 32 dimensions developed by Winder and Brown [392, 393, 394].

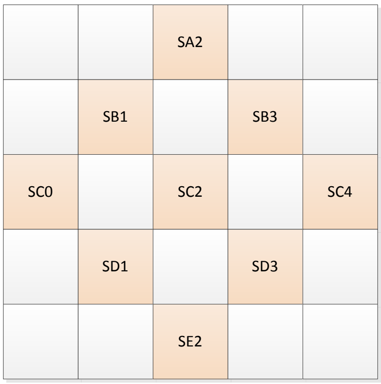
In order to improve both the detection speed as well as the quality performance of the features for image ranking and pairwise matching, we developed a significantly optimized version of the hessian detector, using the determinant of the Hessian (DOH) matrix of a local pixel neighborhood as the saliency measure. Additionally we use a patch based descriptor using gradient histograms within bins defined by a polar coordinate system.

5.5.1 Hessian Interest Point Detection

The elements of the determinant of the Hessian matrix

$$\det\big(H(x)\big) = \begin{vmatrix} L_{xx}(x) & L_{xy}(x) \\ L_{yx}(x) & L_{yy}(x) \end{vmatrix} = L_{xx}(x) * L_{yy}(x) - L_{xy}(x)^2 \tag{5.1}$$

are the second order partial derivatives L_{xx}, L_{yy} and L_{xy} of the image intensities in a pixel neighborhood. Their use as a saliency measure has originally been proposed in [381], and has since been adopted frequently [43, 391].



		SA2		
	SB1		SB3	
SC0		SC2		SC4
	SD1		SD3	
		SE2		

Figure 5.5 Pixel Locations in Local Neighborhood used for Computing DOH score

While most prior work uses primarily maxima of the DOH score for blob detection, one of the key differences in our method is that we use both maxima and minima in order to find two complementary types of features with a single metric to improve the overall performance for pairwise matching and image ranking. The use of minima in addition to maxima is also supported by [42] who found that negative local minima of the DOH function can augment or even outperform local maxima by providing additional point correspondences.

We compute the DOH metric for each pixel and at various levels of a scale space pyramid [379]. Prior to computing the score, the respective pyramid level gets blurred via convolution with a Gaussian filter kernel of a given size (e.g. $\sigma=4$ pixel). In order to minimize the compute time for the Gaussian blur step we use a recursive implementation of the Gaussian filter proposed in [404] leading to a *20-fold* speedup of this step compared to using convolution with horizontal and vertical tap filters of size 55 pixel. Another advantage of this method is that the compute time is independent of the scale of the Gaussian, and no clipping artifacts occur such as when the tap filter has to be reduced in size for performance reasons.

In order to speed up the detection further only a single pass through each blurred pyramid level is performed and the actual score is computed from the pixel values in a local 5*5 neighborhood (see Figure 5.5) according to

$$\begin{aligned} L_{xx}(x) &= SC0 + SC4 - 2\,SC2; \\ L_{yy}(x) &= SE2 + SA2 - 2\,SC2; \\ L_{xy}(x) &= SD3 - SB3 + SB1 - SD1; \end{aligned} \tag{5.2}$$

Figure 5.6 depicts an example low-resolution pyramid level of an image together with the respective DOH score image. Reddish pixels in the DOH image indicate positive scores occurring in blob-like structures at the given scale of the gray scale image such as the arcade windows of the building. Bluish pixels indicate negative DOH scores, occurring around saddle points of the gray scale image along rather elongated structures such as the balconies separating the different floors of the building.

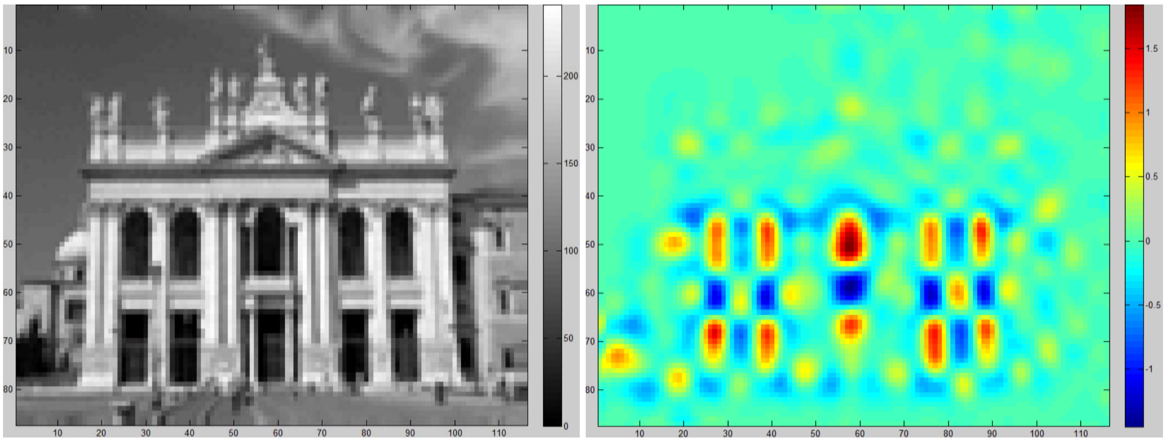


Figure 5.6 Determinant of Hessian - Score (Right) for Sample Pyramid Level (Left) with 116·87 Pixel

This step is followed by 2D non maxima suppression in the resulting score image by comparing pixels against their 8-neighborhood. In contrast to other implementations of hessian detectors [381, 43], and similar to [42] we use both maxima and minima of the DOH score as salient regions.

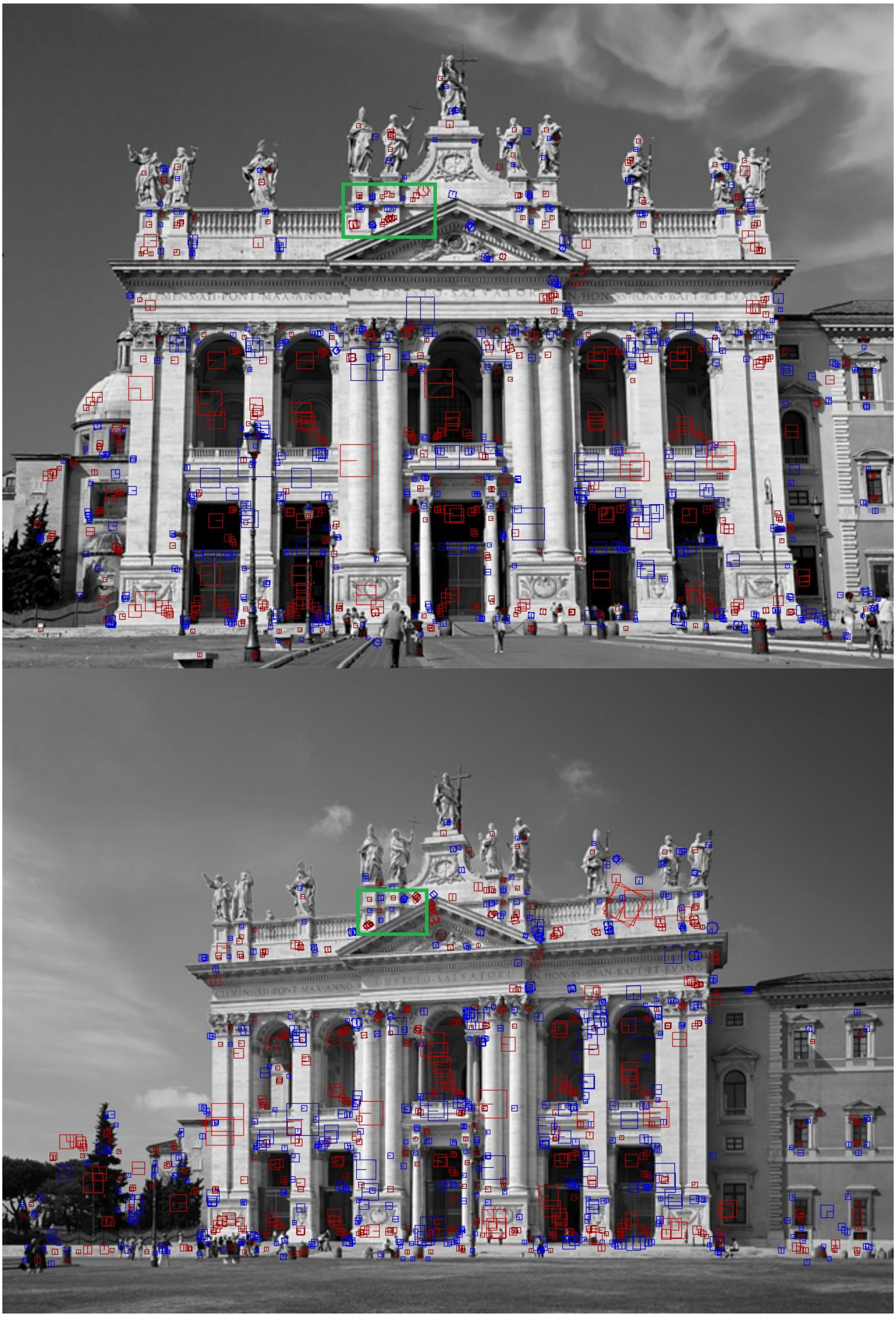


Figure 5.7 Sample Image Regions Detected by DOH Detector for Two Different Images (Basilica of St. John Lateran, Rome); Maxima are Drawn in Red, Minima in Blue; Green Highlight Shows Area with Similar Pattern of Detected Regions;

Optionally the non-maxima suppression can be extended to the scale dimension, by further comparing the score against the next higher and lower pyramid level. However we found that for ranking purposes, this actually reduces the performance, probably as it reduces the total number of features found.

The result of this step is a set of interest point locations in a pixel raster, as well as the pyramid level of the detection, indicating the scale of the feature. Hence the detected point coordinates can be refined by quadratic interpolation either in 2D to obtain sub-pixel accurate point locations, or in 3D to also obtain a more accurate scale estimate [391].

Figure 5.7 illustrates sample points (regions) detected by the described algorithm for a pair of sample images of the Basilica of St. John Lateran in Rome, including points detected as maxima (red) and minima (blue) of the DOH score. The size of the square regions indicates the feature scale, based on the detection pyramid level, the line pointing from the center indicates the feature orientation (which is determined during the descriptor computation explained next - note that some features may have two orientations). Note that as pointed out in [42] the locations of the detected minima around rectangular corners are relatively stable over a range of detection scales, while maxima generally drift. This fact is also visible in Figure 5.7.

Due to the similarity between the two images, maxima and minima of the DOH score occur repeatedly in the same scene location under similar orientations (such as in the region highlighted in green). This behavior is a desired property of an interest point detector. In the following step, a feature descriptor is computed for each detected point, based on its scale and location.

5.5.2 Polar Descriptor Computation

In analogy to many related methods such as SIFT [364], we use a gradient histogram to compute an image descriptor for patches around each of the detected interest points. While SIFT uses normalized orientation histograms within each bin of a 4*4 grid of a patch to compute the descriptor, we instead use a circular region and polar binning.

For this purpose we extract a square patch region around the center of each interest point, at a scale (descriptor level) relative to the detection level of the point. The descriptor level is selected based on an assumed “region size” in the detection level, and a desired “patch size” in the descriptor level. The relation between detection level and descriptor level is one of the tuning parameters of the algorithm. They may be the same or different based on this parameter.

From the extracted image patch we compute x- and y- derivative images, which are represented in 2D arrays of dimension s^2 (e.g. 25·25 pixel). From the x- and y- derivatives we hence compute magnitude and orientation values which are similarly stored in a magnitude array and an orientation array. Note here, that we use the image of the pyramid level before the Gaussian blur was applied to it so the high-frequency content is still present.

In order to de-emphasize magnitude values farther away from the center of the extracted image patch, we apply weight factors following the shape of a Gaussian centered on the region. Outside of a radius r ($r = \frac{s+1}{2}$) the weight factors are set to 0, which means that only magnitude values within a circular region are actually used for the calculation.

Based on the accumulated orientation matrix, we then estimate the “principal orientation” of the region (the most pronounced orientation) from a single orientation histogram with a certain number of bins (e.g. 32). Based on each value in the orientation array, we select a bin in the histogram, in which we accumulate the corresponding weighted magnitude value. To obtain the principal direction from the histogram at a finer granularity than its bin size, we then apply a smoothing filter (e.g. using a 121 kernel) followed by a quadratic fit. The fitted maximum of the orientation histogram is hence used as the principal orientation of the region. Note that in case multiple equivalent maxima are found, we can choose to use only one or several of them. The detected principal orientation is used to “normalize” the way the patch descriptors are computed in the following steps in order to make them rotation invariant. It is also used as the orientation angle of the respective interest point, which may be needed for feature based matching.

After the principal orientation has been determined, we create a 3D histogram of the values in the orientation image weighed by the magnitudes. The histogram is separated into R radial bins, A angle bins and O orientation bins (e.g. $R = 4, A = 8, O = 4$). The radial and angular bins are arranged in a polar coordinate system, while the orientation bins are stacked in a 3rd dimension (See Figure 5.8).

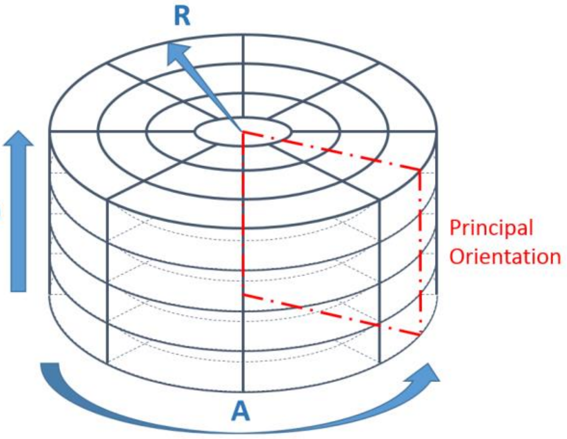


Figure 5.8 Visualization of the 3D Polar Histogram used for the Descriptor Computation for A=8 Angular Bins, O=4 Orientation Bins and R=4 Radial Bins

For each cell in the orientation array, the respective bin in the 3D histogram is selected based on the cell’s radius from the patch center, the normalized (by the principal orientation) angle of the patch as well as the orientation value. The respective bin is incremented by the corresponding magnitude value in the magnitude array.

Since the smallest radial bin is not divided into angle bins, the result is a vector with length N=$O * \left((R-1) * A + 1\right)$, describing the local appearance of the region around the interest point center. The final step is to scale and truncate the orientation histogram so each of the dimensions can be represented as an 8-bit integer. This vector of 8-bit integers is the resulting feature descriptor for the given interest point.

Optionally to reduce the computation times of the following steps, the dimension of the computed descriptors can be reduced by means of a Principal Component Analysis (PCA) to e.g. **32** dimensions [406].

5.6 Image Ranking

A common problem related to image based search is that of retrieving corresponding image matches in a large trellis (up to millions) of indexed images by means of a computationally efficient ranking step. While more recent approaches use different features such as vector of locally aggregated descriptors (VLAD) [291] or Fisher Kernel [407, 408] to aim at even higher computational and memory efficiency, we are following the "bag of features" approach [40, 30]. This method uses a feature quantization to convert high dimensional image descriptors into scalar integer numbers called "visual words" in combination with a standard document retrieval approach using inverted file systems.

In contrast to the prior work mentioned above which used K-Means clustering for feature quantization we apply a nearest neighbor search in a previously trained KD-tree data structure [44] for this purpose. The dataset we used for offline training (Kentucky dataset) is described in 6.1.1. The corresponding visual word for a feature descriptor is derived from the integer index of the leave node containing the nearest neighbor descriptor. A typical vocabulary size is ***V=1,000,000***. Visual words are thus defined by integer numbers in the range between **0** and **V-1**.

5.6.1 Dynamic Image Index[4]

To prepare for image retrieval, an index of the visual words for all index images needs to be generated. For this purpose, we use an inverted file system [409] which is a concept originating from general information retrieval [403], and is used in various applications such as web document search. An inverted file system contains for each word in the dictionary a list of documents containing the specific word. During index generation these lists have to be filled with document indices referring to specific images in the index. During retrieval the visual words from the query image serve as pointers to the inverted file table. For each entry in the inverted document list, and for each visual word from the query image a score is incremented. Finally the documents are sorted according to their scores in decreasing order, and the requested number of top-ranking results is returned by the query.

Several scoring functions are available for determining a weighting score for each document. A common method uses the product of the term frequency of a word in the respective document, and its inverse document frequency in the whole corpus of documents. This weighting is often referred to TF-IDF scoring [40]. Since the inverted document frequency is harder to update in case

[4] The implementation of a dynamic image index described here has been developed by a team of people at Bing Mobile led by David Nistér.

of dynamic deletion of documents after index creation, we decided to reduce the scoring function to only the term frequency, eliminating the inverted document frequency. We found that by using the Boolean term frequency (single count), which is 1 for a document containing a word at least once and 0 otherwise, almost the same retrieval performance can be achieved as with the TF-IDF scoring function. Using the Kentucky score metric proposed by [51] to measure the retrieval performance for the Places dataset described in 6.1.1, the Boolean term frequency method was able to achieve a score of **3.546** (out of 4) compared to **3.568** with the TF-IDF method.

A new document can get ingested in the dynamic index in a similar approach as during index generation by adding its document id to the list of ids for each included visual word. A certain document from the index can be removed by adding the respective document id to a black-list, and ignoring black-listed documents henceforth. Black-listed documents can be eliminated completely whenever the index is re-generated according to some specified schedule, such as once a day or after the removal of a certain number of documents.

5.6.2 Orientation Constrained Ranking

Traditional BOF based image ranking methods make use of quantized visual words that are derived purely from the feature descriptors. The interest point information is usually not directly used during this quantization step.

Due to the information loss caused by quantization, miss-matches between query- and index-features are more likely during retrieval than during nearest neighbor search of the descriptor. Based on our results in [25] we are encouraged to use rotation priors to constrain quantized feature correspondences in a similar way as during pairwise matching. We therefore propose a modified quantization using the orientation information derived from the interest point detector to modify the visual word index and enhance the reliability of quantized matching.

In order to compute the modified visual word indices, the features are sub-divided into *N* equally sized orientation bins which are shifted relative to each other by *W=360°/N*, and have a bin size of *W·f*. The overlap factor *f* is used to avoid quantization of features with similar orientations into distinct bins (e.g. N=8; *f=1.6; W=72°;*). The modified visual word index for a given word is computed as the sum of the original index and a bin offset for bin *n* computed as *n·V*. Features occurring in multiple bins are added multiple times with different bin offsets, thus leading to a higher visual word count for the respective images. It is sufficient to use an overlap factor *f>1* for either the query- or indexed images, while no overlap (f=1.0) is required for the respective other image. Though the results should be similar independent of that choice, note that either the query time or the memory footprint will increase.

To evaluate the rotationally scoped ranking method, we used the same Kentucky score metrics mentioned above, varying the bin count N between 2 and 8 as well as the overlap factor f between 0 and 0.5. As input data, we used a constant set of visual words extracted from the “Places Dataset” described in Section 6.1.1 by using the “DOH features” from Section 5.5. The result of the analysis in Figure 5.9 indicates that rotational scoping with either **N=4** or **N=8** rotational bins performs best. While for 8 rotational bins an overlap factor **f=0.2** is optimal, reducing the number of bins to 4 requires a smaller overlap of **f=0.0** or **f=0.1**. In both cases, the achieved scores of **3.64** are substantially better than the baseline results without rotational scoping (**3.57**).

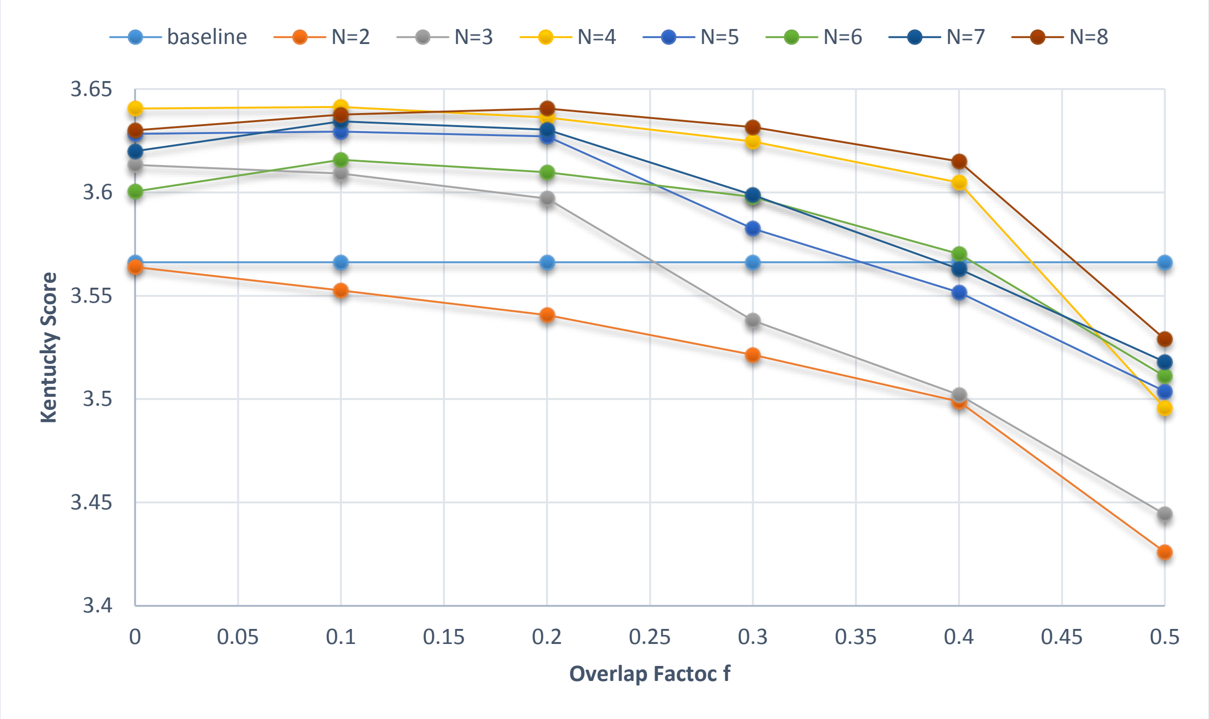


Figure 5.9 Kentucky Score Metric for "Places" Dataset Measured for Different Numbers of Rotational Bins N and Overlap Factors f, and Compared to Baseline

5.7 Pairwise Post-Verification

Results get verified between the query image and each of the top ranked candidate images in a process called post-verification [30], in order to decide which candidates should be accepted and in which ordering they should be returned.

Initially the feature descriptors from both images are used to find match correspondences based on local image similarity by means of a KD-tree algorithm [44]. If prior information about the relative image orientation exists, this prior information can be used to constrain possible matches, thus reducing the likelihood of false matches. Additionally, a ratio between the closest and second closest feature descriptor is tested to filter out ambiguous matches.

Since the pairwise feature-based matching with a ratio test is still likely to create a high number of mismatches, geometric verification of the matched feature pairs is required. Matching images of a 3D-scene usually entails a model describing the epipolar geometry between an image pair. This may be the Fundamental-Matrix described in [24] defining the relation of each point in one image to a line in the other and vice versa. We found that for urban scenes the fundamental matrix provides more is often not discriminative enough to find outliers, and therefore can produce an unacceptable number of false positives. This is also supported by [362] who found that in case of repetitive scene structures miss-matches are often classified as inliers to the Fundamental Matrix.

Hence we decided that for stability reasons it would be better in urban cityscapes to use a more restrictive, homography based model, which basically assumes that the object points must lie on one or more approximately flat surfaces in the 3D-scene. A homography can transform each point in one image into exactly one point in the other image, and hence is more restrictive when filtering out outliers. The homography model is also advantageous for matching objects that are located far from the camera due to the decreasing parallax. The advantage of using homography for location recognition in urban scenes was also noted by the authors of [410, 362, 370]. To estimate the homography efficiently, we use a variant of the RANSAC [45] algorithm described in 5.7.3.

In case of a panoramic candidate image a panorama window gets selected prior to the final geometric verification to determine which sub-window matches most likely to the query image. Once the panorama window has been determined the matched point coordinates within this window can be transformed into a corresponding central-perspective image, and hence the geometric verification can be performed in the same way as for non-panoramic images.

5.7.1 Orientation Constrained Feature Correspondence Search

When matching images using local image features, it is important to solve the correspondence problem with the intention of finding matching image regions across two or more images. This method requires a set of interest points with corresponding feature descriptor vectors, which result from a prior feature extraction step for both the query and candidate image. The query image is described by a set of n descriptor vectors $D^Q = \{D_1^Q, D_2^Q \cdots D_n^Q\}$ in $\boldsymbol{R}^N$ space while the candidate images contains m descriptor vectors $D^C = \{D_1^C, D_2^C, \cdots D_m^C\}$ in the same space.

A naïve solution to this correspondence problem is an exhaustive nearest neighbor search of all descriptors in a candidate image using a given distance metric. While this approach is guaranteed to find the closest solution, it is computationally expensive. A common and more efficient approach used for local image features is to organize the descriptors of single or multiple images in a KD-tree structure [44], which supports approximate nearest-neighbor search for a given query feature in *O(log n)* complexity for n entries in the tree. The method is approximate in that it is not guaranteed to find the nearest neighbor in all cases except if all possible entries are evaluated exhaustively. Nevertheless, for practical image matching applications the nearest neighbor is found efficiently most of the time. In order to reduce the dimensionality of the descriptor space to simplify the nearest neighbor search, and to sort the remaining dimensions in order of decreasing variance, principal component analysis (PCA) can be used on the higher dimensional input descriptors [406].

For many applications it can be assumed, that the query image (e.g. user photo) as well as the candidate image (e.g. panorama) are horizontal to within a certain angular tolerance τ. Therefore it is reasonable to assume that only features with fundamentally similar feature orientations should be matched. Especially for streetside scenes, where repetitive structures and rotationally symmetrical objects (e.g. windows) can occur, this can lead to a better “signal to noise ratio” in terms of the correct versus incorrect matches.

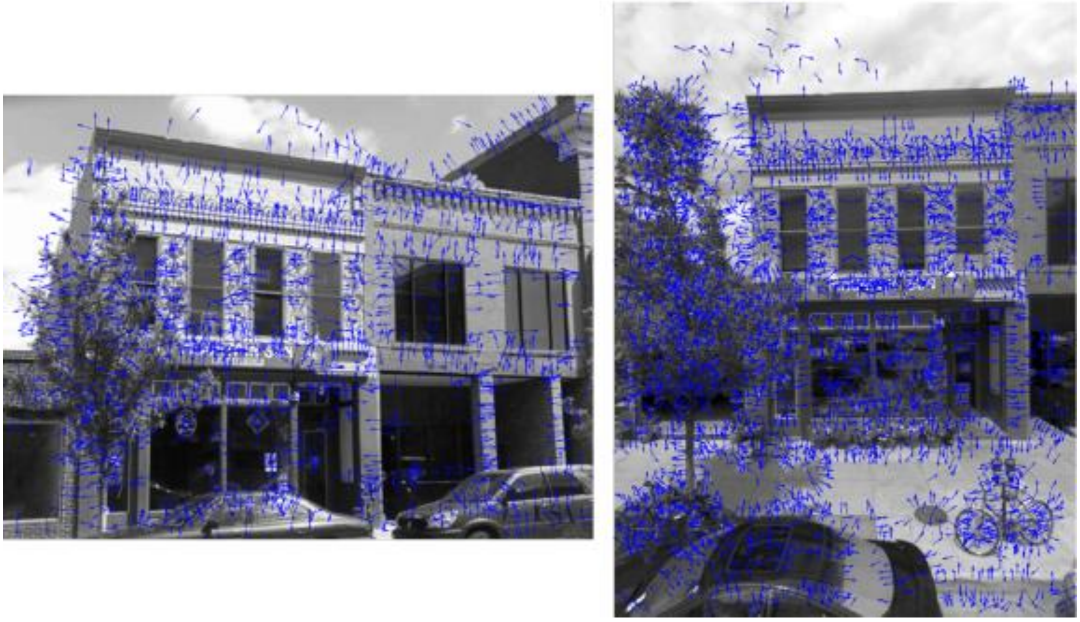


Figure 5.10. Sample pair of query image (Left) and Part of Search Panorama Image (Right) Showing Orientation Vectors of Detected Interest Points.

This method requires that an orientation angle is provided for each interest point during the detection. Figure 5.10 contains a sample image pair, for which the interest point frames F_i are shown as vectors, indicating their locations x_i and y_i, scale s_i and orientation angles φ_i determined during feature extraction.

$$F = (s \cdot \cos\varphi \quad s \cdot \sin\varphi \quad x \quad y)^T \tag{5.3}$$

The proposed method differs from the orientation-binning method described in [25] in that all m descriptors in the candidate image $(D_1^C \,..\, D_m^C)$ are used to generate a single KD-tree and during query, rather than performing queries to multiple KD-trees for different orientation bins. This leads to a simplification of the algorithm and a speed improvement, as only a single KD-tree needs to be generated and queried. Additionally it also produces more accurate results as the orientation thresholding can be done per feature rather than per bin.

During standard KD-tree search, the nodes of the KD-tree are traversed starting at the root node, and following either the left or right path of the given node, depending on which side of a hyper-plane the respective descriptor dimension lies. Hyper-planes are defined as thresholds for a given descriptor dimension which are determined during tree-generation as the median of the data for the respective dimension. This process is repeated until a certain level of the tree is reached, at which all indexed descriptors belonging to a node (leave-node) are compared exhaustively to find the closest one. For this purpose a distance metric such as the Euclidean distance or the cosine distance between the query descriptor and the candidate descriptor can be used in order to determine the closest descriptor. As leave-nodes often contain only a small number of descriptors (e.g. b=64), the actual closest descriptor may lie in a different leave node. Therefore multiple leave nodes may have to be checked in a recursive manner up to a certain maximum (e.g. k=256), and depending on whether it is deemed possible that any other descriptor exists which is closer than the currently best candidate. In case of the Euclidean feature distance, this termination condition result from intersecting a hyper-sphere with a radius of the currently closest distance with the currently active hyper-plane. If the hyper-sphere and hyper-plane don't intersect it is clear that no closer data point lies on the other side of the hyper-plane and the algorithm can terminate [44].

$$\Phi^{Q} = \left\{\varphi_1^Q, \varphi_2^Q \cdots \varphi_n^Q\right\}; \qquad \Phi^{C} = \left\{\varphi_1^C, \varphi_2^C, \cdots \varphi_m^C\right\}; \tag{5.4}$$

To decide whether a candidate descriptor D_j^C is a valid match for a query descriptor D_i^Q under rotational scoping the orientation angles of both interest points (φ_i^Q and φ_j^C) have to be compared to see whether the angular difference is less than or equal to the angular tolerance $\boldsymbol{\tau}$ and reject them otherwise. This can be done as part of the exhaustive feature comparison step of the KD-tree algorithm, given that the respective feature orientation angles are available at this point. Since computing the distance between two descriptor vectors is substantially more expensive than checking the angular difference, this leads to a significant reduction of the search time. As this method still requires *k·n* angular comparisons for *n* descriptors in the query image, we propose a more optimal alternative. For this purpose, the features in the candidate and query images first have to be sorted according to the orientation angle, for example from -180° up to 180°. Further, a binary vector

$$S^C = \left\{s_1^C, s_2^C, \cdots s_m^C\right\}; \tag{5.5}$$

is required, which is initialized to:

$$s_j = \begin{cases} TRUE\ iff\ \varphi_i^Q \hat{-} \varphi_1^C \leq \tau\,; \\ FALSE\ otherwise; \end{cases} \tag{5.6}$$

The "angle difference" operator $\varphi_a \hat{-} \varphi_b$ is defined as the smaller of the two angles between two vectors with the orientations φ_a and φ_b. While performing the nearest neighbor search for each query descriptor D_i^Q using the KD-tree algorithm described above the corresponding binary value s_j has to be verified for each candidate descriptor in the corresponding leave nodes. Since the features are sorted according to the feature orientation, only a few consecutive elements of the binary mask have to be updated for each new query feature *i* depending on whether the orientation condition is fulfilled, resulting in a maximum number of *2·(m+n)* angular comparisons.

To reject ambiguous matches early in the process, a ratio test is performed, comparing the feature distances (in feature space) $\rho_{closest}$ and ρ_{second} between the query descriptor and the closest and second closest descriptor from the index image.

$$\rho_{closest} = \left\|D_{query}^Q - D_{closest}^P\right\| \;\; ; \;\; \rho_{second} = \left\|D_{query}^Q - D_{second}^P\right\| \tag{5.7}$$

According to

$$iff\ \frac{\rho_{closest}}{\rho_{second}} \begin{cases} > \boldsymbol{\vartheta}: & \boldsymbol{Reject\ Match} \\ \leq \boldsymbol{\vartheta}: & \boldsymbol{Accept\ Match} \end{cases} \qquad (5.8)$$

a feature pair is rejected if the ratio is above some threshold ϑ (e.g. **0.8,** see [364]), and accepted otherwise.

5.7.2 Panorama Window Selection

In case of a panoramic candidate image, a sub-window needs to be selected before the actual geometric verification. This selection process is based on the feature correspondences returned by the algorithm described above. The interest point frames F^Q and F^C for a sample image pair are visualized Figure 5.11, where the panorama image format has been extended by 90 degrees to the right in order to represent the cyclic nature of this image format.

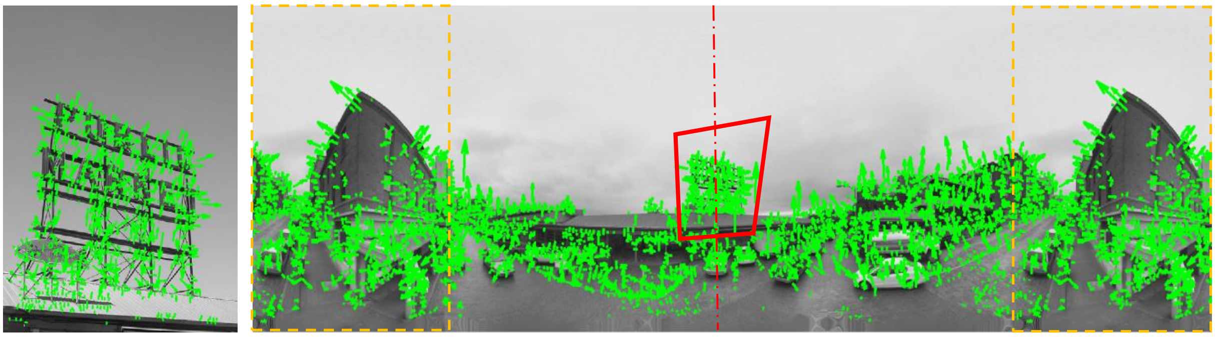


Figure 5.11 Input Image with Feature Frames (Left); Extended Panorama Image (Right)

The interest points and descriptor based matches occurring within the leftmost 90° window yellow box) are replicated at the right, and the same geometric verification method described in section 5.75.7.3 is used to find the best fitting homography model between the feature matches (See Figure 5.11, right). The 90 degree window centered on the median panorama longitude of the inlier matches (red chain-dotted line) is hence used for the remaining post-verification steps. Figure 5.12 shows the selected sub-window from the spherical panorama image as well as a version of the same image warped into a virtual camera view.

All following explanations of the algorithm are based on the assumptions, that the panorama sub-window is known, and that only the features from within this sub-window are used.

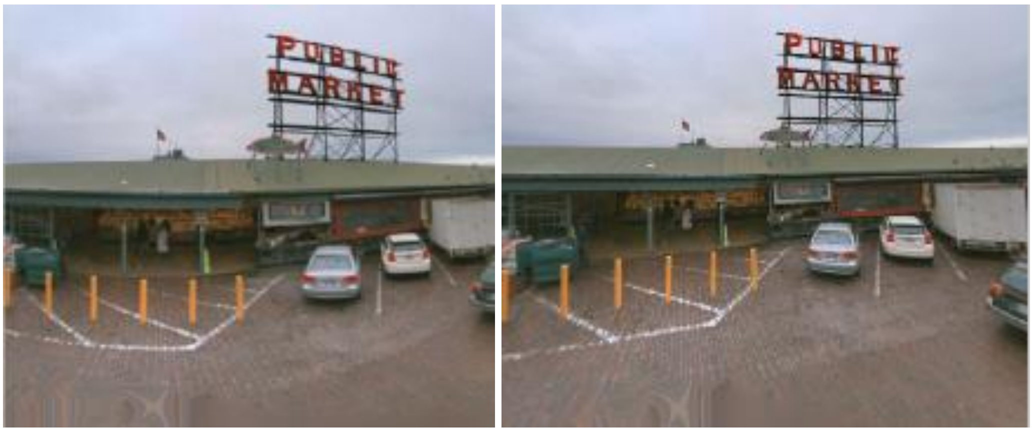


Figure 5.12. Selected Panorama Sub-Window (Left); Unwarped Sub-Window (Right);

5.7.3 Geometric Verification using One-Point Similarity and Projective RANSAC[5]

Most image matching frameworks geometrically validate putative candidate matches obtained during a pairwise correspondence search. In our previous work [25], we showed that for urban scenes consisting primarily of planar surfaces, agreement of the matches to a homography model or projective transform is often sufficient to support the decision whether a candidate image is an acceptable match or not. The homography transformation H from a 2D point in homogeneous coordinates $X = (x \quad y \quad 1)^T$ to a point $X' = (x' \quad y' \quad 1)^T$ is defined as

$$\begin{pmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{1} \end{pmatrix} \cong \boldsymbol{H} * \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{1} \end{pmatrix} = \begin{bmatrix} \boldsymbol{h_{11}} & \boldsymbol{h_{12}} & \boldsymbol{h_{13}} \\ \boldsymbol{h_{21}} & \boldsymbol{h_{22}} & \boldsymbol{h_{23}} \\ \boldsymbol{h_{31}} & \boldsymbol{h_{32}} & \boldsymbol{1} \end{bmatrix} \cdot \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{1} \end{pmatrix} \qquad (5.9)$$

where h_{11} through h_{32} are the variable elements of the homography matrix H. As the homography is defined for homogeneous coordinates, scaling of the matrix doesn't affect the results. Therefore the 9th parameter (h_{33}) can be set to 1, reducing the number of degrees to 8. Figure 5.13 shows the effect of applying a homography to an image in order to align it to a planar building surface of a streetside image.

[5] The one-point similarity and projective RANSAC method described here has been developed by a team of people at Bing Mobile led by David Nistér.

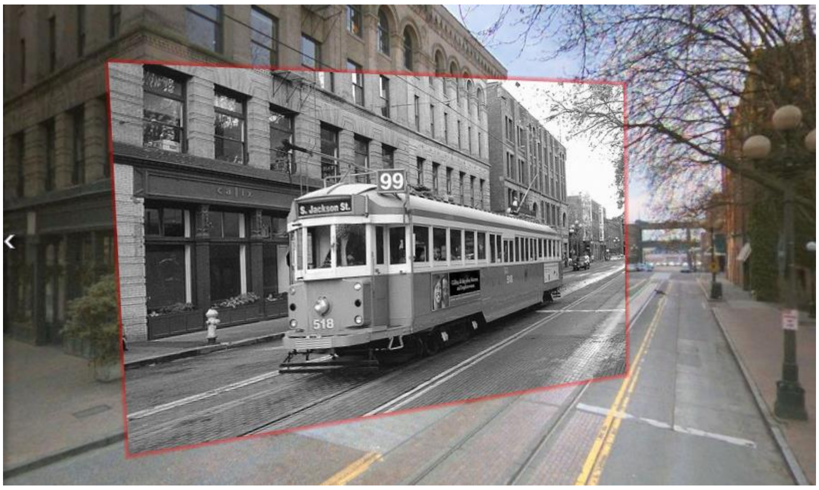


Figure 5.13 Image of Historic Tramway in Downtown Seattle, which has been transformed using a Homography to fit to the Background Streetside Image

A frequently used method for robust geometric verification is the Random Sample Consensus (RANSAC) algorithm [45], which is an iterative way of finding the parameters for a given mathematical model (e.g. homography) from a set of data points. The method is robust in that it can deal with a certain percentage of model outliers in the data, which are common for descriptor based image correspondence search. During each iteration, a sufficiently large random subset of the data is used to estimate a hypothesis of the model parameters, and the hypothesis is verified using the whole dataset. A score defining the degree of data agreement to the model is computed, and eventually the model with the best score is chosen. Examples for typically used scoring methods are the inlier count or a cost function based on the average deviation from the model. While determining the inlier count usually requires hard thresholds for each data point, cost functions allow a fuzzier decision metric such as the reprojection error for a given image point. As the selection process leading to each hypothesis is random, the method is non-deterministic and it finds the solution only with a certain probability p [45]. This probability depends on the expected inlier ratio w, the number of data points n needed to estimate all model parameters, and the number of RANSAC iterations k according to

$$1 - p = (1 - w^{n})^{k} \tag{5.10}$$

As the 9th parameter (h_{33}) of the homography is set to 1, only 8 parameters have to be solved for each hypothesis. Often this is achieved by using the x and y coordinates from *n=4* point correspondences which provide sufficiently many equations to solve the model parameters. The probability of finding the correct solution depends strongly on the number of data points required as well as the inlier ratio. For an inlier ratio of w=20% and a fixed number of k=1,000 RANSAC iterations, the probability to find the correct solution is hence 79.8% while for an inlier ratio of w=10% it is merely 9.5%.

In order to improve the chances of finding the correct solution, or to reduce the number of iterations required for the same probability, we therefore follow a different approach. Taking advantage of the fact that the homographies occurring in typical streetside scenes often exhibit limited perspective distortion we initially solve for a similarity transform using a 1-point RANSAC algorithm with large error margins, followed by a robust optimization of the similarity as well as the homography parameters.

$$\begin{pmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{1} \end{pmatrix} = \begin{bmatrix} \boldsymbol{s_0} & \boldsymbol{s_1} & \boldsymbol{s_2} \\ -\boldsymbol{s_1} & \boldsymbol{s_0} & \boldsymbol{s_3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} * \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{1} \end{pmatrix} \tag{5.11}$$

$$S = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \tag{5.12}$$

Initially a RANSAC step is used to find the best model hypothesis for the 4 parameters of the similarity transform *S*. In addition to the above equation, each correspondence provides a second equation

$$\begin{pmatrix} \boldsymbol{ox_2} \\ \boldsymbol{oy_2} \end{pmatrix} = \begin{bmatrix} \boldsymbol{s_0} & \boldsymbol{s_1} \\ -\boldsymbol{s_1} & \boldsymbol{s_0} \end{bmatrix} * \begin{pmatrix} \boldsymbol{ox_1} \\ \boldsymbol{oy_1} \end{pmatrix} \tag{5.13}$$

describing the transformation of the scale and orientation vectors $(ox_1\ oy_1)^T$ and $(ox_2\ oy_2)^T$ of the two feature regions. Presuming that reliable scale and orientation information are provided by the feature detector for each region only n=1 correspondence is required in order to solve for the 4 similarity parameters. Hence it is essentially guaranteed that the correct model is found by RANSAC even for a much smaller number of iterations. For w=0.1 and k=100 the probability of finding the correct solution is 99.998%. For a given point correspondence $[ox_1\ oy_1\ x_1\ y_1]^T$ and $[ox_2\ oy_2\ x_2\ y_2]^T$ the parameters s_0 and s_1 can be solved by transforming Eq. (5.13) into the form

$$\underbrace{\begin{bmatrix} \boldsymbol{ox_1} & \boldsymbol{oy_1} \\ -\boldsymbol{oy_1} & \boldsymbol{ox_1} \end{bmatrix}}_{\boldsymbol{A}} * \underbrace{\begin{pmatrix} \boldsymbol{s_0} \\ \boldsymbol{s_1} \end{pmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{pmatrix} \boldsymbol{ox_2} \\ \boldsymbol{oy_2} \end{pmatrix}}_{\boldsymbol{b}} \tag{5.14}$$

and solving for x via multiplication by A^{-1}. Hence s_2 and s_3 can be determined using

$$\begin{pmatrix} \boldsymbol{s_2} \\ \boldsymbol{s_3} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x_2} \\ \boldsymbol{y_2} \end{pmatrix} - \begin{bmatrix} \boldsymbol{s_0} & \boldsymbol{s_1} \\ -\boldsymbol{s_1} & \boldsymbol{s_0} \end{bmatrix} * \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{y_1} \end{pmatrix} \tag{5.15}$$

For each hypothesis S a cost function $C(S)$ is evaluated by applying the hypothesis on all i point correspondences and computing a robustified cost C_{Cen} for the reprojection error as well as costs C_{Ori} and C_{Sca} for discrepancies in orientation and scale between the putative matches.

$$\begin{aligned} \boldsymbol{C(S)} &= \boldsymbol{C_{Cen}} + \boldsymbol{C_{Ori}} + \boldsymbol{C_{Sca}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \left(\boldsymbol{C^i_{Cen}} + \boldsymbol{C^i_{Ori}} + \boldsymbol{C^i_{Sca}}\right) \\ &= \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \left(\boldsymbol{R}\left(\boldsymbol{e^i_{Cen}}, \boldsymbol{sc_{Cen}}\right)^2 + \boldsymbol{R}\left(\boldsymbol{e^i_{Ori}}, \boldsymbol{sc_{Ori}}\right)^2 + \boldsymbol{R}\left(\boldsymbol{e^i_{Sca}}, \boldsymbol{sc_{Sca}}\right)^2 \right) \end{aligned} \tag{5.16}$$

In order to reduce the effect of model outliers a robustified cost function is used, based on the robustifier $R(e, sc)$ which is also plotted as a function of e in Figure 5.14.

$$R(e, sc) = re = \frac{e^2}{e^2 + sc^2} \tag{5.17}$$

The scale parameters *sc* for each error measure have to be selected large enough to compensate for the modelling error from using a similarity transform. Typical scale factors for normalized image coordinates in the range 0..1 are $sc_{Sca} = 0.2$, $sc_{Ori} = 0.39$ and $sc_{Sca} = 0.5$.

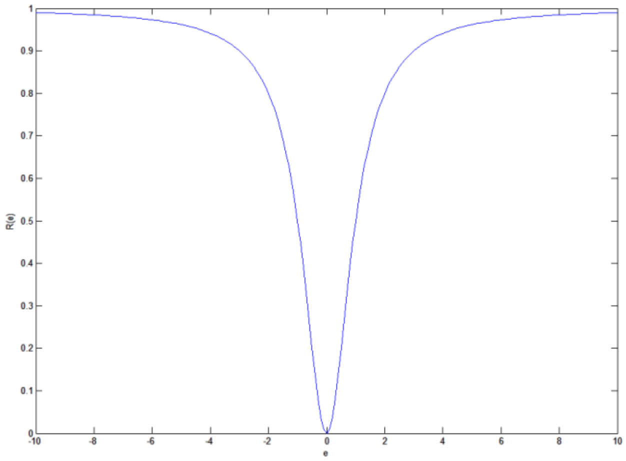


Figure 5.14 Robustifier Function R(e,sc) for sc=1.0

The center error measure for a given pair of points is the reprojection error between the transformed coordinates of point 1 and the coordinates of point 2:

$$e_{Cen} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} = \begin{bmatrix} s_0 & s_1 \\ -s_1 & s_0 \end{bmatrix} * \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} s_2 \\ s_3 \end{pmatrix} - \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \tag{5.18}$$

To measure the scale and orientation errors a transformation T

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \frac{1}{\sqrt{ox_2 + oy_2}} * \begin{bmatrix} ox_2 & oy_2 \\ -oy_2 & ox_2 \end{bmatrix} \tag{5.19}$$

is applied on both the reprojected scale/orientation vector from image 1 as well as the scale/orientation vector from image 2, such that the latter is transformed into a unity vector. The deviations in x and y between the transformed first vector and the unity vector constitute error measures e_{Sca} for scale and e_{Ori} for orientation discrepancies according to

$$\begin{aligned} \begin{pmatrix} e_{Sca} \\ e_{Ori} \end{pmatrix} &= T * \left(\begin{bmatrix} s_0 & s_1 \\ -s_1 & s_0 \end{bmatrix} * \begin{pmatrix} ox_1 \\ oy_1 \end{pmatrix} - \begin{pmatrix} ox_2 \\ oy_2 \end{pmatrix} \right) \\ &= T * \begin{bmatrix} s_0 & s_1 \\ -s_1 & s_0 \end{bmatrix} * \begin{pmatrix} ox_1 \\ oy_1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned} \tag{5.20}$$

The cost function in Eq. (5.16) can be evaluated for all *k* iterations of the RANSAC algorithm, and the hypothesis S^0 with the lowest cost is chosen as the starting point for the following optimization. If the number of correspondences *n* is small enough, it may be even feasible to set *k*=*n*, thus evaluating all possible hypotheses exhaustively.

We use the Levenberg-Marquardt (LM) method [411, 412] for numerical optimization in order to fit a more accurate similarity model to the data than that obtained by the above RANSAC algorithm. Iterative optimization updates the model *S* by some small amount δ_S in order to converge at a local minimum of the cost function. As this cost function is non-linear with respect to *S*, it can be locally approximated by a quadratic Taylor expansion

$$C(S^{t+1}) \approx C(S^t + \delta_s) \approx C(S^t) + \nabla_C(\mathrm{S}^t) * \delta_s + \frac{1}{2}{\delta_s}^T * H_C(S^t) * \delta_s \qquad (\,5.21\,)$$

in order to predict its value after the update where $\nabla_C(\mathrm{S}^t)$ is the gradient,

$$\nabla_C = \left[\frac{\partial C}{\partial S_0} \cdots \frac{\partial C}{\partial S_3}\right] = 2 * \sum_{i=1}^{n}\left({J_{re}^i}^T * re^i\right) = 2 * \left(\sum_{i=1}^{n}\left({J_{re_{Cen}}^i}^T * re_{Cen}^i + {J_{re_{Sca}}^i}^T * re_{Sca}^i + {J_{re_{Ori}}^i}^T * re_{Ori}^i\right)\right) \qquad (\,5.22\,)$$

of C at S^t and $H_C(S^t)$ is the Hessian,

$$H_C = \begin{pmatrix} \frac{\partial^2 C}{\partial S_0 \partial S_0} & \cdots & \frac{\partial^2 C}{\partial S_0 \partial S_3} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 C}{\partial S_3 \partial S_0} & \cdots & \frac{\partial^2 C}{\partial S_3 \partial S_3} \end{pmatrix} = 2 * \sum_{i=1}^{n}\left({J_{re}^i}^T * J_{re}^i\right) = 2 * \left(\sum_{i=1}^{n}\left({J_{re_{Cen}}^i}^T * J_{re_{Cen}}^i + {J_{re_{Sca}}^i}^T * J_{re_{Sca}}^i + {J_{re_{Ori}}^i}^T * J_{re_{Ori}}^i\right)\right) \qquad (\,5.23\,)$$

of C at S^t. Both are computed as the sum of the individual elements for all n data points where re_{Cen}^i, re_{Ori}^i and re_{Sca}^i are the individual robustified errors and $J_{re_{Cen}}^i$, $J_{re_{Ori}}^i$ and $J_{re_{Sca}}^i$ are their Jacobians relative to the elements of S. The latter are the products of the Jacobians for the robustifiers and the Jacobians of the non-robustified errors relative to the elements of S.

$$J_{re_{Cen}}^i = J_{R_{Cen}}^i * J_{e_{Cen}}^i; \quad J_{re_{Sca}}^i = J_{R_{Sca}}^i * J_{e_{Sca}}^i; \quad J_{re_{Ori}}^i = J_{R_{Ori}}^i * J_{e_{Ori}}^i; \qquad (\,5.24\,)$$

$$J_{R_{Cen}}^i = \frac{1}{\sqrt{{sc_{Cen}}^2 + {e_x^i}^2 + {e_y^i}^2}}\begin{bmatrix} 1 - {re_x^i}^2 & -re_x^i * re_y^i \\ -re_x^i * re_y^i & 1 - {re_y^i}^2 \end{bmatrix}; \quad J_{e_{Cen}}^i = \begin{bmatrix} x_1 & y_1 & 1 & 0 \\ y_1 & -x_1 & 0 & 1 \end{bmatrix} \qquad (\,5.25\,)$$

$$J_{R_{Sca}}^i = \frac{1 - {re_{Sca}^i}^2}{\sqrt{{sc_{Sca}}^2 + {e_{Sca}^i}^2}}; \quad J_{e_{Sca}}^i = [T_{11} \quad T_{12} \quad 0 \quad 0] \qquad (\,5.26\,)$$

$$J_{R_{Ori}}^i = \frac{1 - {re_{Ori}^i}^2}{\sqrt{{sc_{Ori}}^2 + {e_{Ori}^i}^2}}; \quad J_{e_{Ori}}^i = [T_{12} \quad -T_{11} \quad 0 \quad 0] \qquad (\,5.27\,)$$

By substituting Eq. (5.22) and (5.23) the cost function becomes

$$C(S^t + \delta_s) \approx C(S^t) + 2 * \sum_{i=1}^{n}\left({J_{re}^i}^T * re^i\right) * \delta_s + {\delta_s}^T * \sum_{i=1}^{n}\left({J_{re}^i}^T * J_{re}^i\right) * \delta_s \qquad (\,5.28\,)$$

In order to determine the update δ_S for the current iteration, the derivative with respect to δ_S can be computed, set to 0 and rewritten to the form A·x=b. This is a linear system of equations that can be solved for δ_S e.g. using the Cholesky decomposition [413].

$$\frac{\partial C(S^t + \delta_s)}{\partial \delta_s} \approx 2 * \sum_{i=1}^{n}\left({J_{re}^i}^T * re^i\right) + 2 * \sum_{i=1}^{n}\left({J_{re}^i}^T * J_{re}^i\right) * \delta_s = 0 \qquad (\,5.29\,)$$

$$\underbrace{\sum_{i=1}^{n}\left({J_{re}^i}^T * J_{re}^i\right)}_{A} * \underbrace{\delta_s}_{x} = \underbrace{\sum_{i=1}^{n}\left({J_{re}^i}^T * re^i\right)}_{b} \qquad (\,5.30\,)$$

To improve the speed of the convergence in the presence of small gradients, A is replaced by the sum of itself and its diagonal, scaled by a damping factor $(1 + \lambda)$ as proposed by Marquardt. The value of λ is adjusted per iteration based on whether the cost function actually decreased or not.

Updates to S are only applied in case of an improvement. We start with a λ value of 0.001 which is multiplied by a factor 0.1 in case of an improvement and by a factor of 100.0 in case of no improvement.

$$\underbrace{\left(\sum_{i=1}^{n}\left({J_{re}^{i}}^{T} * J_{re}^{i}\right) + (1+\lambda) * \operatorname{diag}\left(\sum_{i=1}^{n}\left({J_{re}^{i}}^{T} * J_{re}^{i}\right) * J_e\right)\right)}_{A} * \underbrace{\delta_S}_{x} = \underbrace{\sum_{i=1}^{n}\left({J_{re}^{i}}^{T} * re^{i}\right)}_{b} \tag{5.31}$$

The optimization is terminated either if the cost improvement is below a threshold indicating that convergence is reached, if λ exceeds a maximum (e.g. 10^{10}) or after a maximum number of iterations (e.g. 5).

In order to refine the transformation further, a second optimization can be performed to find an optimal homography model (8 parameters) based on the same point correspondences. For this purpose, the similarity transform needs to be converted into a homography by setting *H=S*. The error measures for the point center, scale and orientation in (5.18) and (5.20) are updated according to the homography model.

$$e_{Cen} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} - \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \overbrace{\frac{1}{h_{31} * x_1 + h_{32} * y_1 + 1}}^{o} * \begin{pmatrix} h_{11} * x_1 + h_{12} * y_1 + h_{13} \\ h_{21} * x_1 + h_{22} * y_1 + h_{23} \end{pmatrix} - \begin{pmatrix} \boldsymbol{x_2} \\ \boldsymbol{y_2} \end{pmatrix} \tag{5.32}$$

$$\begin{pmatrix} e_{Sca} \\ e_{Ori} \end{pmatrix} = T * \left(\begin{matrix} \overbrace{h_{11} * ox_1 + h_{12} * oy_1 - x_p \overbrace{\left(h_{31} * ox_1 + h_{32} * oy_1\right)}^{r}}^{p} \\ \underbrace{h_{11} * ox_1 + h_{12} * oy_1 - x_p\left(h_{31} * ox_1 + h_{32} * oy_1\right)}_{q} \end{matrix} \right) - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5.33}$$

The Jacobians for the unrobustified errors in Eq. (5.25), (5.26) and (5.27) have to be updated accordingly, while the Jacobians of the robustifiers remain the same.

$$J_{e_{Cen}}^{i} = o * \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 * x_p & -y_1 * x_p \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 * y_p & -y_1 * y_p \end{bmatrix} \tag{5.34}$$

$$J_{Dir}^{i} = o * \left(\begin{bmatrix} ox_1 & oy_1 & 0 & 0 & 0 & 0 & -ox_1 * x_p - o * p * x_1 & -oy_1 * x_p - o * p * y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -ox_1 * y_p - o * q * x_1 & -oy_1 * y_p - o * q * y_1 \end{bmatrix} - r * J_{e_{Cen}}^{i} \right) \tag{5.35}$$

$$\begin{bmatrix} J_{e_{Sca}}^{i} \\ J_{e_{Ori}}^{i} \end{bmatrix} = \frac{1}{{ox_2}^2 + {oy_2}^2} \begin{bmatrix} ox_2 & oy_2 \\ -oy_2 & ox_2 \end{bmatrix} \cdot J_{Dir}^{i} \tag{5.36}$$

Since the homography models the actually occurring geometric transformations better than the similarity transform used, the scale factors for the robustifier may be slightly reduced.

Finally the point correspondences can be separated into inliers and outliers of the computed homography model based on thresholds for the different error metrics. An inlier match is defined as a point correspondence for which the actual residual errors are below defined thresholds th_{Cen} for center position, th_{Sca} for scale and th_{Ori} for orientation. Typical values are $th_{Cen} = 1.1 * sc_{Cen}$, $th_{Sca} = 1.2 * sc_{Sca}$ and $th_{Ori} = 1.4 * sc_{Ori}$. In order to count only independent inlier matches belonging to different image regions in both images, the regions are first sorted by their size, and an overlap test is performed for each new region. If it is included in a previous inlier region, the match is rejected despite its error metrics, otherwise it is accepted. This leads to a more discriminative inlier score than if all inliers were counted despite referring to the same image regions.

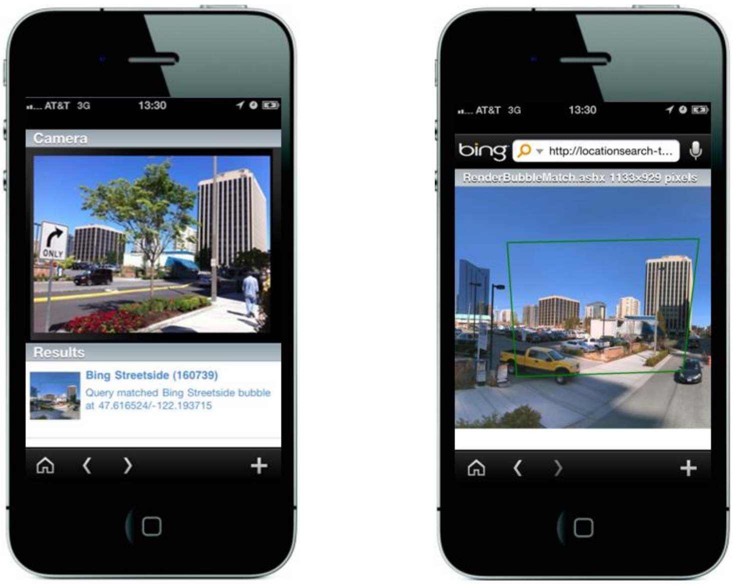
5.8 Summary

In this chapter, we have presented a workflow addressing the problem of location recognition by means of image retrieval in a city scale corpus of up to millions of geospatial images. The index supports both systematically captured human scale panoramic imagery, as well as user created content from CPC, mobile queries or other sources. It comprises an improvement in scalability over our prior work in this area [25] and can be used for various applications in areas such as internet mapping, robot localization, SLAM, augmented reality or landmark recognition.

In the following chapter we will evaluate the presented workflow based on various applications related to internet mapping and location based search.

6 Location Search Applications and Evaluation

After presenting the workflow in the previous chapter, we evaluate the presented workflow and sub-components of it, and we describe two applications in the context of internet mapping. We first define an evaluation framework for measuring the performance of image features for retrieval and pairwise image matching (Section 6.1), and compare the proposed feature extraction method with a range of alternatives. We further investigate ways for speed optimization of the presented feature extraction method, leading to a **16-fold** speedup from **400 ms** to **25 ms** for a **VGA** image without a significant degradation of their matching performance.



Bing Streetside (160739)
Query matched Bing Streetside bubble at 47.616524/-122.193715

RenderBubbleMatch.ashx 1133x929 pixels

Figure 6.1 Sample Query Image from Smartphone (Left); Matched Bing Maps Streetside Panorama (Right) with Detected Bounding Polygon Showing Overlap with Query Image;

We then revisit the application of matching user photography from CPC to human scale panorama images described in [25] in Section 6.1.6. By using the improved workflow presented in Chapter 4.6, we could achieve improvements of the recall rate from **59.5%** to **73.3%** while retaining the same low false positive rate of **0.5%**. Additionally matches to human scale panoramas can be achieved in real-time with from typical query times around **5-10 seconds** compared to several minutes with the previous workflow. Figure 6.1 features an example of a user query image captured using a smartphone application (left) with a list of query results below the image. One of the query results showing a matching streetside image with a highlighted bounding region indicating the detected frustum of the query image is also shown (right). A second application (Section 6.2.3) addresses the problem of improving business-geocoding by means of image matching, in order to increase the quality of business data on internet mapping sites. By using the same matching workflow, we can achieve a recall rate for storefront images of **75.8%**, which comprises a significant improvement over **56.3%** achieved using the previous method [25]. Business locations get updated by **40 m** on average, to within a required **10 m** radius from the actual business.

Finally (Section 6.4) we evaluate the effect of reducing the data volumes transmitted over wireless networks either by image compression or down-sampling, on the quality of matching user-photos to human scale data. We found that a **10-fold** reduction in file size from **188 kB** to **19 kB** per image led to only minor changes in both true- and false positive rates.

6.1 Feature Evaluation

When designing an image retrieval or matching system using local image features, decisions have to be made, which of a large variety of interest point detectors or feature descriptors to use. Since different applications such as image retrieval, SFM or wide baseline stereo matching often differ in the exact way the features are used, the requirements for the feature stack vary. For example, depending on the degree of variance in the images to be matched, such as whether the objects have mostly planar or 3-dimensional structures, how much perspective distortion is expected, or to what degree the scene itself has changed, different choices may be optimal. When dealing with different applications such as Photosynth [163], image based product search [298] or location recognition [354], our intuition suggests that no single choice of a detector or descriptor is optimal for all applications. Therefore it is generally advantageous to validate the choice using representative datasets for each application as well as meaningful metrics. Once a set of components has been chosen, the same datasets and metrics can also be used to validate how changes to parameters controlling the different components or speed optimizations influence the quality performance of an algorithm.

In this work we do not aim at providing a comprehensive performance comparison of different interest point detectors and descriptors, as this would be biased by implementation details of methods available to us. Instead, we are proposing a method for comparing various options and settings for the applications of image retrieval in large databases, and pairwise post-verification of candidate matches obtained by ranking. While other applications such as SFM use the same type of image features, they often require different kinds of metrics and datasets for comparison, which are outside of the scope of this work.

6.1.1 Datasets

In order to compare the performance of different interest point detectors as well as feature descriptors for ranking a large set of images as well as for matching of image pairs, we used primarily two datasets. The first dataset was originally created and publicly shared by the University of Kentucky [51] to evaluate the image retrieval work published in [30] and to serve as a reference for general recognition of known objects. This dataset, referred to in the following as

“Kentucky dataset” includes **10,200** non-“geo” VGA images, consisting of **2,550** individual quadruples. Within each quadruple, the same object or scene has been captured under certain variations in perspective or lighting.

Since the Kentucky dataset contains mostly images of small objects, and only very few locations, we decided to capture a new dataset in the same schema which was more representative of the application we were mostly concerned with – geo-location recognition. Additionally, we wanted to remove any uncertainties caused by the fact that the vocabulary training for the visual word based ranking is generally done using the Kentucky dataset. This second dataset, hence called “Places dataset”, consists of **3,444** mobile phone images resized to VGA resolution. It is organized in **861** quadruples captured at the same location with a similar degree of variability in the camera poses. However the lighting was not varied. Examples from both datasets are shown in Figure 6.2.

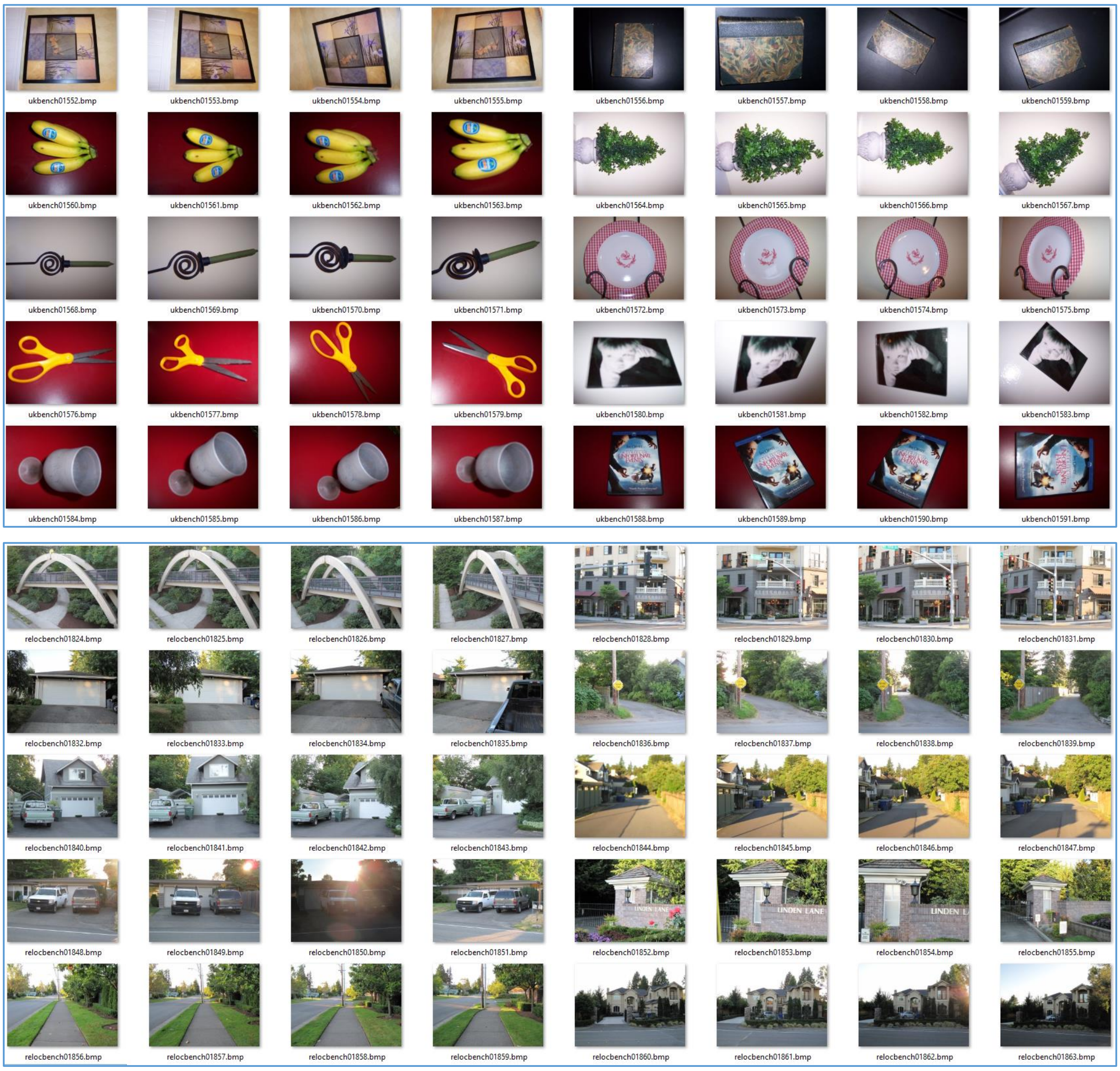


Figure 6.2 Example Images from the Kentucky Dataset (Top 5 Rows) and Places Dataset (Bottom 5 Rows).

6.1.2 Metrics

We chose to use different metrics for ranking a large set of images and for the pairwise matching problem. The metric used for measuring the performance of an image retrieval system proposed by [51], is the average number of correct entries in the top 4 ranked results when querying a complete index with each image individually. We hence call this ranking metric "Kentucky score", if evaluated on the Kentucky dataset, and "Places score" if evaluated on the Places dataset. Since the images are organized with increasing indices in quadruples, the evaluation if a match is correct is trivial by checking whether the query index modulo 4 equals the result index modulo 4.

In addition to measuring the performance of features for ranking images, we were also interested in how well the same features could be used in a pairwise image matching problem, such as the post-verification component of the location matching system described in Section 5.7. For this purpose we applied the post-verification algorithm including KD-tree ranking [44] and RANSAC [45] on the 8 top ranked images after the ranking step, making sure that the 4 correct matches were always included. If they were not, we replaced the lowest ranked mismatch with the missing match. Note that the 4 non-matches from this ranking are harder than a random choice, as they were the result of ranking based on visual similarity. Computing these 8 pairwise matches for all images in the set, the true positive rate (TPR) and false positive rate (FPR) can be computed. As a metric we used the numerical integral of the ROC curve [52], obtained by varying the inlier threshold for the post-verification step. While for certain applications it may be more feasible to compare the actual TPR values for a defined FPR value as a quality metric, we feel that for general comparison the ROC-integral is a more comprehensive metric as it covers multiple different configuration settings. In order to minimize the dependency of the post-verifier metrics on changes in the ranking results, we used a fixed set of candidate images for this test, which was based on the best ranking result obtained during our tests. We hence refer to these scores as "Kentucky ROC-integral" or "Places ROC-integral" depending on the dataset used.

Along with measuring these quality metrics for different features, we also keep track of the computation time required for detecting the various interest points on a single core of an Intel Xeon 5150 CPU. This is significant for deciding which option to use, especially on computationally constrained platforms such as mobile phones. While we were using optimized versions of the different algorithms, note that the computation time depends strongly on the actual platform and implementation. Therefore the timings shown below may not be representative for other implementations of the same features. Rather than that, the described method and visualizations should serve as a reference for others, when performing similar comparisons of available options.

6.1.3 Feature Comparison Results

The image features used in the following comparison include several detectors from a Microsoft internal library developed by Microsoft Research (MSR), such as the Laplacian, Harris, Hessian,

Hessian Laplace, MSER, and Fast detectors, in addition to our own[6] implementations of the Hessian, MSER and Fast detectors [43, 414, 415]. In the following comparisons, these features will be referred to using the M_ or B_ prefix to indicate which version was used (M=MSR, B=BING).

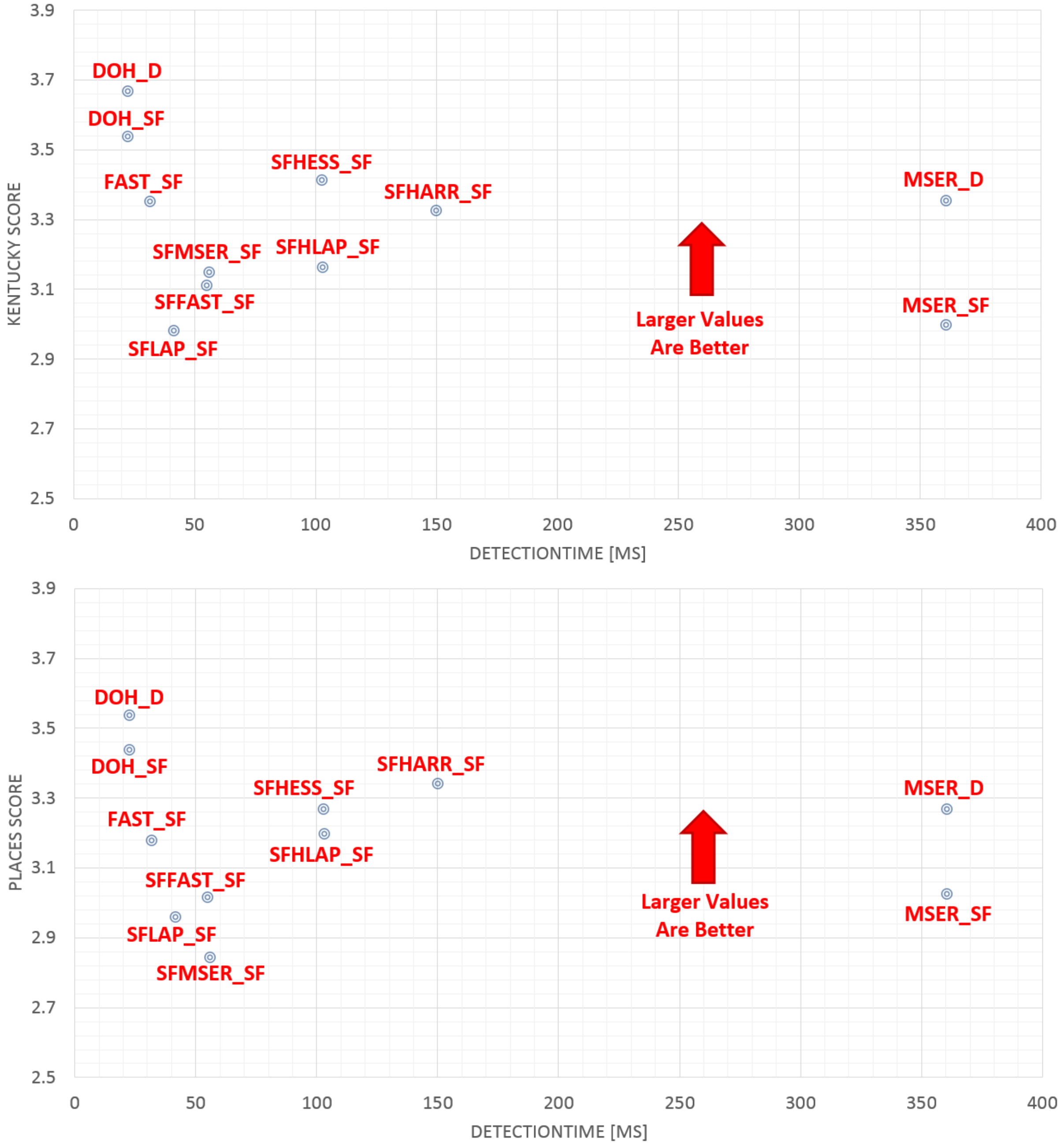


Figure 6.3 Comparison of Ranking Scores of Different Interest Point Detectors for Kentucky Dataset (Top) and Places Dataset (Bottom)

Two different versions of feature descriptors were used in the comparison. All interest point detectors in the MSR library were evaluated with the corresponding descriptor code from the same library, while our versions of the detectors were evaluated with both MSR's and our own

[6] "Our" implementation here refers to Bing Mobile's version of these features, which has been implemented by a team of people led by David Nistér.

descriptors to allow the distinction whether differences were caused by descriptor or detector. Based on the descriptor version used, the following results include either the postfix _M or _B.

The graphs in Figure 6.3 provide a clear comparison of the performance of different feature combinations for the task of image retrieval, based on two datasets captured in different flavors. Both differences in the quality as well as the feature detection time are visualized. For example, it becomes clear that while B_MSER_B achieves about the same Kentucky score as B_FAST_M, the detection time is an order of magnitude higher, potentially making it unfeasible for use on a mobile device. On the other hand, B_HESSIAN_M, using the same descriptor as M_FAST_M is both faster and better for ranking according to this metric. It also turns out that our descriptors when compared to the MSR descriptors by using the exact same interest points, tend to perform better for image ranking.

Given that B_HESSIAN_B has been heavily tuned for speed and image ranking performance, this tendency is not surprising, and a more fair comparison would be to use tuned versions of all different detectors. While there are differences between the scores for the Kentucky and Places datasets between individual pairs of features, the B_HESSIAN_B still ranks optimally compared to the others, indicating that the features have not been over-fitted to any specific dataset. The same tendency occurs for the post-verifier ROC curve integrals for the Places dataset (Figure 6.4), although it is interesting that while the Harris detector apparently worked better for ranking than most alternatives, this is not the case for pairwise matching.

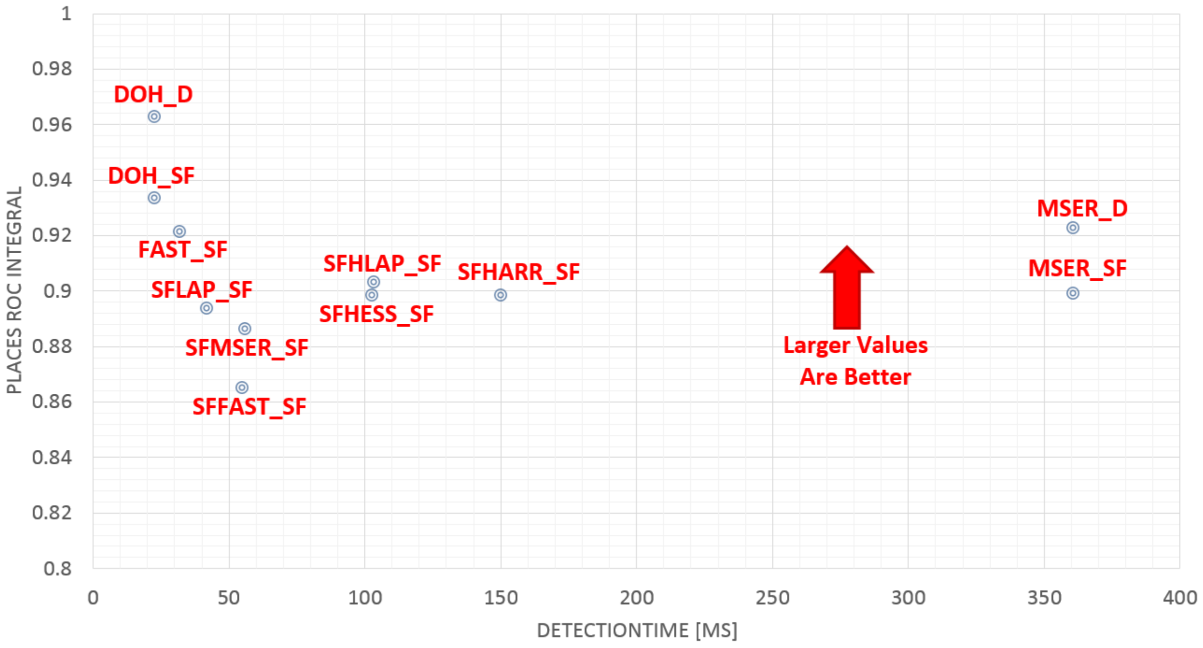


Figure 6.4 Comparison of ROC Integral of Different Interest Point Detectors for Places Dataset

6.1.4 Feature Parameter Tuning

In addition to comparing different feature detectors, the same datasets and metrics can also be used for evaluating the effect of individual parameters of a particular feature extraction algorithm.

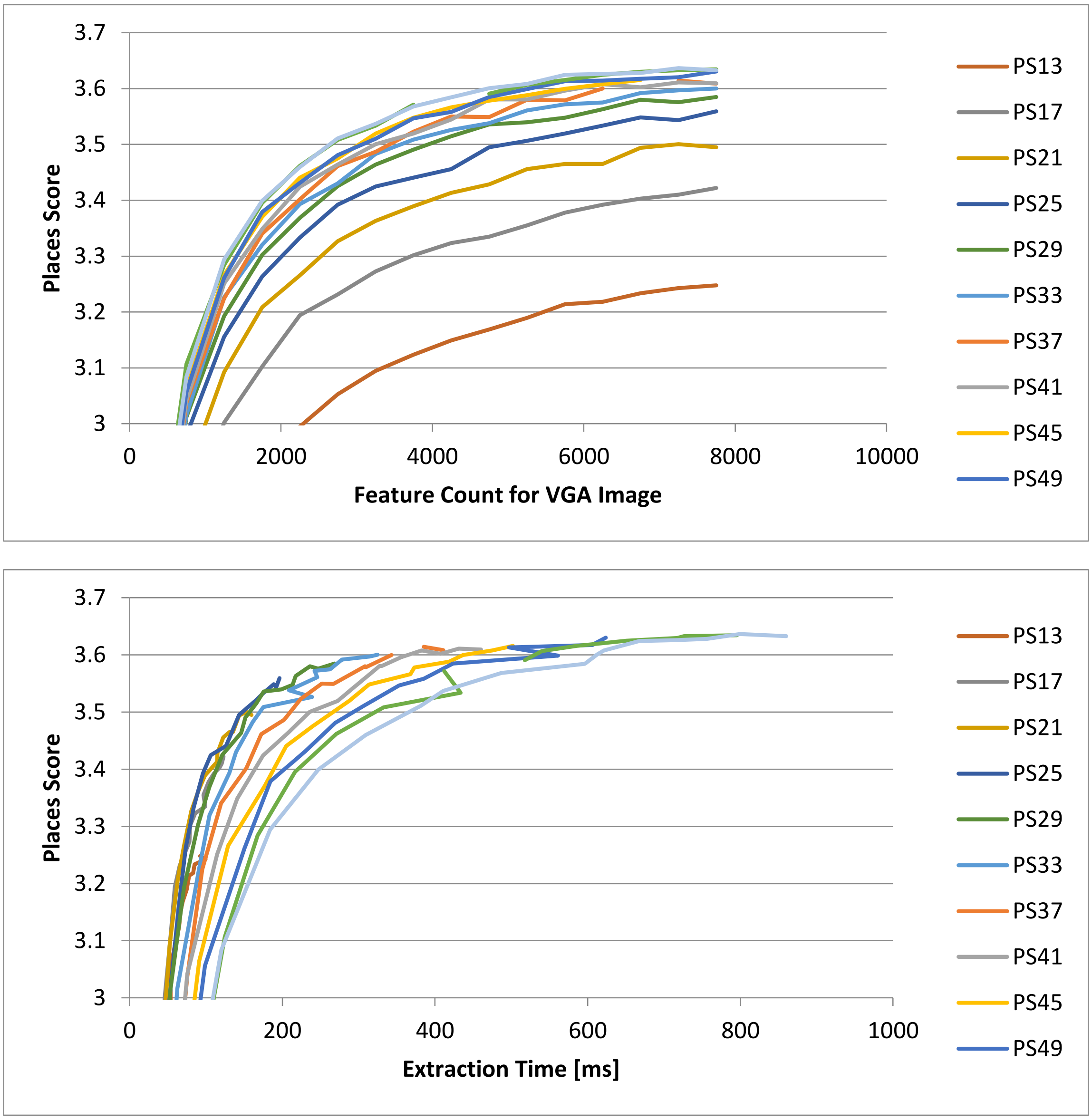


Figure 6.5 Places Score for different Choices for the Number of Features as well as the Patch Size (PSxx) used as a Function of: the Feature Count (Top); the Extraction Time (Bottom);

Two examples of parameters which can often be freely chosen, are the number of features to be detected per image (or the feature density), and the patch size that used for computing the feature descriptor for each interest point. Based on the patch size, image patches are chosen from different levels of the image pyramid, thus containing fewer or more pixels that have to be taken into account for computing orientation histogram based descriptors. While the descriptor compute time is linearly related with the number of features, it is typically proportional to the square of the patch size.

Questions such as if a larger number of smaller patches or a smaller number of larger patches is preferable can best be answered by comparing relevant metrics. The chart in Figure 6.5 (a) shows the relation between patch size, feature count and the Places score. From this chart it is still not clear whether to use 5,000 features with 25 pixel patches, or 2,200 features with 57 pixel patches, as both perform equally well for ranking. The chart in Figure 6.5 (b) which shows the Places score as a function of the extraction time (detection + feature computation) makes it clear, that fewer features are a better choice, as the extraction is about 2.5 times faster. The same reasoning can be made between any pair of parameter sets that can be evaluated, and for which data points can be drawn in the above diagrams.

6.1.5 Optimizing Quality vs. Speed

In order to fit a certain algorithm, such as a feature extraction method, into a given form factor CPU or other compute device, significant effort is often required to optimize code, improve algorithmic performance, or even replace one algorithm with another. As improvements are made towards higher speed, the performance of the algorithm tends to decline. For this purpose, using a set of clearly defined metrics describing the quality performance of said algorithm is essential for keeping track of the progress made. In our experience, the quality-speed graphs presented above prove to be a valuable way of tracking this progress.

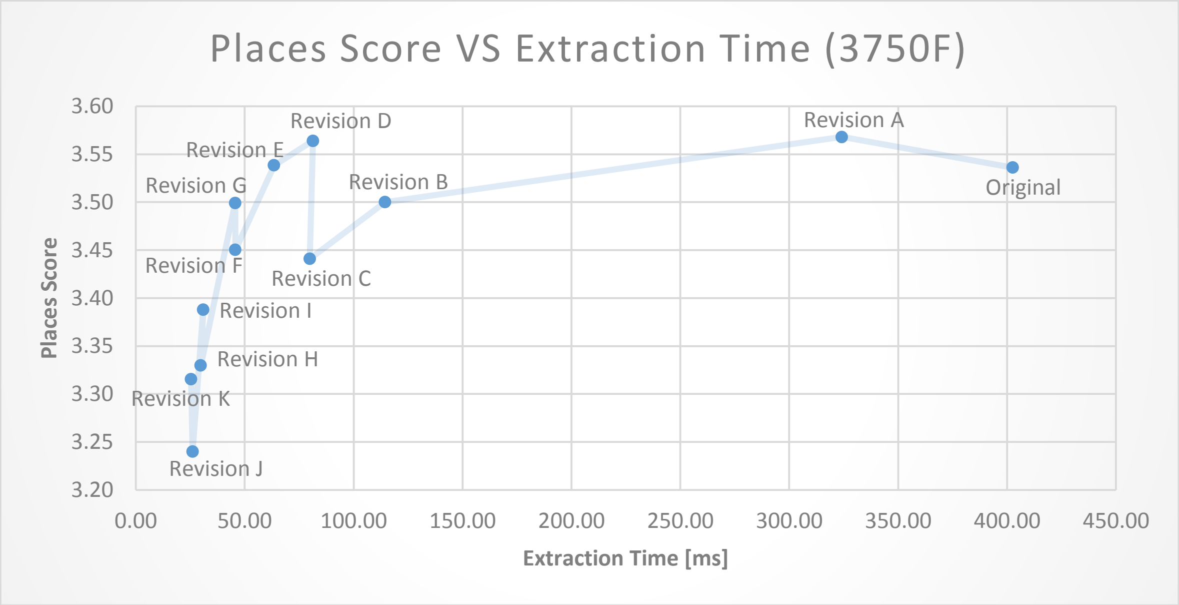


Figure 6.6 Quality-Speed Progress while Optimizing the Performance of Hessian Feature Detector

The quality-speed graph in Figure 6.6 visualizes the progress of Places scores and feature extraction times during several revisions of algorithmic improvements and parameter tuning for the hessian interest point detector and polar descriptor described in Section 5.5.1. For the test 3,750 interest points and descriptors were extracted from a VGA image.

The individual revisions shown in the diagram, as well as the significant parameters and changes made are detailed in Table 6-1. The biggest speed gain to the detector (Revision A) could be achieved by algorithmic improvements such as by switching to a recursive Gaussian blur [238] implementation originally described in [404] and by code optimizations in the determinant of hessian computation as well as the extrema-search. Further improvements were achieved by reducing the descriptor patch size and the resolution of the highest pyramid level used for the detection. For each major change in these primary parameters, several other parameters of the interest point detection and descriptor algorithm were tuned to optimize the Places score. Overall, a **16-fold** speedup from **400 ms** to **25 ms** could while still retaining a acceptable retrieval rate.

While the optimizations performed during revisions H-K provide further speed enhancements, we think that revision G provides the best tradeoff between the quality for image ranking and the extraction time. To compare the timing on the PC with typical timings on a mobile CPU, we further evaluated Revision G on an ARM Cortex-A8 mobile CPU present in an IPhone 4 device. On this CPU the feature detection on a VGA image took approximately 143 ms and the extraction of 3750 descriptors took 455 ms, which means that the total extraction time (598 ms) is 13 times longer than on the PC.

Version	Extraction Time [ms]	Places Score	Highest Pyramid Level	Descriptor Patch Size	Change
Original	402.65	3.54	640·480	57	-
Revision A	324.23	3.57	640·480	57	Detector Code Optimizations
Revision B	114.46	3.50	640·480	33	Patch Size 33 Pixel
Revision C	79.92	3.44	640·480	25	Patch Size 25 Pixel
Revision D	81.38	3.56	640·480	25	Parameter Tuning
Revision E	63.52	3.54	320·240	25	Reduced Pyramid Resolution
Revision F	45.72	3.45	320·240	19	Patch Size 19 Pixel
Revision G	45.62	3.50	320·240	19	Parameter Tuning
Revision H	29.88	3.33	320·240	13	Patch Size 13 Pixel
Revision I	30.95	3.39	320·240	13	Parameter Tuning
Revision J	26.10	3.24	192·144	13	Reduced Pyramid Resolution
Revision K	25.40	3.32	192·144	13	Parameter Tuning

Table 6-1 Revisions of Hessian Feature Extractor during Optimization

6.1.6 Summary

We have presented an evaluation method and metrics allowing the comparison of various image features for image retrieval as well as pairwise image-matching as part of a post-verification. The evaluation is based on two different datasets for general objects as well as location-specific imagery, and considers both the quality as well as the feature extraction time in a quality. We have further compared several interest point detectors and feature descriptors using this method. While the DOH descriptor and polar descriptor presented in Section 5.5 compare favorably to the compared options, this evaluation should be repeated for other problems than the one considered. Finally we have presented the utility of the evaluation method for tuning parameters of a feature extraction method, as well as for tracking the progress of code-optimizations of a specific extractor. In case of the feature extraction method presented in 5.5, a **16-fold** speedup from **400 ms** to **25 ms** could be achieved by means of code optimizations and parameter tuning.

6.2 Bing Maps Streetside Photos

The results of our prior work described in Section 5.3 are a set of geo-located user photographs superimposed on the matching streetside imagery of the same location, which were later shipped as a feature of Bing Maps called Streetside Photos [46]. Since the same application is also supported by the improved image search index described in Section 5.3.2 we repeat the same test metrics and datasets in order to compare the quality performance of the old and new algorithm.

6.2.1 Offline Dataset

For this purpose we used a set of roughly ***300,000*** precisely (within +/- 2 m) geocoded panorama images in Seattle (Figure 6.7) covering an area of 10·13 km as the base model for location matching. As query images we used a test set of ***11,000*** images downloaded from Flickr that were geocoded within a radius of ***100 m*** to at least one of the Streetside-panorama images. The image dimensions of the Flickr images used are ***500·375*** pixel. Out of those query images, ***2,615*** had been hand-labeled as outdoor-images containing recognizable image content (23%). We consider only those images as potential match candidates.

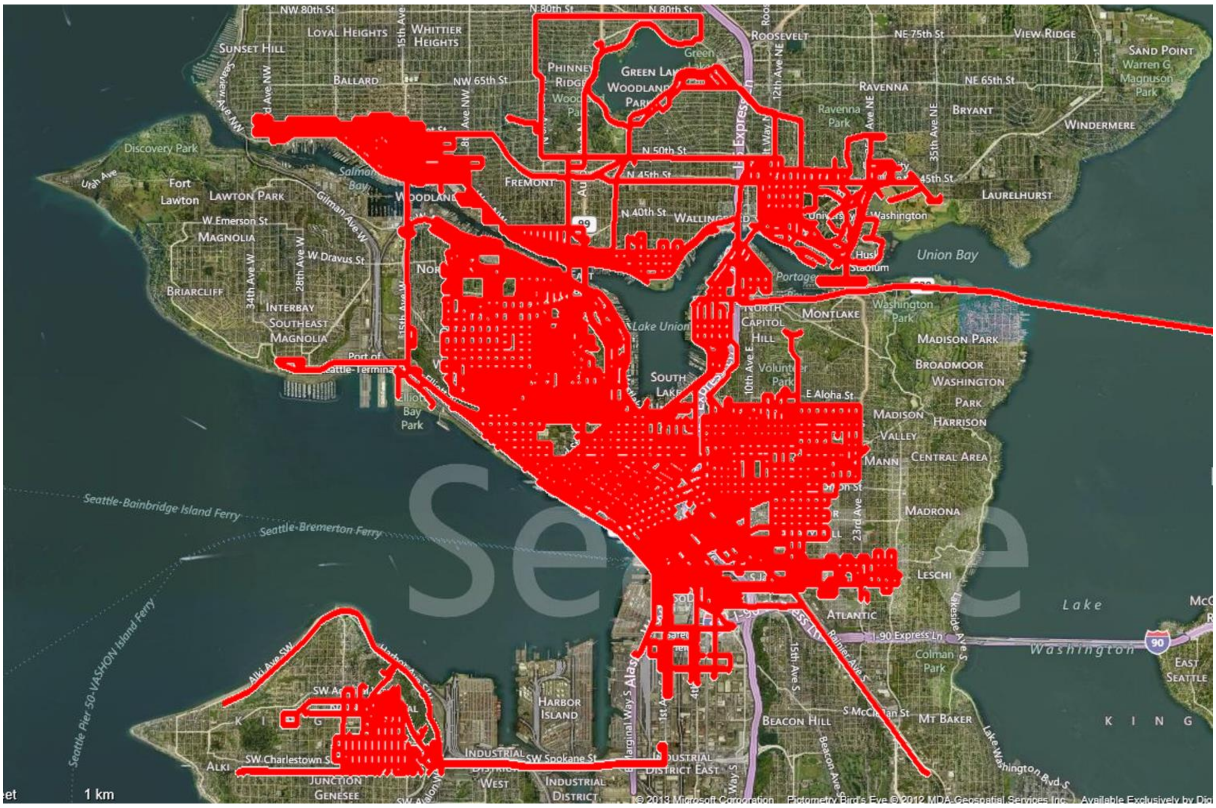


Figure 6.7 Overview of Test Set of 300,000 Human Scale Panoramas in Seattle, WA, USA

For the evaluation of the new method we used a search radius of ***100 m*** around the prior location and a maximum number of ***200*** post-verifications per query image. We evaluated two variants of the RANSCAC method, either using the similarity transform determined after the first optimization, or the homography after the second optimization. We used the same Laplacian interest point detector and Daisy descriptor [393] as for the original method, hence the changes in the results are mainly caused by differences in the image ranking and pairwise post-verification process. The compute time spent per query on a 2006 CPU (Intel Xeon 5150) is approximately **1 second** for feature extraction and ranking, and **0.081 seconds** for each post-verification (0.03 seconds for I/O, 0.05 seconds for KD-tree search and 0.001 seconds for RANSAC). Altogether this adds up to **17.2 CPU-seconds** assuming 200 post-verifications. As the post-verifications are distributed across 10 machines in the production system, the total server side latency (without networking) turns out to be roughly **3 seconds** (≈**1 + 20 * 0.081**).

6.2.2 Offline Evaluation Results

The comparison of the results for the new as well as the original methods are listed in Table 6-2. As one can see, the true positive rate (TPR) has improved substantially from 59.5% to **71.1%** for the similarity based model and to **73.3%** for the homography based model, with a similarly low false positive rate (FPR) of **0.6%**. The alignment accuracy for features on planar surfaces usually is better than **5 pixel** (1%) reprojection error measured in query image coordinates. For 3D structures, reprojection errors are significantly higher, as expected due to shortcomings of the homography model.

	Original Results from [25]	New Method Using Similarity Based Geometric Model	New Method Using Homography Based Geometric Model
True Positives	1556	1859	1918
True Positive Rate	59.5%	71.1%	73.3%
False Positives	59	88	67
False Positive Rate	0.5%	0.7%	0.6%

Table 6-2 Comparison of True and False Positive Matches for Original and New Method

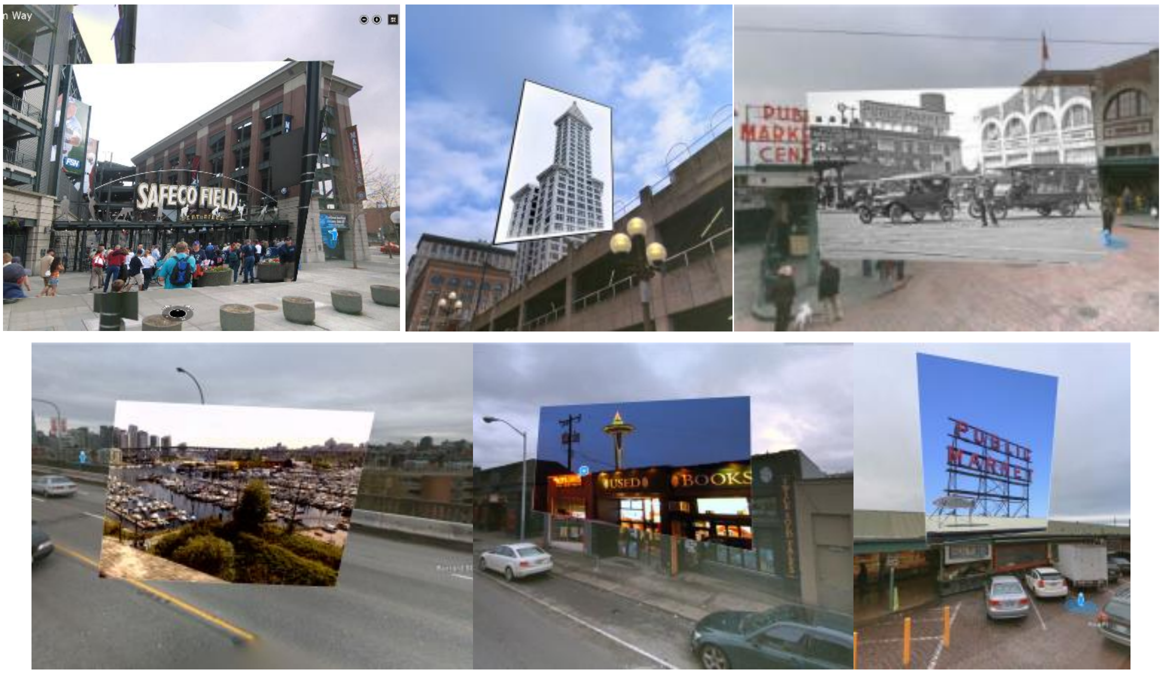


Figure 6.8. Examples of Images Shown in the Context of the Matched Streetside Panorama Images, in the Bing Maps Silverlight Client. Top Right is a Historic Image from 1919 at Pike Place, Seattle

Some samples results from the original method are shown in Figure 6.8, in the context of the matched panorama images. More match results can be seen in the Bing Maps Application "Streetside Photos" [46]. In addition to the matches listed above, 376 query images (Figure 6.9)s could be matched successfully using the homography based method, despite substantial differences compared to the panorama images, in resolution, sharpness, illumination, perspective, camera geometry or due to noise or occlusions.

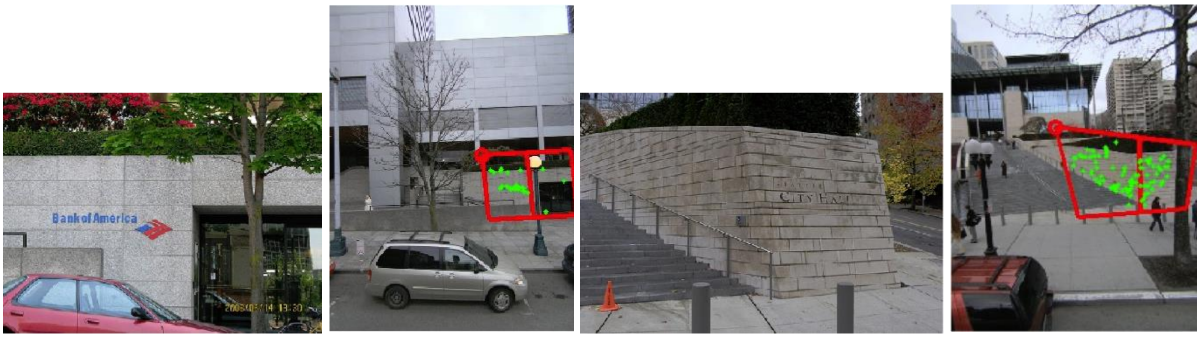


Figure 6.9 Examples of Successful Matches with Minor Overlap between Query and Index Images; Detected Bounding Polygon is Shown in Red, Matched Feature Points in Green;

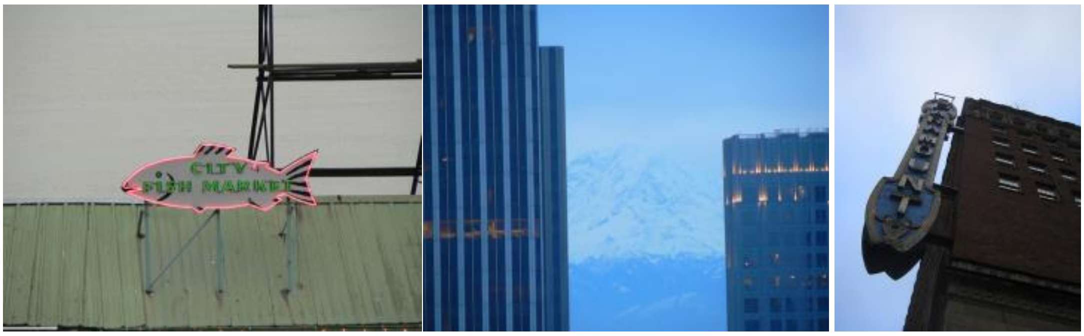


Figure 6.10. Samples of false negatives (Images not successfully matched)

In case of the homography post-verification, **697** false negatives were detected. Based on a sampling we estimate that about 200 of them had been geocoded incorrectly, such that the correct human scale image was outside of the search scope of 100 m. Most of the other images (Figure 6.10) either had a very narrow field of view (left example) containing too few uniquely identifiable features, large amounts of repetitive structures such as building façades with many windows (center example), or they were taken from a perspective too different from the panorama image (right example). In addition, due to the use of the homographic geometry model for match-verification, scenes with pronounced 3D structure were also more challenging to match.

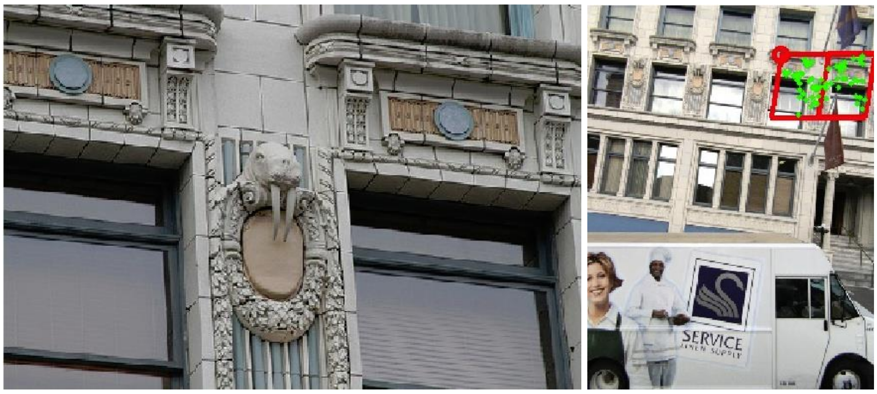


Figure 6.11 Sample Query Image and Streetside Match Demonstrating the Problem of Repetitive Structure in Urban Environments;

The problem of repetitive scenes in urban environments is also visualized in Figure 6.11. While the image of the ornament (left) could actually be matched automatically by the algorithm, there are several visually indistinguishable ornaments on the same building (right). The matching algorithm is unable to uniquely identify the particular ornament which has been photographed and instead returns an arbitrary one of them.

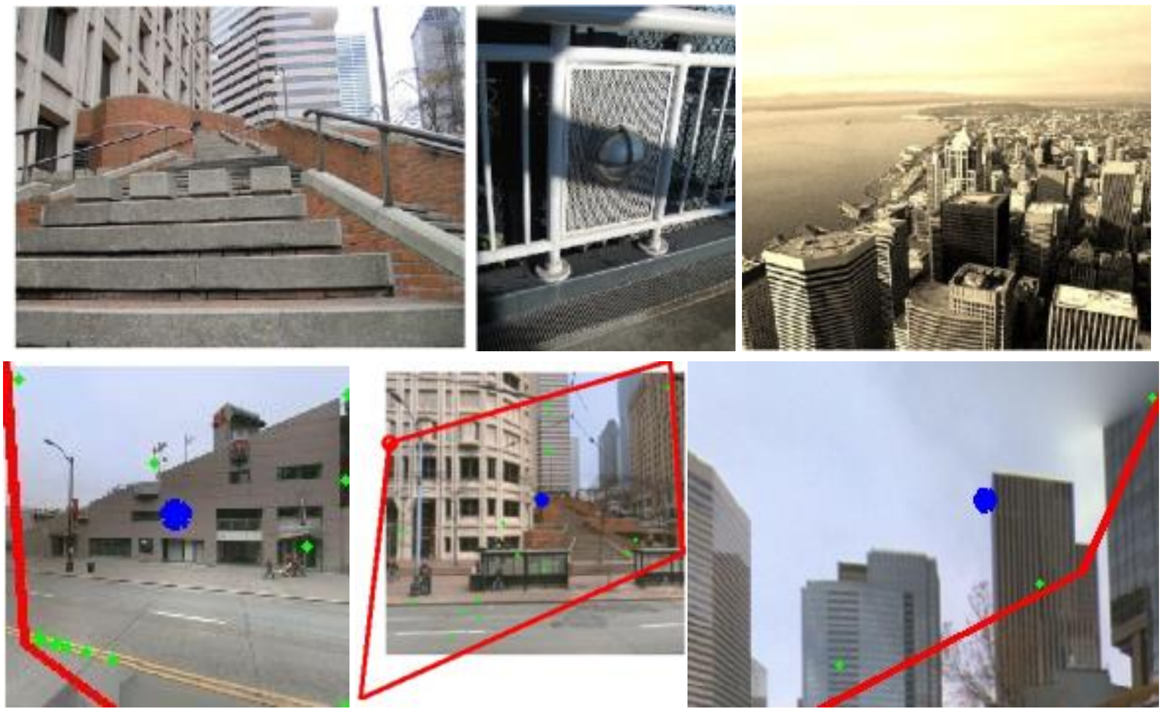
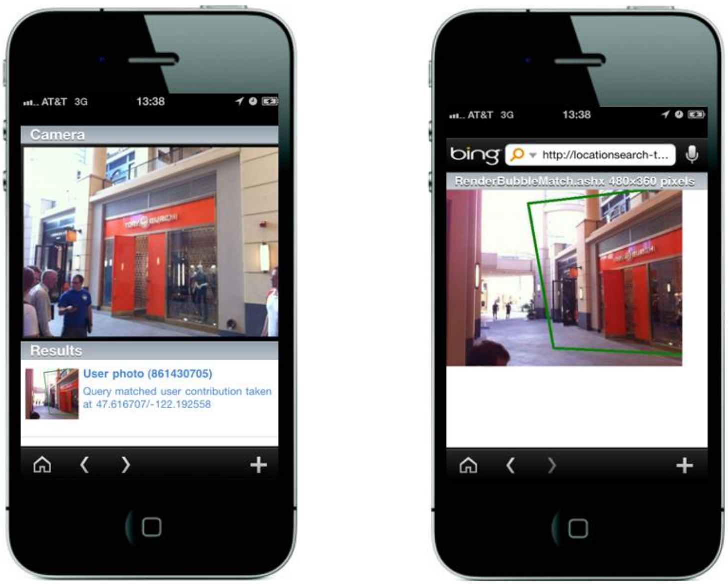


Figure 6.12. Samples of False Positive Matches. Query Image (Top) and Corresponding Mismatched Index Image (Bottom). Red Lines Show Projected Outline of Query Image if Within View, Blue Dots Show Projected Image Center.

Altogether ***67*** false matches were counted using the homography RANSAC, corresponding to a false positive ratio of ***0.6%***. Images containing repetitive structures, such as window shutters, building fronts with repetitive window-patterns or similar textures were more likely to be mismatched (See Figure 6.12 for samples of false positive match pairs), even though their matching scores were usually relatively low.

6.2.3 Real-Time Matching

In addition to supporting offline matching of query images from various sources to an index of Bing Maps streetside panoramas, the image index in Chapter 4.6 was optimized particularly for real-time queries and ingestions from mobile phones. Various applications are supported by real-time queries, such as image-based localization based on accurately (+/- 2 m) positioned streetside data, or queries for information related to specific landmarks. For example, a user may capture an image of a landmark, add it to the index, and create a label linking to an online article of related information. A second user would hence be able to obtain the respective information by issuing an image query to the index. While the example above in Figure 6.1 highlighted the ability to perform real-time matches to streetside imagery, Figure 6.13 shows the same capability for user-uploaded index images. The example query (left) was issued inside of a shopping mall and returned the match to a previously transmitted and ingested query image. Note that since the dynamic image index described in 5.6 supports real-time ingestion of new images, the matched image may have been added as little as 5-10 seconds earlier either by the same or a different user.


AT&T 3G
13:38
Camera
Results
AT&T 3G
13:38
bing
http://locationsearch-t...
RenderBubbleMatch.ashx 480×360 pixels

User photo (861430705)
Query matched user contribution taken at 47.616707/-122.192558

Figure 6.13 Query Image Captured with Smartphone in Indoor Location (Left); Matched User-Photo with Detected Bounding Polygon (Right);

While we did not perform an extensive analysis of the real-time query performance, we experienced satisfactory matching rates clearly above **80%** in about **250** user issued queries containing recognizable image content also present in the index. The matching rate was higher than for the offline case with CPC images, likely due to the fact that query images were intentionally captured with a large enough portion of recognizable image content, such as buildings on the opposite side of the street.

The end-to-end query times varied between **5 s** and **20 s** between capturing an image and obtaining the resulting list. We verified visually that the bounding polygon indicated a plausible area in the index image for all successfully answered queries. The only ambiguous results were due to repetitive structures similar to the example in Figure 6.11, where a similarly looking scene feature was returned rather than the exact object in the query image.

6.2.4 Summary

In this section we have shown that the presented geospatial image index can be used effectively for matching user photographs from CPC and smartphone cameras, to streetside imagery and other geocoded data. The achieved recall rate of **73.3%** as well as well as the false positive rate of **0.6%** compare favorably to comparable solutions such as [363] or [365]. Server query time of less than **3 seconds** allow real-time application of the system for self-localization in a map of streetside imagery, as well as offline processing of **10,000** queries per hour.

6.3 Improving Business Geocoding Using Storefront Images

6.3.1 Problem Description

A key functionality of any internet mapping service is the search for points of interest (POI) such as businesses in an area, and provide information about the location of the POI as well as navigational instructions. As users see the world at increasing scales in human scale imagery, the need for accurate POI location information grows. While an error in location of about 50 m may be acceptable when using an aerial view, it is unacceptable in human scale. Due to the fact that the geocoding for a large percentage of POI is derived from their street addresses, by interpolating across coordinates of street intersections, the geocoding errors can be significant, and often tens or hundreds of meters away from their actual location [140]. If the geocoding for a business is offset by more than 20 m, it is often unlikely to be spotted from the streetside image closest to that location. This assumption is supported by the example of a business storefront shown in Figure 6.14 from streetside images offset along the street by 0, 5, 7.5, 10, 20 and 30 m. While at **10 m** distance the business is still visible and may be recognizable, this is no longer true for 20 m. A manual analysis of a set of POIs on Bing Maps, that we conducted in several US cities with human scale data conducted in 2010, showed that about **2/3** of all POI entries were offset by more than **30 m** from their actual location.

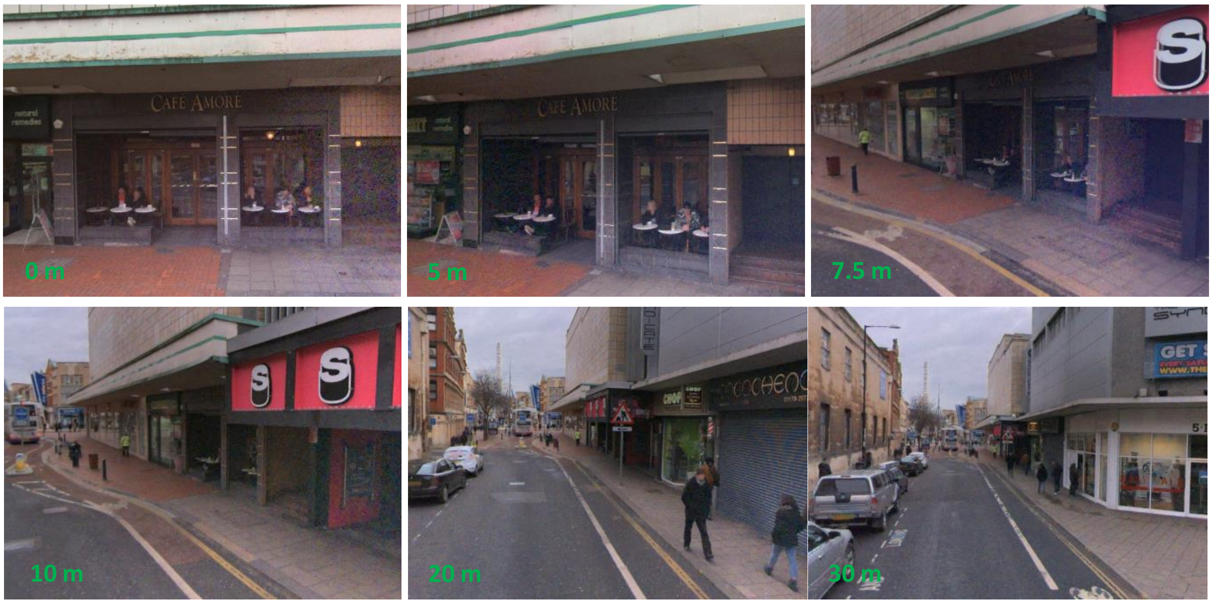


Figure 6.14 Business Storefront of "Café Amore" in Bristol, GB - Viewed from Streetside Images at Different Distances along the Street

For that reason it is a priority for a mapping service to improve the geocoding of existing POIs as much as possible, in order to provide precise information to its users. An expensive way to improve the geocoding of businesses is to send a GPS equipped team of people into the field to obtain updated GPS locations for a list of POIs. A cheaper alternative is the use of human scale imagery, which is captured more frequently in urban areas with businesses, where accurate geocoding matters the most. For example, the human scale imagery can be used by paid data-labelers to search for the correct location of a business, and manually update the geocoding information accordingly. While this approach certainly can achieve the desired result of more accurate geocoding, it is also expensive, and cannot easily be scaled up to millions of POIs in thousands of cities. Alternatively, users of map services could be involved by means of crowdsourcing, which may have other draw-backs such a lack of motivation to participate or the risk of spam being introduced. Therefore, a reliable automatic solution would be preferable.

6.3.2 Proposed Solution

In [47] we have propose an approach to improving business geocoding by using image matching techniques. This approach takes advantage of the fact, that for many POIs, storefront images (Figure 6.15) had been used even prior to capturing streetside imagery (see 3.4.2). In addition to systematically collected storefront images, community sources such as Yelp could be used, as many business database entries contain user generated business storefront images. The system described in Chapter 4.6 may serve as a means to achieve reliable image matching. By providing a reliable match between the storefront images and the streetside panorama images, precise 5DOF locations and orientations can be determined for the respective POIs.

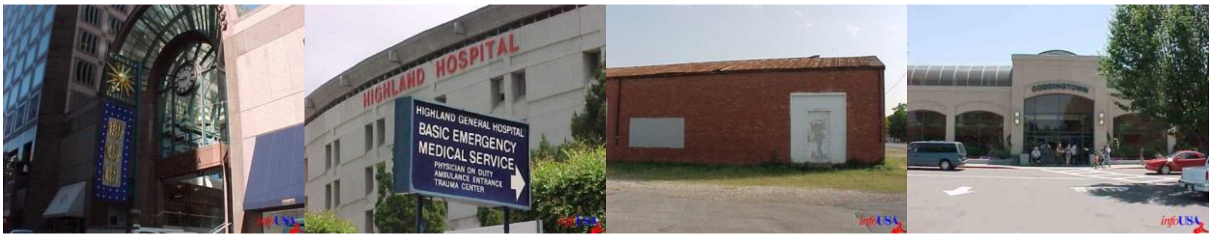


Figure 6.15 Sample Business Storefront Images Captured by InfoUSA [247]

A goal for any automatic processing system, in addition to achieving the best possible result, is to minimize the compute time required for the task, as it often directly affects the cost. As mentioned in Section 0 the average compute time per query image is approximately **17.2 CPU-seconds** on a 2006 CPU (Intel Xeon 5150), assuming that 200 images need to be post-verified in total. Assuming current pricing of a cloud computing service such as Windows Azure of 18 cents/hour for a medium sized node [416] and an inefficiency factor of 3 it would cost **15,000 USD** to update 1 million business entries using the above system. As an alternative a crowdsourcing solution such as Amazon Mechanical Turk [417], which may cost 5 cent per POI transaction [418] would cost **150,000 USD** with a redundancy factor of 3 for the same number of POIs. Hence an automated solution appears to be favorable in this case.

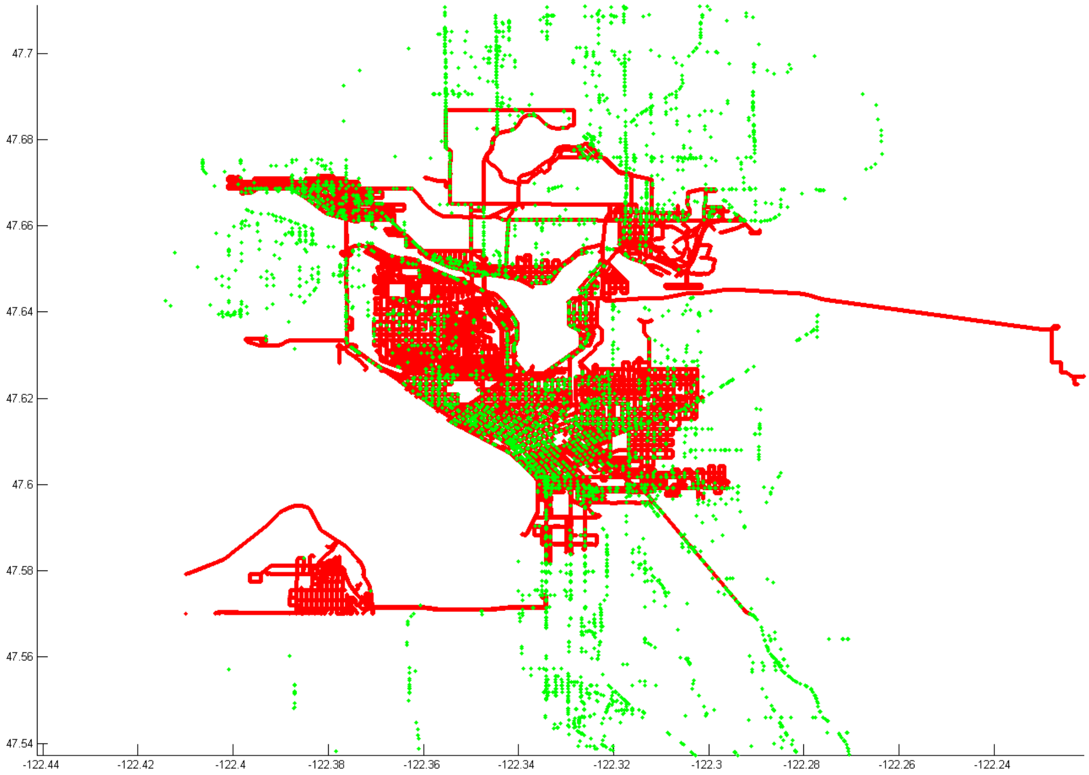


Figure 6.16 Overview of Streetside Panoramas (Red) and Storefront Images (Green) used for Geocoding Experiment

6.3.3 Experiments and Results

In order to evaluate the feasibility of using image matching for updating the geocoding of POI entries on Bing Maps, we used a set of roughly ***300,000*** precisely geocoded panorama images in Seattle, in combination with ***17,600*** storefront images in the same area, collected by InfoUSA [247]. The storefront images provided (see examples shown in Section 3.4.2) have a resolution of 400·300 pixel, and the image quality is often suboptimal due to challenging exposure conditions, blurriness, perspective distortions etc. The geographic distribution of the two image types used is visualized in Figure 6.16.

Out of the total amount of storefront images, ***9,116*** were actually close enough to the available streetside panoramas (within a 100 m radius) to be considered for image matching. Out of this set, ***6,906*** images could be matched to the streetside dataset in order to improve their geocoding. This corresponds to a recall rate of **75.8%**, which is an encouraging number considering the large variation in the appearance of streetside scenes generally, and the suboptimal image quality of the storefront images used. Additionally, an unknown but non-zero percentage of the storefront images located within a 100 m radius likely is not actually visible from any of the streetside panoramas, as the businesses are located in cross streets or within malls.

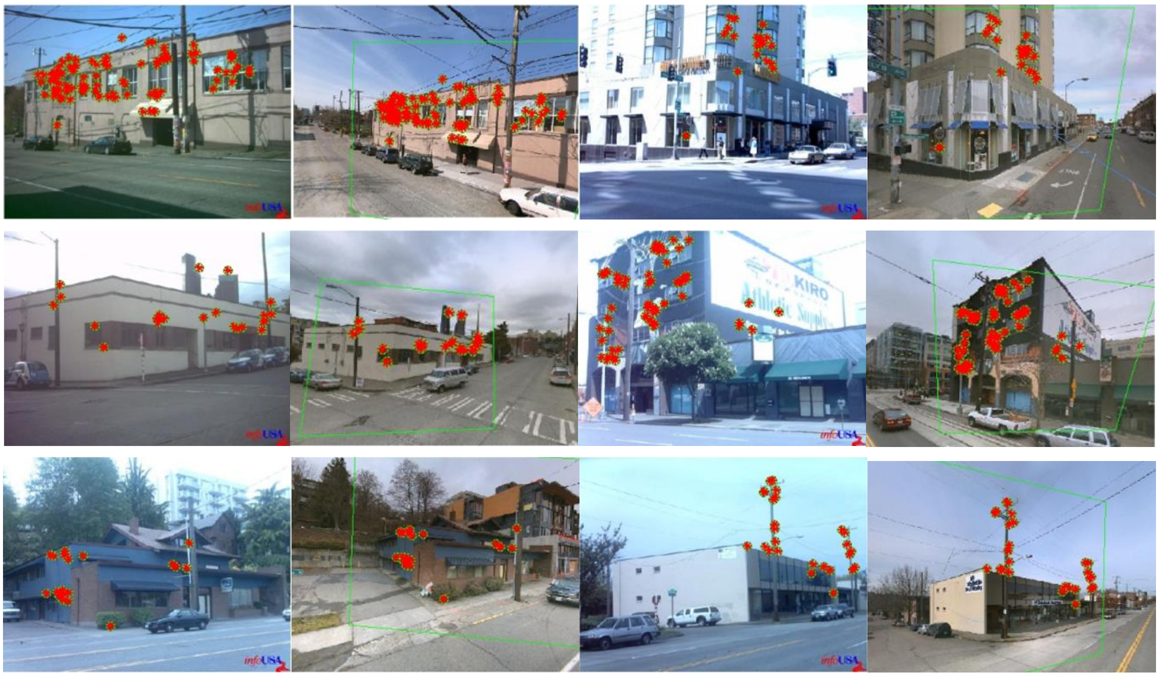


Figure 6.17 6 Successfully Matched Image Pairs of Storefront Images and Streetside Panoramas

Examples of matched image pairs of storefront images and streetside panoramas can be seen in Figure 6.17, illustrating the differences in the ratiometry and geometry between the two image sets.. The location of the matched features as well as the estimated overlap region have been highlighted in this view.

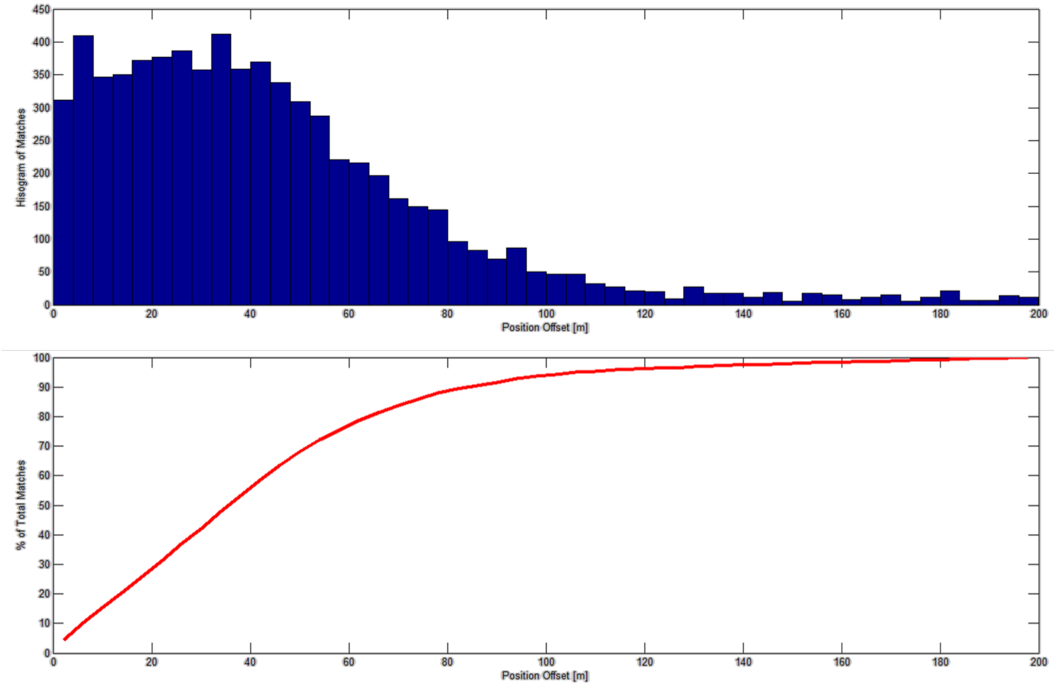


Figure 6.18 Histogram and Cumulative Histogram of Position Offsets between Prior Geocoding Location and Location Determined by Image Matching

The histogram and cumulative histogram (Figure 6.18) of the actual position offset determined by comparing the prior geocoding information for a POI with the location of the streetside panorama show that approximately 60% of businesses previously had geocoding errors in excess of 30 m.

The average prior error was **43.7 m**, far above the tolerable distance of 10 m illustrated in 6.3.1. A visual inspection confirmed that the updated business locations were all within a tolerable radius of 10 m such that the businesses could actually be seen from the closest streetside panorama.

As an alternative approach of automatic business geocoding from streetside imagery, we applied optical character recognition (OCR) on the imagery, and tried matching the retrieved text strings with the names of nearby POIs in the database. This approach only applies to a subset of business which have their name displayed on the building storefront or on signs. For this experiment we used an OCR algorithm tuned for text in natural scenes on a smaller set of ***30,000*** streetside panoramas and ***1,305*** POIs. Out of them, ***99*** could be found correctly, corresponding to a recall of approximately ***7.6%***. The details of this experiment, which was based on a variant of the OCR algorithm used for Bing Text Search and the Levenshtein distance [419] as a text similarity metric, are not subject of this work. For the same smaller area, a prior version of the image matching approach similar to what was presented in [25], was able to match ***735*** images correctly, corresponding to a recall of ***56.3%***. While we didn't repeat the OCR based experiment on the larger dataset described above, the difference in recall suggests that image matching has a significantly higher potential to achieve the desired result.

6.3.4 Summary

The results presented above show that matching business storefront images to streetside panoramas by means of the proposed image index, can be used to update business geocoding successfully in **75.8%** of cases. This improves the usability of the corrected POI significantly when viewed in human scale mode, as **2/3** of the POI were previously located outside of a presumably tolerable radius of **20 m**. The cost and performance compares favorably to alternative approaches involving crowdsourcing or optical character recognition.

6.4 Constraining Data Upload Volumes for Mobile Search

6.4.1 Problem Statement

Recent advances of mobile internet technology such as LTE have led to significantly increased bandwidths available for uploading and downloading data in certain geographic locations. Nevertheless in most locations the typical data volumes used for image based search still pose a challenge if the goal is to minimize the latency for a user initiated image query. Typical upload bandwidth for UMTS mobile networks, which is one of the most common standards [420], are on the order of **25 kB/s**. This means that uploading an uncompressed RGB image in VGA resolution (640·480 pixel) to a web service takes on the order of 35 seconds which is unacceptable for most applications. Therefore it is desirable to reduce the upload file size in order to minimize the latency added by the file upload, ideally to less than **1 second**.

Various options of reducing the file size for an image exist such as by lossless or lossy image compression or down-sampling the image for transmission. Since the required compression ratio makes lossless compression infeasible, we therefore evaluate only lossy compression using the JPEG algorithm [421] with varying degrees of compression. For resizing the images to different dimensions we use the bicubic interpolation method [238]. Another option would be to perform the feature extraction on the mobile device and upload only the extracted interest points and descriptor vectors, which requires additional compute latency on the mobile device.

An important question in this context is which of the above options leads to the best tradeoff between file size and matching performance. In order to answer this question, we used the same streetside image dataset and matching algorithm used in Section 6.1.6 containing 11000 geocoded user images and 300,000 streetside panoramas. Hence we used a VGA version of the user images from Flickr as a starting point for resizing and compression to different settings. The metrics for comparing different options are the TPR and FPR of the matching results. In the following sections 6.4.2 through 6.4.4 we describe the results of three different parameter variations.

- Vary the image dimensions for transmitting the image, using a constant JPEG quality setting of **100%**.
- Vary the JPEG quality setting between **10%** and **100%** while keeping the image dimensions constant at **500·375** pixel
- Vary the image dimensions as well as the JPEG quality setting concurrently in order to achieve a constant average file size of roughly **20 kB**. At receiver resize image back to 500·375 before feature extraction.

6.4.2 Constant JPEG Compression Quality

For the first test series we left the JPEG quality setting unchanged at 100% leading to a minimal compression and quality loss, while varying the image dimensions in a range between 320·240 and 590·443 pixel. Table 6-3 gives an overview of the parameters used as well as various other relevant metrics, including the average file size per image, the average feature count as well as statistics about true and false positives.

Image Dimensions	JPEG Quality	Average File Size [kB]	Average Feature Count	TP	FP	TPR	FPR
320·240	100	67.3	603.2	1668	11	0.63786	0.00088
350·263	100	81.0	711.8	1746	16	0.66769	0.00128
380·285	100	94.8	823.8	1822	24	0.69675	0.00192
420·315	100	115.3	985.6	1865	26	0.71319	0.00208
450·338	100	131.6	1116.0	1913	34	0.73155	0.00272
480·360	100	147.4	1249.7	1926	52	0.73652	0.00416
500·375	100	159.9	1339.0	1935	61	0.73996	0.00488
540·405	100	180.0	1498.5	1977	74	0.75602	0.00592
590·443	100	206.2	1709.9	1980	95	0.75717	0.00760

Table 6-3 Resulting Metrics for Varying Image Dimensions with Constant JPEG Quality Setting

TPR and FPR are further plotted as a function of the average file size in Figure 6.19. From this result it becomes obvious that while both TPR and FPR increase with higher image dimensions the file size is in all cases too large to transmit an image within 1 s on a network with 25 kB/s transfer speed. The increased TPR and FPR are likely explained by the increased feature count for higher image dimensions.

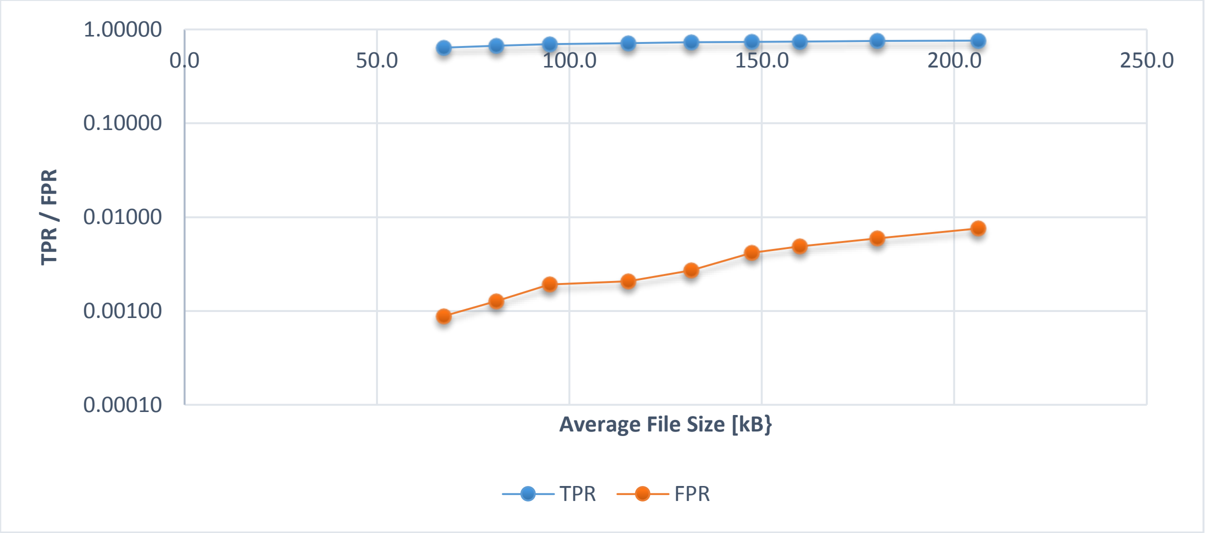


Figure 6.19 TPR and FPR vs. File Size for Varying Image Dimensions

6.4.3 Constant Image Dimensions

In a second experiment we evaluated how varying the JPEG quality for a given dimension of the query images (500·375 pixel) affects the file size and the quality metrics. In addition to the TPR and FPR metrics Figure 6.20 also shows the relation between the file size and the JPEG quality setting.

Image Dimensions	JPEG Quality	Average File Size [kB]	Average Feature Count	TP	FP	TPR	FPR
500·375	100	159.9	1339.0	1935	61	0.73996	0.00488
500·375	90	56.6	1344.4	1940	59	0.74187	0.00472
500·375	80	38.1	1359.0	1935	52	0.73996	0.00416
500·375	70	30.2	1382.0	1912	62	0.73117	0.00496
500·375	60	25.2	1410.2	1933	50	0.73920	0.00400
500·375	50	22.1	1437.3	1943	63	0.74302	0.00504
500·375	40	19.1	1472.9	1929	55	0.73767	0.00440
500·375	30	16.1	1520.4	1911	59	0.73078	0.00472
500·375	20	12.5	1596.0	1858	52	0.71052	0.00416
500·375	10	8.3	1655.1	1720	49	0.65774	0.00392

Table 6-4 Resulting Metrics for Varying JPEG Quality Setting with Constant Image Dimensions

In this case the TPR and FPR metrics remain surprisingly constant throughout a wide range of the sweep despite a large reduction of the file size. The average feature count increases for lower quality settings which is likely caused by the additional amount of compression noise present in the images. A drop of the TPR can only be noticed from 30% to 20% JPEG quality. Generally this test suggests that an image quality setting above **40%** doesn't add significant value to the matching results. The average file size for 40% JPEG quality is only **19.1 kB** which means it would take **0.764 seconds** to upload an image via a 25 kB/s network connection.

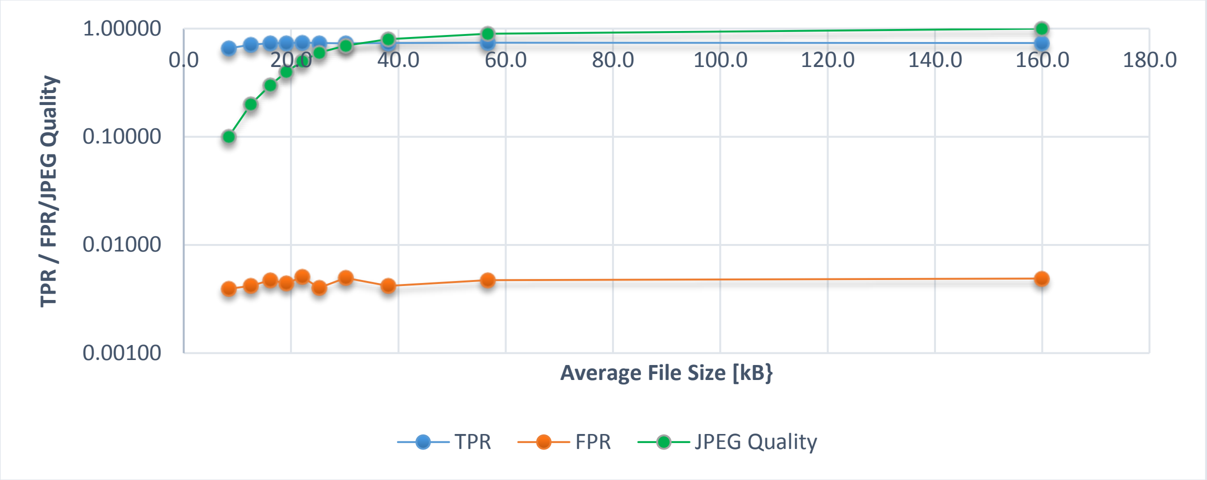


Figure 6.20 TPR and FPR vs. File Size for Varying JPEG Quality

6.4.4 Constant File Size

Based on the results presented in 6.4.3 an interesting question is whether it is preferable for a given file size of roughly 19 kB (corresponding to the 40% quality setting above) to use a larger image resolution with a higher compression, or a lower resolution with less compression. To answer this question we varied both the JPEG quality as well as the image dimensions in conjunction while transmitting the image, keeping the image file size roughly the same. To make sure the image dimensions used during the feature extraction don't influence the feature count, we resize the images back to 500·375 pixel on the receiver side before feature extraction.

Image Dimensions	JPEG Quality	Average File Size [kB]	Average Feature Count	TP	FP	TPR	FPR
320•240	82	17.5	1173.3	1825	46	0.69790	0.0036
350•263	77	18.2	1489.7	1933	72	0.73920	0.0057
380•285	71	18.5	1505.2	1922	69	0.73499	0.0055
420•315	61	18.7	1510.7	1942	88	0.74264	0.0070
450•338	52	18.7	1512.8	1937	70	0.74073	0.0056
480•360	45	18.9	1504.6	1926	64	0.73652	0.0051
500•375	39	19.0	1475.4	1919	63	0.73384	0.0050
540•405	32	18.7	1497.7	1912	80	0.73117	0.0064
590•443	25	18.4	1505.6	1902	48	0.72734	0.0038
640•480	20	18.6	1510.7	1904	47	0.72811	0.0037
800•600	10	18.4	1554.5	1873	57	0.71625	0.0045

Table 6-5 Resulting Metrics for Varying Quality and Dimensions to Achieve Constant File Size

Since the file size is kept roughly constant during this test, we draw the TPR, FPR and JPEG quality as a function of the maximum image dimension in Figure 6.21. While the TPR shows a moderate peak for an image size of 420·315 pixel and a quality setting of 61%, the FPR numbers fluctuate relatively much in this range, making it difficult to select a clear optimum. Part of these fluctuations could be caused by the fact that the images had to be resampled twice using bicubic interpolation for this experiment.

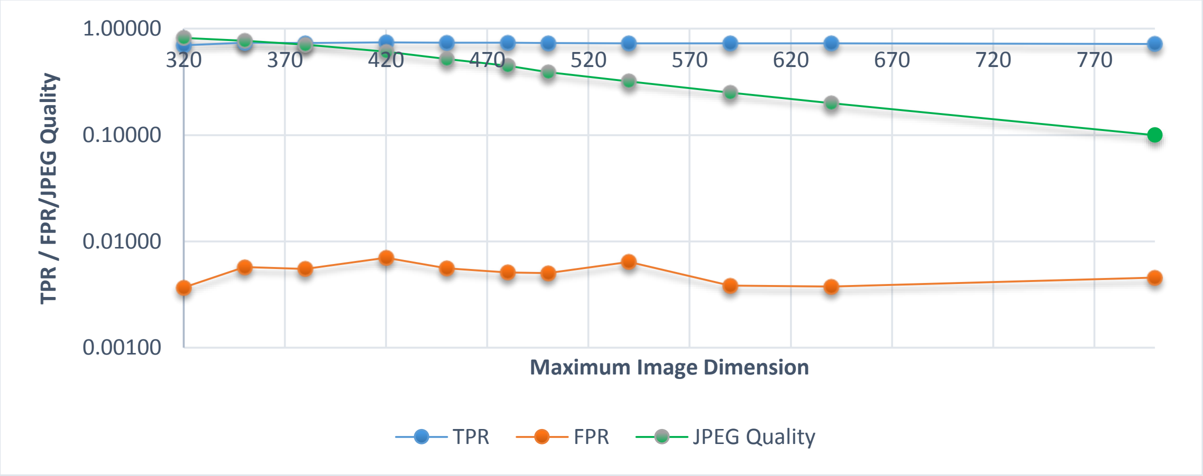


Figure 6.21 TPR, FPR and JPEG Quality vs. Image Dimension

6.4.5 Summary

The results of the above experiments indicate that it is possible to significantly reduce the file size sent over the network from **188 kB** to roughly **19 kB** without affecting the matching performance significantly. Whether this reduction in file size is achieved by image compression alone, or by a combination of image compression and resizing does not significantly affect the matching quality.

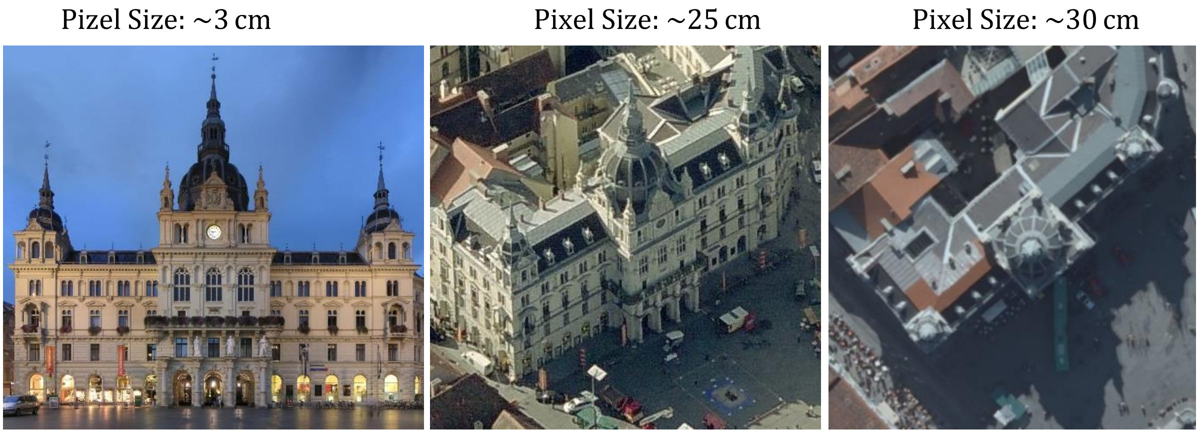
Apart from the added latency, the option of performing the feature extraction on the mobile device has the disadvantage that the extracted image features often exceed the size of the compressed input image. For example the data volume required for 1,500 interest points (4 single precision floating point numbers) together with the respective descriptor vectors (32 bytes after applying PCA) is 70.3 kB. This compares to an average size of 57 kB required for JPEG compressed images with dimensions of 500·375 pixel and a JPEG quality of 90%, and hence makes this option less feasible.

7 Image Registration in the Presence of Large Scale Differences

7.1 Dissimilar Geo-Images

The method described in Chapter 4.6 robustly matches images in cases of roughly the same scale, while differences in the capture time, illumination, image quality and pose will be acceptable. However, it is a significantly more challenging problem to perform automatic registration of images across large scale and pose differences, such as between aerial and human scale imagery. As the example in Figure 7.1 illustrates, the differences in the image resolution (GSD), perspective and image quality between aerial and terrestrial imagery still pose considerable challenges for direct image matching methods. While these methods may occasionally work for oblique aerial imagery they will most likely fail for aerial or satellite based orthophotos. In fact it is often even challenging for a human to relate scene objects across such differences.

Dissimilarities of course also exist between images for visible light cameras, thermal sensors, radar images and other imaging modalities [70]. However in internet application with user-query and index images we can focus on visible light camera images.



Pizel Size: ∼3 cm

Pixel Size: ∼25 cm

Pixel Size: ∼30 cm

Figure 7.1 The Same Building Façade in a Terrestrial, Oblique Aerial and Orthophoto Perspective (Example City Hall, Graz, Austria)

We propose a solution to a related problem, the automatic alignment of sparse point clouds obtained by SFM from terrestrial imagery [15] to oblique aerial views. The motivation to obtain such alignment is in order to place 3D reconstructions such as those obtained by Photosynth (see "Prague Old Town Square" example in Figure 7.2) at an accurate location, angle and scale on the map, allowing more intuitive visualizations and transitions between overhead- and 3D views.

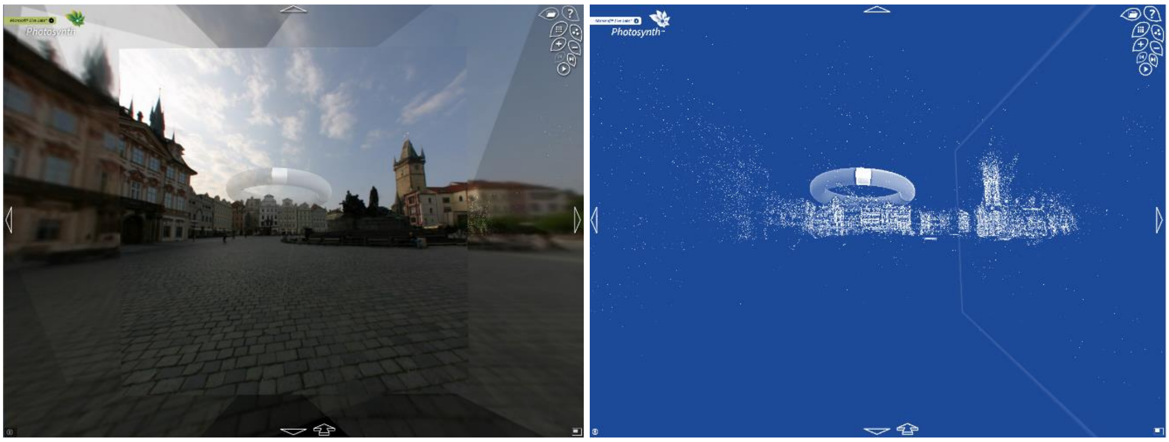


Figure 7.2 Photosynth Reconstruction of Old Town Square in Prague, Czech Republic (Left); Corresponding SFM Point Cloud (Right);

7.2 Reference Work

Our method is a modification of the work described by Kaminsky et al. [14]. The authors take advantage of the fact that vertical planes such as building façades appearing in typical urban scenes often coincide with intensity edges in the corresponding ortho-views. This fact is illustrated in Figure 7.3 using the example of an edge image (center) extracted from an ortho-image (left) using Canny edge detection [422], as well as a corresponding SFM point cloud obtained from user-photographs (right).

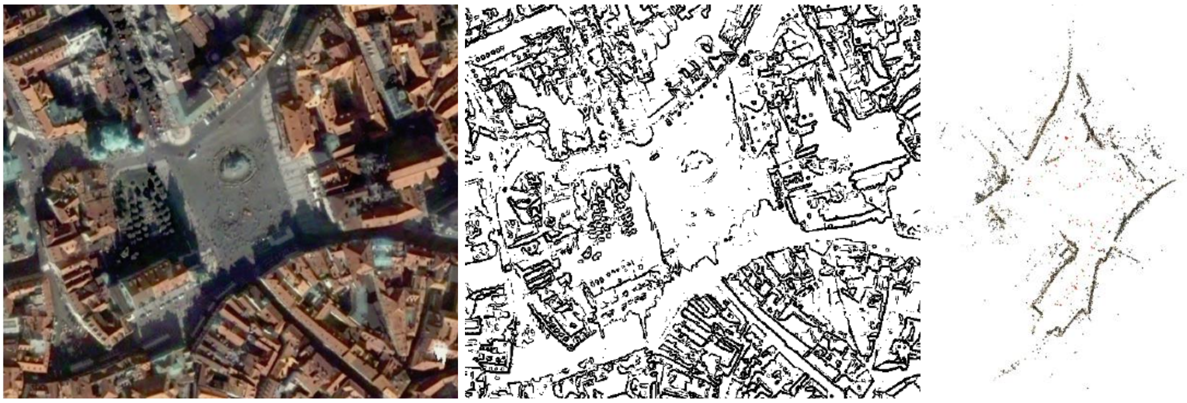


Figure 7.3 Example from Kaminsky et al. [14]: Overhead Ortho-Image of Old Town Square in Prague, Czech Republic (Left); Corresponding Edge Image (Center); Overhead View of Structure-from-Motion Point Cloud Obtained from User Photos using Photosynth (Right);

The solution proposed by Kaminsky et al. is formulated as an optimization problem, with an alignment cost function $A(i,j,\theta,s)$ containing terms for an edge cost $E(i,j,\theta,s)$ as well as a free space cost $F(i,j,\theta,s)$ where $0 \leq \alpha \leq 1$. The parameters to be optimized describe the translation (i,j), rotation (θ) and scale (s) of a similarity transform between the two top-down views. The optimal solution of $T_{i,j,\theta,s}$ with respect to the alignment cost $A(i,j,\theta,s)$ is determined by brute force search over the parameter space.

$$\boldsymbol{A(i,j,\theta,s) = \alpha * E(i,j,\theta,s) + (1-\alpha) * F(i,j,\theta,s)} \tag{7.1}$$

The edge cost is defined as the average L2 distance between (transformed) points $T_{i,j,\theta,s}(p)$ in the 3D point cloud and the nearest pixel (x,y) in a binary edge image B (Figure 7.3, right) obtained from an orthophoto by means of Canny edge detection [422]. The minimum distance for each point is computed efficiently by means of the Distance Transform [423] of B.

$$\boldsymbol{E(i,j,\theta,s) = \frac{1}{n}\sum_{p\in P} \min_{(x,y)\in B} \left\| T_{i,j,\theta,s}(p) - (x,y) \right\|_2} \tag{7.2}$$

Additionally, the cost function contains a free space term to avoid alignments to extraneous edges in the overhead image. The free space cost is based on the idea that viewing rays between the camera centers and the observed scene points (Figure 7.4, right) should not intersect with edges in the ortho-image created by occluders which would interfere with the point visibility. It is defined as the sum of the pixel-wise product of each pixel in a transformed ray image $R(x,y)$ and the binary edge image.

$$\boldsymbol{F(i,j,\theta,s) = \frac{1}{n}\sum_{x,y} R\left(T_{i,j,\theta,s}(x,y)\right) B(x,y)} \tag{7.3}$$

For the “Old Town Square” example shown above, the correct alignment result between the point cloud and the overhead image is visualized in Figure 7.4. This alignment provides significantly more accurate geocoding for the individual user images including orientation, than if only GPS data had been used [14].

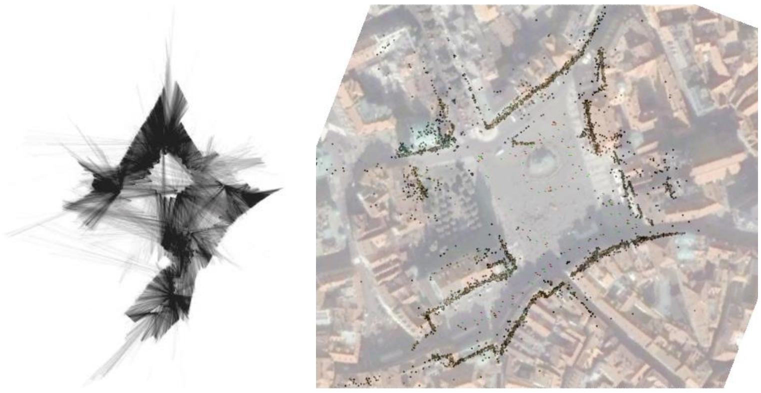


Figure 7.4 Ray Image as Reported by Kaminsky et al. Showing an Accumulation of all Viewing Rays between Camera Centers and Reconstructed Scene Points (Left); Correct Alignment Result (Right);

As pointed out in [14] this method is not limited to outdoor images but can also be applied to 3D reconstructions of indoor scenes, where the edge image may be replaced by a binary floor plan image of the building.

However, a weakness of this method is that the edge cost often forms minima plateaus in regions of densely populated edges as the average point distance is generally low. While the free space cost aims at ruling out such solutions, incorrect solutions are still likely to occur, especially for small scales where fewer edges can potentially intersect with the ray image. Additionally the assumption used for the free space cost, that edges in ortho-images coincide with occluders is often invalidated in real environments.

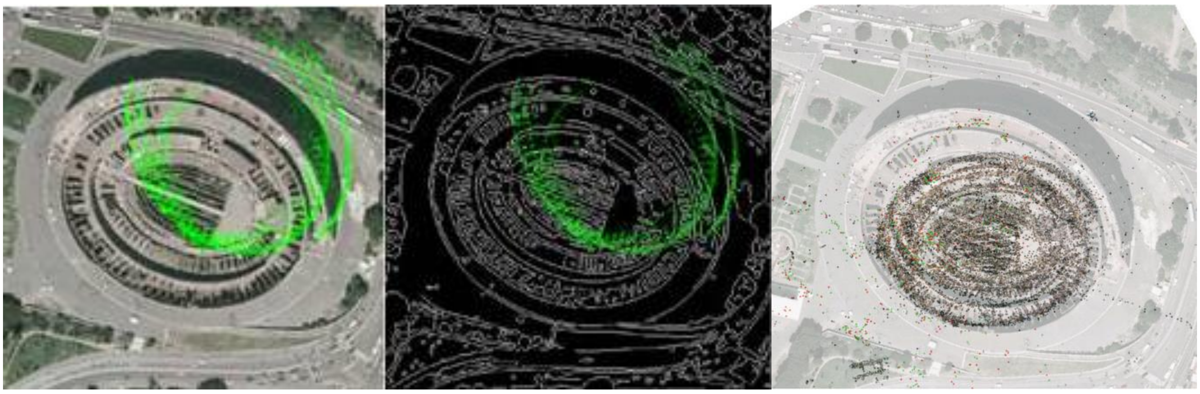


Figure 7.5 Example of Incorrect Alignment (Left and Center) using Kaminsky's method due to a High Density of Edges in Part of the Image (Location: Coliseum, Rome, Italy); Result of Mitigation using GPS Data as Additional Input as Reported by Kaminsky et al.]

Figure 7.5 (left and center) visualizes an example of an incorrect alignment using Kaminsky's method due to a high edge density within a large part of the scene. Kaminsky showed that this problem can be reduced by using additional information, such as GPS data for the user images (See Figure 7.5 right). However since these data are only available for a subset of user images, it would be desirable to improve the basic alignment algorithm itself to mitigate such problems.

7.3 A New Approach

While following the same principle, our solution differs from the reference method in several ways. The most significant difference is that we use a modified edge cost function based on the orientation of edges in the point cloud as well as the orthophoto. The original definition results in a low edge cost if any edge is close to a transformed point despite its orientation (Figure 7.6, left). Our method however, requires an edge of a similar orientation (Figure 7.6, right) near the transformed point. Additionally we punish edges which are oriented orthogonal to the expected direction of a given point by means of an orthogonal edge distance term.

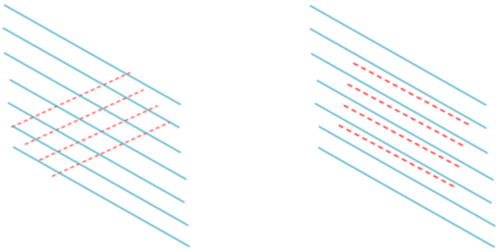


Figure 7.6 Illustration of Incorrectly Rotated Alignment (Left) between Point Cloud (Red) and Edges (Blue); Better Alignment (Right);

Further improvements aim at reducing the computational cost. Instead of computing the edge distance for each point of the point cloud individually, we first aggregate all points in top-down template images of the point clouds by computing a 2D histogram in XY-space. Similar to the ray image in the original method this image can be rotated and scaled using standard resampling methods. The edge cost is hence computed by convolution with the distance images. As suggested by Kaminsky et al. we then use the fast Fourier transform (FFT) [424] to compute the convolution instead of actually convolving the image with the relatively large templates.

The combination of the above changes helps to significantly boost the computation speed as well as the robustness of the method.

7.4 New Algorithm Description

As mentioned above the new edge cost differs from to the reference method in several ways, taking into account the orientation $\tau(p)$ of a given point as well as the orientation of the edge pixels $\tau(x,y)$. While the distance of a given point to a parallel edge with similar orientation should be small, the opposite is the case for edges that are orthogonal to the orientation of the point. Note that the rotations by $\boldsymbol{\pi}$ are considered irrelevant for this purpose as intensity gradient directions are not related to the direction of the façade. The combined cost is computed as the sum of two edge cost terms, a cost $E_{||}$ for parallel edges and a cost $E_{\perp}$ for orthogonal edges.

$$\boldsymbol{E(i,j,\theta,s) = E_{||}(i,j,\theta,s) + E_{\perp}(i,j,\theta,s)} \tag{7.4}$$

The orientations are considered parallel or orthogonal based on the following conditions:

$$\begin{aligned} &\boldsymbol{\tau(p) || \tau(x,y) \quad iff\ \cos^2\big(\tau(p) - \tau(x,y)\big) \leq \cos^2\varepsilon} \\ &\boldsymbol{\tau(p) \perp \tau(x,y) \quad iff\ \cos^2\left(\tau(p) - \tau(x,y) + \frac{\pi}{2}\right) \leq \cos^2\varepsilon} \end{aligned} \tag{7.5}$$

where $\boldsymbol{\varepsilon}$ is an tolerance window size (e.g. $\frac{\pi}{8}$). The parallel term is computed similar to Eq. (7.2) but only takes into account the shortest distance to points with a similar orientation.

$$\boldsymbol{E_{||}(i,j,\theta,s) = \frac{1}{n}\sum_{p\in P} \min_{(x,y)\in B|\tau(x,y)||\tau(p)} \left\|T_{i,j,\theta,s}(p) - (x,y)\right\|_2} \tag{7.6}$$

The orthogonal term is non-zero if the distance to an orthogonal edge is smaller than a certain radius $\boldsymbol{r}$ (e.g. 10 pixel) and increases for smaller distances.

$$\boldsymbol{E_{\perp}(i,j,\theta,s) = \frac{1}{n}\sum_{p\in P} di_{i,j,\theta,s}(p)} \tag{7.7}$$

$$\boldsymbol{di_{i,j,\theta,s}(p) = \begin{cases} r - d_{i,j,\theta,s}(p)\ iff\ d_{i,j,\theta,s}(p) < r \\ 0\ otherwise \end{cases}} \tag{7.8}$$

$$\boldsymbol{d_{i,j,\theta,s}(p) = \min_{(x,y)\in B|\tau(x,y)\perp\tau(p)} \left\|T_{i,j,\theta,s}(p) - (x,y)\right\|_2} \tag{7.9}$$

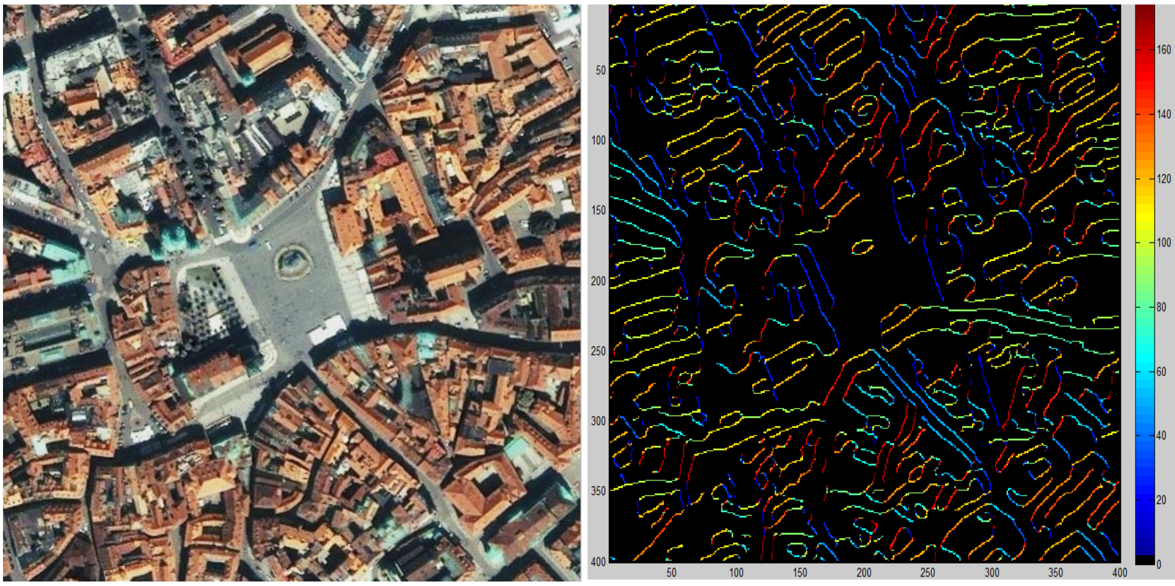


Figure 7.7 Sample Orthophoto of Prague Old Town Square from Bing Maps (Left) with Corresponding Edge Orientation Map (Right)

The orientation $\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{y})$ of a given edge pixel is computed by convolution of the edge image with a line segment (e.g. l=31 pixel) under a number of different rotations in the range $0..\pi$ and selecting the rotation angle for each pixel that gives the highest convolution value. An example edge orientation map for an orthophoto ("Prague Old Town Square") is visualized in Figure 7.7.

$$\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{y}) = \underset{\boldsymbol{\tau}\in\{\mathbf{0},\Delta\boldsymbol{\tau},\mathbf{2}*\Delta\boldsymbol{\tau},\ldots\boldsymbol{\pi}-\boldsymbol{\tau}\}}{\boldsymbol{arg}\ \mathbf{max}} (\boldsymbol{B} * \boldsymbol{L}_{\boldsymbol{\tau}})_{\boldsymbol{x},\boldsymbol{y}} \qquad (7.10)$$

The actual computation of the edge cost requires the preparation of a template image representing a projection of the 3D SFM point cloud on the X-Y plane (Figure 7.8, left). We compute this as a 2D histogram such that the intensity of a pixel corresponds to the number of points within the given cell on the X-Y plane. Thresholding to a minimum count helps to remove points not located on vertical surfaces. For this template image K we can compute the orientation map similar to the ortho photo (Figure 7.8, right).

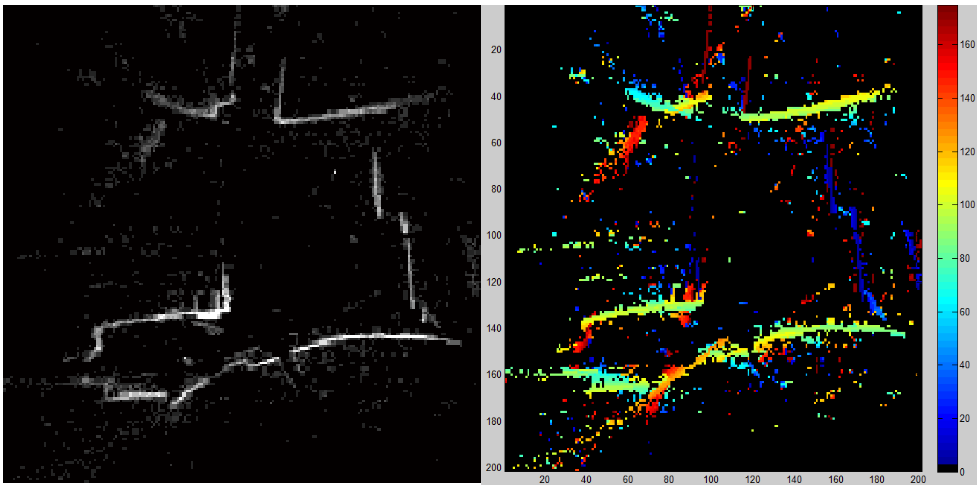


Figure 7.8 Template Image from SFM Point Cloud (Left); Corresponding Orientation Map (Right);

Based on the orientation map we hence select a subset of points within a given orientation window (e.g. $\frac{5\pi}{8}..\frac{7\pi}{8}$), leading to a filtered template image $K_{\tau_{min},\tau_{max}}$. Note that the template $K^{\theta,s}_{\tau_{min},\tau_{max}}$(Figure 7.9, left) has been rotated and scaled in accordance with the map in Figure 7.7.

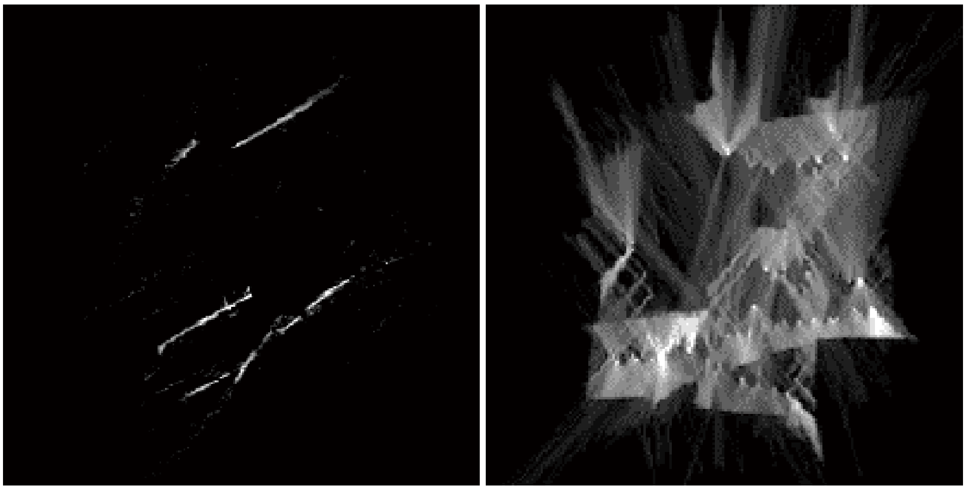


Figure 7.9 Rotated and Scaled Template Filtered by Orientation Window $\frac{5\pi}{8}..\frac{7\pi}{8}$ (Left); Ray Image (Right);

With the intention of reducing the computation time by avowing individual computation of the point distance for every point we approximate Eq. (7.7) by a sum of n convolutions (e.g. $n = 4$) each corresponding to a certain orientation window $\alpha \pm \varepsilon$.

$$\boldsymbol{E(i,j,\theta,s)} \approx \sum_{\alpha\epsilon\left\{0,\frac{\pi}{2},\pi,\frac{3\pi}{2}\right\}} \left(\boldsymbol{D}_{||_{\alpha-\varepsilon,\alpha+\varepsilon}} + \boldsymbol{D}_{\perp_{\alpha-\varepsilon,\alpha+\varepsilon}}\right) * \boldsymbol{K}^{\theta,s}_{\theta+\alpha-\varepsilon,\theta+\alpha+\varepsilon} \qquad (7.11)$$

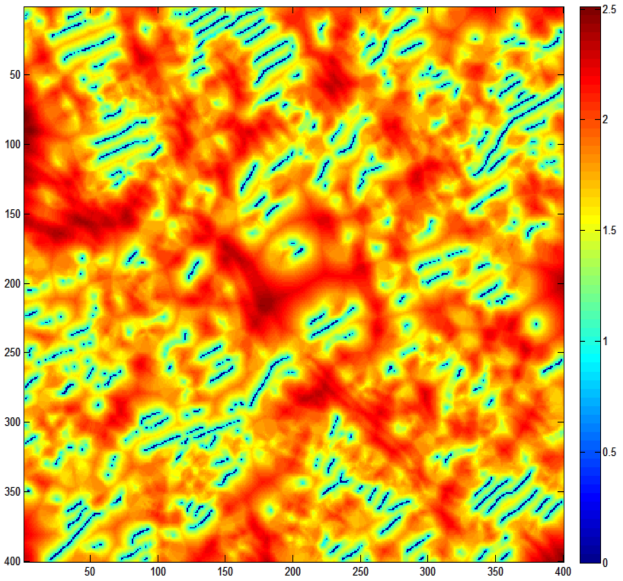


Figure 7.10 Sum of Distance Images $\boldsymbol{D}_{||_{\alpha-\varepsilon,\alpha+\varepsilon}} + \boldsymbol{D}_{\perp_{\alpha-\varepsilon,\alpha+\varepsilon}}$ for Window $\frac{5\pi}{8}..\frac{7\pi}{8}$

Both the parallel edge distance image $D_{||_{\alpha-\varepsilon,\alpha+\varepsilon}}$ as well as the orthogonal edge distance image $D_{\perp_{\alpha-\varepsilon,\alpha+\varepsilon}}$ have to be convolved by the same template image for each selection of θ, s and α. Hence the sum of the two images can be used to reduce the number of convolutions (see Figure 7.10), which is only required once for each value of α. While convolution in the image domain is feasible for small kernel sizes, it can be computationally expensive for images of 100+ pixel in size since the complexity of convolving an N·N image with an M·M image is approximately $O(N^2 + M^2)$. Therefore it is customary to instead convert the two images into the frequency domain using 2D FFT, as the equivalent operation in the Fourier domain is the product of the two Fourier transforms [424].

$$\boldsymbol{A(x,y) * B(x,y) \equiv \mathcal{F}^{-1}\Big(\mathcal{F}\big(A(x,y)\big) \cdot \mathcal{F}\big(B(x,y)\big)\Big)} \qquad \textbf{(7.12)}$$

Similarly we use FFT for convolution of the ray image in Figure 7.8 (d) with the edge image in order to compute the free space cost. Finally the alignment cost image is computed for each combination of θ and s and for each location i/j in the image, and the parameters resulting in the globally best alignment cost are chosen as the result of the search.

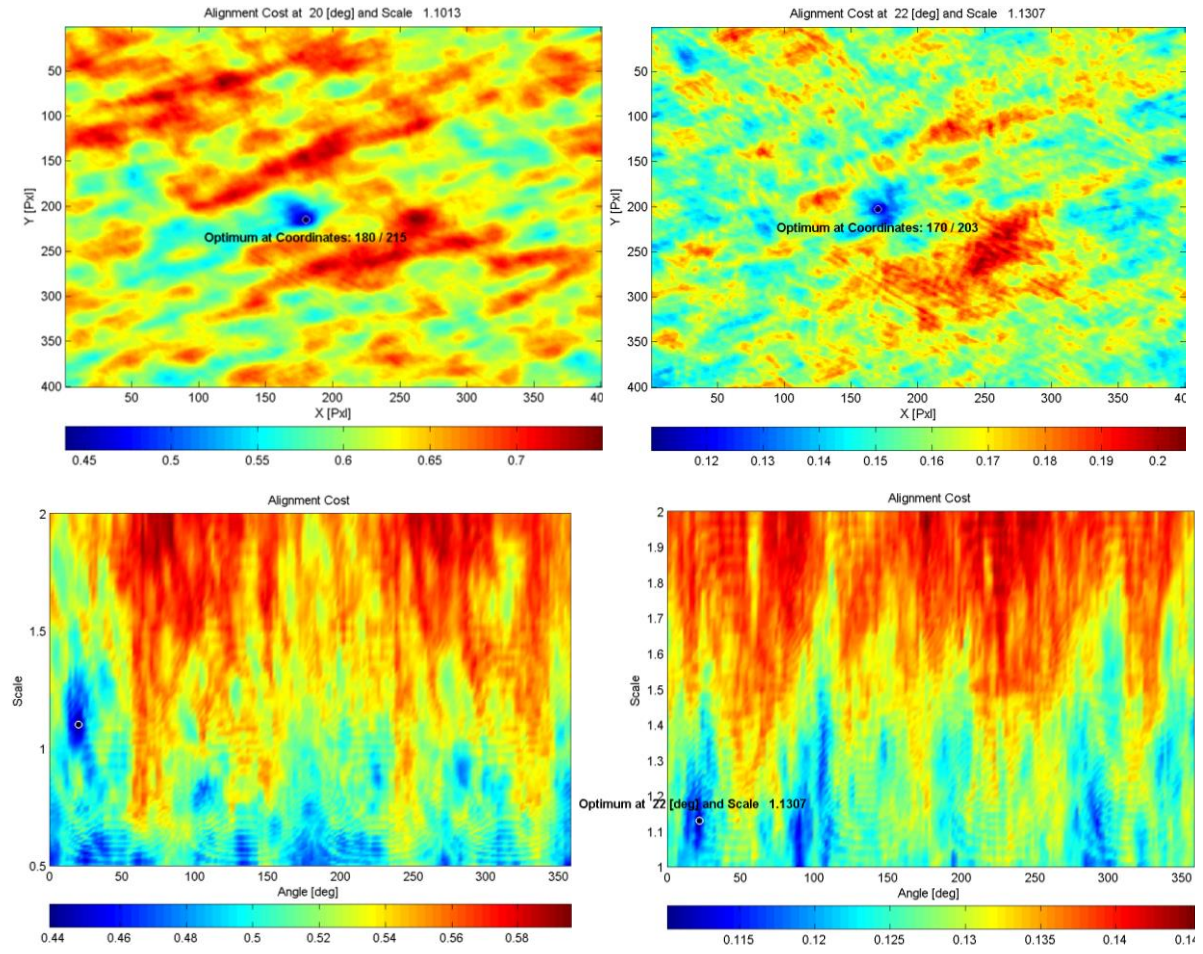


Figure 7.11 Results Shown for Modified Edge Cost (Left) and Original Edge Cost Defined by Kaminsky et al. (Right); Alignment Cost as Function of Position in Orthophoto for the Selected Scale and Orientation (Top); Best Alignment Cost as Function of Scale and Orientation (Bottom);

7.5 Experiments

We used a set of 12 different scenes which were processed and shared by users via Photosynth together with corresponding aerial views from Bing Maps for evaluating both the original version of the edge cost described by Kaminsky et al. as well as our modified version. It is important to note that we used the modified way of computing the convolution via FFT to evaluate both methods. Following the idea presented in the reference publication, we assumed that a rough prior localization including location and scale, but lacking rotation is present for each dataset, such as obtained via GPS positioning and stored in the image metadata.

Typically we compared the alignment cost across 10 different scales between 0.5 and 2 relative the prior scale estimate, 180 different rotations, 1-pixel increments (effective due to FFT) and a search area about 3 times the size (length and width) of the 3D reconstruction. We further resized the orthophoto to a size of 400·400 pixel, and used a format of 201*201 pixel for the template and ray images. Overall we therefore scanned **288 million** possible alignments. With these settings the compute time for the "Prague Old Town Square" sample used in 7.4 was **3.5 min** using Matlab on a single core of a 2006 CPU (Intel Xeon 5150). This compares to **67 min** reported in [14] for the same number of rotations and scales, but using pixel increments of 5-10 pixel within an orthophoto of a size of 1,000·1,000 pixels (**70 million** alignments overall).

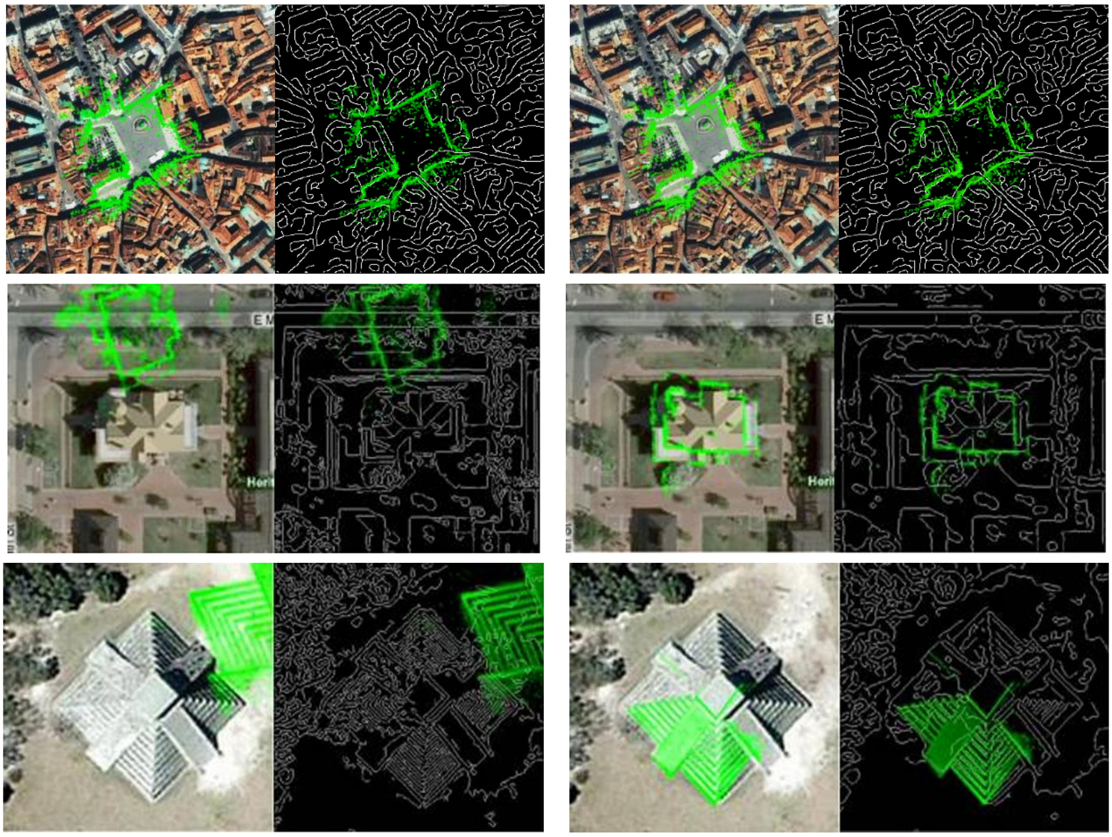


Figure 7.12 Point Clouds Superimposed on Orthophotos and Edge Images using Kaminsky's Method (Left), Our Method (Right)

The selected alignment cost image for the above example is depicted in Figure 7.11 for our proposed version of the edge cost (Top Left) as well as the original version definition by Kaminsky et al. (Top Right) as a function of the x- and y- translation. The coordinates of the best alignment have been highlighted. For this optimal choice of x and y, the alignment image as a function of scale and orientation is shown below, again for the modified and original edge cost definition. Similarly, the best scale and orientation have been highlighted. As can be seen both methods roughly find the same solution for all four similarity parameters. Nevertheless the optima in the alignment cost graphs are significantly more pronounced with the new orientation-based edge cost than for the non-orientation-based reference version.

In fact the range of scales compared had to be restricted to ≥ 1 for the latter version as otherwise miss-registrations at small scales would have occurred. This may be a consequence of the fact that the ray image obtained from the 3D reconstruction we used had significantly more rays ranging behind walls as some pictures had been captured from an elevated location behind buildings.

The correct alignment result between the point cloud and the orthophoto obtained with our method is depicted in Figure 7.12 (Top Right) compared to the result obtained with Kaminsky's method. Two more examples are given where Kaminsky's method failed to find the correct alignment, while the orientation based approach worked satisfactory.

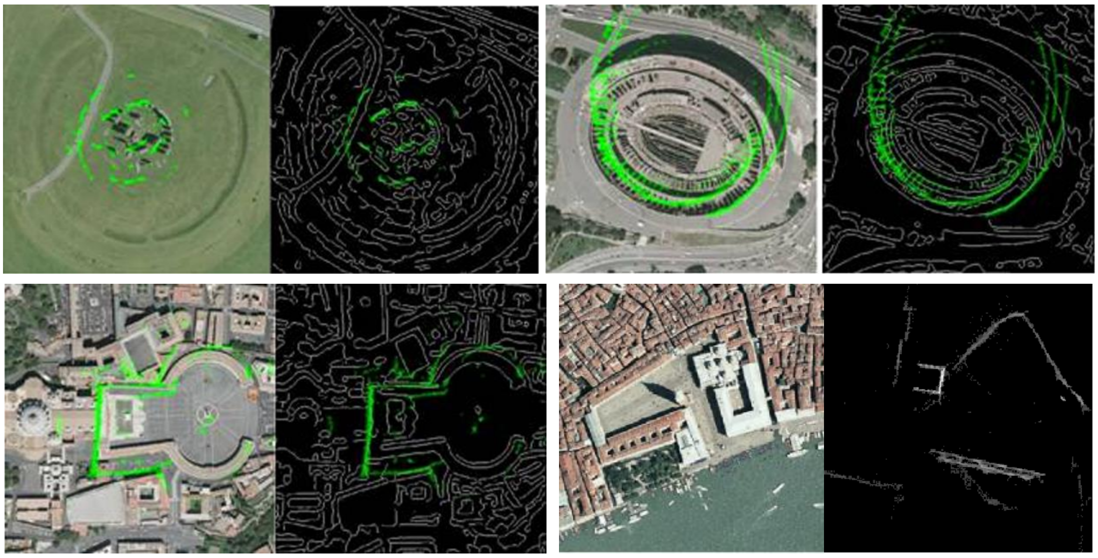


Figure 7.13 Examples of Failure Cases which could not be aligned with Either Method

Some failure cases shown in Figure 7.13 pointing out the limitations of edge based alignment. The top left dataset of Stonehenge is challenging due to the occurrence of several concentric circles. Therefore only the center but neither scale nor orientation could be determined. In case of the Colosseum dataset (top right) none of the parameters could be determined correctly although some edges align relatively well. The bottom row shows examples of datasets where the SFM reconstruction failed to estimate the scene geometry correctly. Although the alignment was partly possible in case of the Peter's square dataset (bottom left) it failed completely in case of the Marcus Square example (bottom right).

7.6 Summary

We have shown that our modifications to the method proposed by Kaminsky et al. for performing 2D alignment of SFM sparse point clouds (such as from Photosynth) to aerial photographs lead to improvements in the stability of the alignment process as well as the computational complexity. However there are still challenges with the robustness of the method due to problems with certain scene geometries as well as the quality of the SFM point clouds for many Photosynth scenes.

Potential ways of improving the automated method results would be to use local image feature based methods for aligning the various edge images. Additionally since the described method requires a relatively precise prior estimate of the location and scale of the captured 3D scene, a geospatial image search index such as described in 5.3.2 may be a helpful for obtaining such prior information in the absence of GPS data. Wendel et al. [425] have further shown that the Kaminsky-based alignment process of 3D point clouds can also be supported by using digital surface models (DSM) in addition to orthophotos, leading to significantly more robust alignments in case of objects on the ground, unoccupied space and models covering small areas.

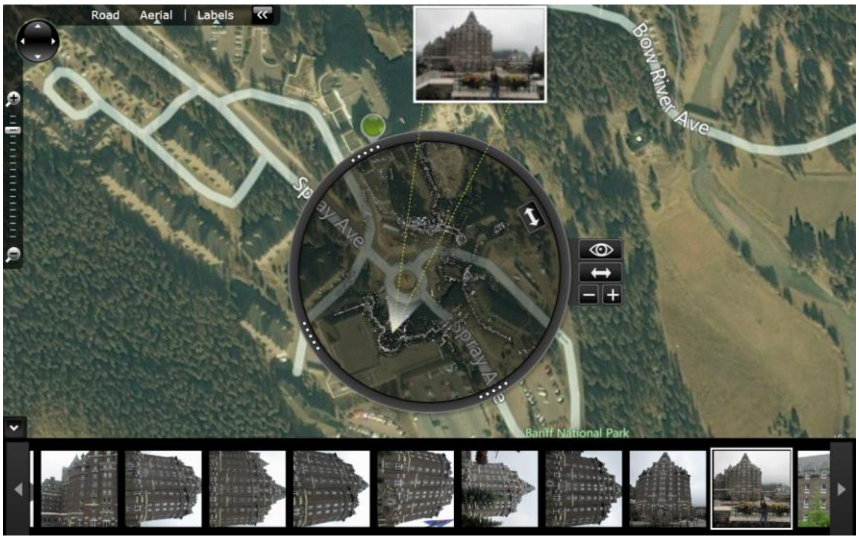


Figure 7.14 Manual Alignment Tool of Photosynth Point Clouds to Overhead Image Based on [264]. Example: Fairmont Banff Springs Hotel, Banff, AB, Canada (See Figure 3.34 for 3D View)

With the aim of enabling users to improve the geocoding of 3D reconstructions as well as individual images, Microsoft has released an update to Photosynth based on [264] and [426] which allows manual alignment of point clouds to overhead-images (See Figure 7.14). The tool allows rotation, scaling and translation, as well as the labeling of individual locations in the overhead view.

8 Conclusion and Outlook

8.1 The Internet Ecosystem for Geodata and Mapping

8.2 Contributions and Results

Internet mapping has advanced considerably over the last two decades from being a collection of static maps to a globally available and ever-growing ecosystem of geographic data, including both vector data and image data in various forms, and made available on many different devices and form factors. This thesis has addressed several key problems and research questions related to internet mapping and geospatial images in the context thereof, as well as geospatial image retrieval and location search in particular. Particular contributions were made in the area of human scale data capture and processing in preparation for use in an internet mapping platform. For this purpose, the key characteristics of human scale images and requirements for a systematic streetside image capture were identified, and hence a system design was proposed to accomplish these requirements, which was later used for streetside data capture for the Bing Maps platform. Additionally an algorithm for automatic detection and anonymization of streetside imagery was proposed and verified successfully on a representative ground truth dataset. A variant of this algorithm has also been used for privacy protection of Bing Maps streetside imagery.

We were able to show in experiments that …

Further contributions were made in the domains of geospatial image retrieval and location search. A scalable real-time image index was proposed, featuring several novel algorithmic sub-components. Specifically a variant of the Hessian interest point detector was combined with a polar orientation-based feature descriptor, both of which were evaluated and optimized for quality and speed based on several standard and custom datasets and metrics. A dynamic image ranking method based on prior work in [30] was introduced, using orientation constrains to increase the ranking success. Finally an orientation constrained pairwise post-verification algorithm feasible for central-perspective and panoramic images was presented, using a novel 1-point RANSAC algorithm [45] for robust geometric verification. The presented algorithms were tested for various applications related to geospatial imagery, such as for connecting crowdsourced user images to streetside images, or for improving geocoding of business addresses.

In order to support automatic geo-positioning of existing 3D scene reconstructions obtained by using structure from motion methods on crowdsourced images, a method based on [14] was suggested. The method automatically finds the alignment of the SFM point cloud with features in the aerial image and hence is capable of handling much larger scale differences than other methods. In addition to the original work, the new method uses rotation constraints to reduce the likelihood of miss-matches, as well as FFT for performing convolutions with a template image more efficiently. Finally we proposed a method for automatic selection of relevant geocoded images based on textual metadata associated with community images in order to augment the systematically collected aerial or streetside data. This method can be used for automatic selection and visualization of the most relevant images for a given part and scale of a map, which can be applied in densely populated areas of the map such as cities, as well as in more sparsely photographed regions.

8.3 Outlook

The ongoing evolution of internet mapping, and especially the shift towards mobile devices like cell phones or tablets, opens a large room for new applications and research in this area. The increasing sensory and processing capabilities of such devices, the advancements in mobile internet technologies as well as the emergence of cloud computing creates the basis for new types of LBS and AR. Especially AR algorithms using the cameras and inertial sensors included in mobile devices such as simultaneous localization and mapping (SLAM) will benefit substantially from these increased capabilities. Recent developments in SLAM [427] allow mapping of small to medium-sized spaces (indoor) in real-time on mobile devices, which is likely to influence how mapping will be performed in the future. A particularly interesting aspect to investigate will be the registration of small-scale maps for visual tracking with global and systematically collected map data such as streetside and indoor imagery at a higher precision than GPS/compass data allow. Such registration will be required for obtaining relevant augmentation data from global mapping databases and showing them to the users of AR systems. The image localization work described herein such as the image features and geometric verification algorithm can serve as the basis for some SLAM applications, such as for localization in existing world-maps or re-localization in a given map.

Additionally new types of image sensors such as aerial drones, surveillance cameras and the increased uses of mobile camera-enabled devices provide trillions of independent observations of the world from different locations and at different times. Each camera image covers a specific slice in the 4D space of the world, with time as the fourth dimension. Current image matching techniques such as described in this thesis manage to connect individual segments of this 4D space together, with increasing difficulty for larger scale and time differences. By creating more links between individual images, the geometric relationships between them can be determined at increasing precision. Hence by propagating pose information across images, new images can be positioned more precisely relative to a global reference frame. Information propagation can further extend to metadata such as image tags, photographer information, event names etc. While many individual pieces already exist, and work such as [11] has shown impressive results for city-scale 3D reconstructions, a common data organization scheme and image registration framework is still to be developed. Such a framework needs to cope with the massive scale of all available image sources to create an integrated comprehensive 4D world map. Creating such a framework at a truly global scale, allowing registration of images and 3D scene reconstructions representing different points in time is still an unresolved problem with large research potential.

Bibliography

[1] F. Leberl, "Time for NEO-Photogrammetry?," *GIS Development,* vol. February 2010, pp. 22-24, 2010.

[2] M. Goesele, N. Snavely, B. Curless, H. Hoppe and S. Seitz, "Multi-view stereo for community photo collections," in *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.*, 2007.

[3] F. Leberl, P. Meixner, A. Wendel and A. Irschara, "Automated photogrammetry for three-dimensional models of urban spaces," *Optical Engineering,* vol. 51, no. 2, 2012.

[4] W. Chen, A. Battestini, N. Gelfand and V. Setlur, "Visual summaries of popular landmarks from community photo collections," in *Asilomar Conference on Signals, Systems and Computers*, 2009.

[5] W. Walcher, F. Leberl and M. Gruber, "The Microsoft Global Ortho Program," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume I-4, 2012*, Melbourne, Australia, 2012.

[6] B. Carey, "How It Works: The Best View From Space Yet," 13 3 2008. [Online]. Available: http://www.popsci.com/node/19968.

[7] DigitalGlobe, "Global Basemap," 8 6 2013. [Online]. Available: http://digitalglobe.com/products/information/global-basemap#features-benefits.

[8] D. Cade, "Google Street View Has Snapped 20 Petabytes of Street Photos," 6 6 2012. [Online]. Available: http://petapixel.com/2012/06/06/google-street-view-has-snapped-20-petabytes-of-street-photos/. [Accessed 1 7 2013].

[9] Flickr, "Flickr Worldwide Meetups," [Online]. Available: http://www.meetup.com/flickr/. [Accessed 26 6 2013].

[10] L. Parfeni, "Flickr Boasts 6 Billion Photo Uploads," 5 8 2011. [Online]. Available: http://news.softpedia.com/news/Flickr-Boasts-6-Billion-Photo-Uploads-215380.shtml.

[11] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz and R. Szeliski, "Building Rome in a day," in *Proceedings of IEEE International Conference on Computer Vision (ICCV '09)*, 2009.

[12] M. Shirai, M. Hirota, S. Yokoyama, N. Fukuta and H. Ishikawa, "Discovering multiple HotSpots using geo-tagged photographs," in *20th International Conference on Advances in Geographic Information Systems*, 2012.

[13] P. Cho, N. Snavely and R. Anderson, "3D exploitation of large urban photo archives," in *SPIE Defense, Security, and Sensing, International Society for Optics and Photonics*, 2010.

[14] R. Kaminsky, N. Snavely, S. Seitz and R. Szeliski, "Alignment of 3D Point Clouds to Overhead Images," in *Second IEEE Workshop on Internet Vision*, 2009.

[15] N. Snavely, S. Seitz and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *Proceedings of SIGGRAPH '06*, 2006.

[16] X. Li, C. Wu, C. Zach, S. Lazebnik and J. Frahm, "Modeling and recognition of landmark image collections using iconic scene graphs," *Computer Vision–ECCV,* pp. 427-440, 2008.

[17] Y. Li, D. Crandall and D. Huttenlocher, "Landmark classification in large-scale image collections," in *IEEE 12th International Conference on Computer Vision, 2009*, 2009.

[18] R. Andrews, "Facebook has 220 Billion of Your Photos to Put on Ice," 17 10 2012. [Online]. Available: http://benton.org/node/137340.

[19] M. Brenn, "Eiffel Tower," 5 9 2010. [Online]. Available: http://www.flickr.com/photos/28145073@N08/5237960579/in/photolist-8YRVca-dPwm9k-dPwibR-dPBVwu-dPwj1t-e5E3mY-dSeVWx-aDX3VL-aqpkn5-bABsEq-7Q25nj-dPBXv1-dPC1nJ-dPC1CG-dPBYbC-dPBYmm-dPBZL7-dPBWFG-dPBUUC-dPBVfy-dPwnAT-dPwkqT-dPwnZt-dPwo9H-dPwjwk-dPwhRe-dPBVpw-. [Accessed 3 7 2013].

[20] J. Mayer, "Eiffel Tower," 12 9 2009. [Online]. Available: http://www.flickr.com/photos/8584048@N05/4048783042/in/photolist-7aM59b-7bKvue-7h6Q4U-7h6QZo-7hCNps-7m3mcs-7nxcUn-8YRVca-dPwm9k-dPwibR-dPBVwu-dPwj1t-e5E3mY-dSeVWx-aDX3VL-aqpkn5-bABsEq-7Q25nj-dPBXv1-dPC1nJ-dPC1CG-dPBYbC-dPBYmm-dPBZL7-dPBWFG-dPBUUC-dPBVfy-d. [Accessed 3 7 2013].

[21] Daily Mail, "When past and present collide: London scenes of today overlaid with historical photographs taken in the exact same spot," 29 6 2013. [Online]. Available: http://www.dailymail.co.uk/news/article-2351509/Unique-phone-app-shows-photographs-iconic-London-locations-overlaid-historical-views-exact-spots.html. [Accessed 2 7 2013].

[22] *Emmeline Pankhurst being arrested while trying to present a petition to the King.* [Art]. © Museum of London, 1914.

[23] *Shoe shine Piccadilly: 1953.* [Art]. © Henry Grant Collection/Museum of London, 1953.

[24] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, second edition, Cambridge University Press, ISBN: 0521540518, 2004.

[25] M. Kroepfl, Y. Wexler and E. Ofek, "Efficiently locating photographs in many panoramas," in *Proceedings of GIS - Workshop on Advances in Geographic Information Systems, ACM SigSpatial 2010*, San Jose, CA, 2010.

[26] I. Omer, M. Kroepfl, E. Ofek, K. Muktinutalapati and M. Tabb, "Identifying Plane Outliers In Scenes Using Re-Projection For Privacy Protection," *IP.com Prior Art Database Disclosure,* 2009.

[27] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven and L. Vincent, "Large-scale Privacy Protection in Google Street View," in *Proceedings of International Converence of Computer Vision (ICCV '09)*, 2009.

[28] J. Turnbull, "Street View Face Blurring," 14 5 2008. [Online]. Available: http://googlesightseeing.com/2008/05/street-view-face-blurring/.

[29] M. Peterson, "A critical assessment of maps and the internet," *Revista Brasileira de Cartografia No 60/03,* pp. 287-291, 2007.

[30] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR '06)*, 2006.

[31] L. Vincent, "Taking Online Maps Down to Street Level," *Computer,* pp. 118-120, 12 2007.

[32] L. Graham, "Mobile Mapping Systems Overview," 3 2010. [Online]. Available: www.asprs.org/a/webinarseries/Overview_of_Mobile_Mapping.pdf .

[33] M. Kroepfl, M. Gruber, M. J. Ponticelli, S. L. Lawler, J. Bauer, F. Leberl, K. Karner, Z. Cosic, H. Hegenbarth, G. Kimchi and J. C. Curlander, "Data Capture System". US Patent 2010/0182396, 22 7 2010.

[34] M. Kroepfl, M. J. Ponticelli, H. Hegenbarth, G. Kimchi and J. C. Curlander, "Determining Exposure Time in a Digital Camera". US Patent 2010/0182444 , 22 7 2010.

[35] M. Kroepfl, J. Pehserl, J. Bauer, S. L. Lawler, G. Kimchi and J. Curlander, "Determining velocity using multiple sensors". US Patent 8244431, 14 8 2012.

[36] M. Kroepfl, G. Neuhold, S. Bernoegger, M. J. Ponticelli, J. Pehserl, G. Kimchi and J. C. Curlander, "Synchronization of multiple data sources to a common time base". US Patent 7974314, 5 7 2011.

[37] M. Kroepfl, J. Bauer, G. Neuhold, S. Bernoegger, G. Kimchi and J. C. Curlander, "Determining Trigger Rate for a Digital Camera". US Patent 8284250, 22 7 2010.

[38] H. Grabner and H. Bischof, "On-line boosting and vision," in *Conference on Computer Vision and Pattern Recognition*, 2006.

[39] E. Ofek, M. Kroepfl, I. Omer, M. Tabb and K. Muktinutalapati, "Detection of Objects in Images". US Patent 20100246890, 30 9 2010.

[40] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of International Conference on Computer Vision (ICCV) 2003*, 2003.

[41] D. Wagner and D. Schmalstieg, "First steps towards handheld augmented reality," in *7th Intl. Symposium on Wearable Computers (ISWC'03), October 2003*, White Plains, NY, 2003.

[42] R. Lakemond, C. Fookes and S. Sridharan, "Negative determinant of hessian features," *International Conference on Digital Image Computing Techniques and Applications (DICTA), 2011,* pp. 530-535, 2011.

[43] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27(10), 2005,* pp. 1615-1630, 2005.

[44] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM, Volume 18, Issue 9, September 1975,* 1975.

[45] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM, 24(6), June 1981,* p. 381–395, 1981.

[46] "Bing Maps Streetside Photos CTP," [Online]. Available: http://www.bing.com/maps/explore/#/bqx21pyfpdn6h2ly.

[47] M. Kroepfl, E. Ofek, Y. Wexler, D. Wysocki and G. Kimchi, "Geocoding by Image Matching". US Patent 8189925 B2, 29 5 2012.

[48] D. Buchmueller, M. Kroepfl, D. Nistér, V. Cugunovs, R. Sagula, B. Agüera y Arcas, S. Fynn and E. Ofek, "Spatial Image Index and Associated Updating Functionality". US Patent 20120155778, 21 6 2012.

[49] E. Ofek, M. Kroepfl, J. Walker, G. Ramos and B. Aguera y Arcas, "Viewing Media in the Context of Street-Level Images". US Patent 20110173565, 14 7 2011.

[50] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision, 65(102),* pp. 43-72, 2005.

[51] H. Stewénius and D. Nistér, "Recognition Benchmark Images," 2006. [Online]. Available: http://www.vis.uky.edu/~stewe/ukbench/.

[52] E. Alpaydin, Introduction to machine learning, MIT Press, 2004.

[53] J. Luis Borges, Artist, *Of exactitude in science.* [Art]. 1946.

[54] F. Leberl, "Human Habitat Data in 3D for the Internet," in *Computer Vision, Imaging and Computer Graphics*, Berlin, Springer, 2010, pp. 3-17.

[55] Microsoft, "MSN Virtual Earth Gives People an Immersive Way to Search, Discover and Explore Their World Online," 24 7 2005. [Online]. Available: http://www.microsoft.com/en-us/news/press/2005/jul05/07-24VirtualEarthBetaPR.aspx.

[56] Microsoft, "Microsoft Adds 3-D City Models to Live Search," 6 11 2006. [Online]. Available: http://www.microsoft.com/en-us/news/press/2006/nov06/11-06VE3DLaunchPR.aspx.

[57] "Bing Maps," [Online]. Available: http://www.bing.com/maps/.

[58] "MapQuest," [Online]. Available: http://www.mapquest.com/.

[59] "Google Maps," [Online]. Available: http://maps.google.com/.

[60] Nokia, "Nokia Here," [Online]. Available: http://here.com. [Accessed 1 7 2013].

[61] Herold, "Herold.at," [Online]. Available: http://www.herold.at/. [Accessed 5 7 2013].

[62] BEV, "Amap.at," [Online]. Available: http://www.amap.at. [Accessed 5 7 2013].

[63] Baidu, "Baidu Maps," [Online]. Available: http://maps.baidu.com/. [Accessed 5 7 2013].

[64] T. Frenz, "What is a "Map" and What are maps used for?," 2002. [Online]. Available: http://wwws.phil.uni-passau.de/histhw/tutcarto/english/index-frames-en.html.

[65] A. Pike, D. Hoffmann, M. Garcia-Diez, B. Pettitt, J. Alcolea, R. De Balbin, C. Gonzalez-Sainz, C. de las Heras, J. Lasheras, R. Montes and J. Zilhao, "U-Series Dating of Paleolithic Art in 11 Caves in Spain," *Science, Vol. 336 no. 6087,* pp. 1409-1413, June 2012.

[66] T. Campbell and M. Destombes, "The earliest printed maps," British Library, London, 1987.

[67] J. Aber, "Brief History of Maps and Cartography," 2008. [Online]. Available: http://academic.emporia.edu/aberjame/map/h_map/h_map.htm. [Accessed 5 7 2013].

[68] T. Poiker and I. Crain, "Canadian GIS," 2012. [Online]. Available: http://www.thecanadianencyclopedia.com/articles/geographic-information-systems.

[69] A. Ducher, "Photogrammetry—The largest operational application of remote sensing," *Photogrammetria,* vol. 41, no. 2, pp. 72-82, 1987.

[70] T. Lillesand, R. Kiefer and J. Chipman, Remote Sensing and Image Interpretation, John Wiley & Sons Inc., 2008.

[71] A. Felch, J. Nageswaran, A. Chandrashekar, J. Furlong, N. Dutt, R. Granger and A. Veidenbaum, "Accelerating brain circuit simulations of object recognition with cell processors," in *International workshop on innovative architecture for future generation high-performance processors and systems*, 2007.

[72] A. Hodgkiss, "The Bildkarten of Hermann Bollmann," *The International Journal for Geographic Information and Geovisualization,* vol. 10, no. 2, pp. 133-145, 1973.

[73] J. Short, "The World Through Maps," in *Mapping the Mordern World*, Firefly Books, 2003, pp. 184-185.

[74] J. Kopf, M. Agrawala, D. Bargeron, D. Salesin and M. Cohen, "Automatic generation of destination maps," *ACM Transactions on Graphics (TOG),* vol. 29, no. 6, p. 158, 2010.

[75] J. Hughes, A. Van Dam, M. McGuire, D. Sklar, J. Foley, S. Feiner and K. Akeley, Computer Graphics - Principles and Practice, Addison-Wesley, 2013.

[76] J. Döllner, "Non-Photorealistic 3D Geovisualization," in *Multimedia Cartography*, Berlin, Springer, 2007, pp. 229-240.

[77] Oxford Dictionaries, "Oxford Dictionaries," [Online]. Available: http://oxforddictionaries.com/. [Accessed 22 6 2013].

[78] "Map," 2013. [Online]. Available: http://en.wikipedia.org/wiki/Map.

[79] H. Saggs, Civilization Before Greece and Rome, Yale University Press, 1989.

[80] Spertus, "Cartography as Art and Science: Advent of the Printing Press," [Online]. Available: http://www.spertus.edu/exhibits/cartography-art-and-science-advent-printing-press-0. [Accessed 6 7 2013].

[81] A. Madrigal, "How Google Builds Its Maps—and What It Means for the Future of Everything," 6 9 2012. [Online]. Available: http://www.theatlantic.com/technology/archive/2012/09/how-google-builds-its-maps-and-what-it-means-for-the-future-of-everything/261913/. [Accessed 6 7 2013].

[82] B. Virgin, "TRAVEL: The decline of Washington state paper map," 4 3 2012. [Online]. Available: http://www.tri-cityherald.com/2012/03/04/1850726/travel-the-decline-of-washington.html. [Accessed 5 7 2013].

[83] G. Sterling, "Google Introduces Offline Maps For Mobile, Claims a Billion Users Globally For Maps, Earth," 12 6 2012. [Online]. Available: http://searchengineland.com/live-blogging-the-google-maps-next-dimension-event-123617. [Accessed 6 7 2013].

[84] Encyclopedia Britannica, "Mercator Projection," [Online]. Available: http://www.britannica.com/EBchecked/topic/375638/Mercator-projection. [Accessed 7 7 2013].

[85] Google, "Google Maps Coordinates," [Online]. Available: https://developers.google.com/maps/documentation/javascript/v2/overlays#Google_Maps_Coordinates. [Accessed 7 7 2013].

[86] J. Schwartz, "Bing Maps Tile System," n.d.. [Online]. Available: http://msdn.microsoft.com/en-us/library/bb259689.aspx.

[87] Maps International, "Personalised Historic World Map - 1965," 7 12 2010. [Online]. Available: http://www.mapsinternational.co.uk/blog/index.php/2010/12/07/personalised-gift-idea/. [Accessed 7 7 2013].

[88] weather.com, "weather.com," [Online]. Available: weather.com/. [Accessed 7 7 2013].

[89] M. Rosenberg, "Map Colors - The Role of Colors on Maps," 13 8 2007. [Online]. Available: http://geography.about.com/od/understandmaps/a/mapcolors.htm. [Accessed 7 7 2013].

[90] R. Davidson, "Reading Topographic Maps," 2008. [Online]. Available: http://www.map-reading.com/intro.php. [Accessed 5 7 2013].

[91] Wikipedia, "Category:Map types," [Online]. Available: http://en.wikipedia.org/wiki/Category:Map_types. [Accessed 7 7 2013].

[92] M. Peterson, "Trends in Internet Map Use," in *Proceedings of the 18th International Cartographic Conference*, 1997.

[93] P. Meixner and F. Leberl, "3-Dimensional Building Details from Aerial Photography for Internet Maps," *Remote Sensing 2011,3,* pp. 721-751, 2011.

[94] M. Peterson, "Trends in Internet Map Use.: A Second Look," in *Proceedings of the 19th International Cartographic Conference* , 1999.

[95] R. Galbreach and B. Harrison.US Patent 4539701 - Photogrammetric Stereoplotter, 1985.

[96] E. Baltsavias, "On the performance of photogrammetric scanners," *D. FRITSCH/R. SPILLER (EDS.), PHOTOGRAMMETRIC WEEK'99,* 1999.

[97] F. Leberl, M. Gruber, M. Ponticelli, S. Bernoegger and R. Perko, "The Ultracam large format aerial digital camera system," in *Proceedings of the American Society For Photogrammetry & Remote Sensing*, Anchorage, AL, 2003.

[98] D. Caitilin, "GIS Data Explored – Vector and Raster Data," 1 May 2012. [Online]. Available: http://www.gislounge.com/geodatabases-explored-vector-and-raster-data/.

[99] G. Lawton, "New ways to build rich internet applications," *Computer,* vol. 41, no. 8, pp. 10-12, 2008.

[100] Wikipedia, "Microsoft Streets & Trips," [Online]. Available: http://en.wikipedia.org/wiki/NextBase. [Accessed 7 7 2013].

[101] Microsoft, "MapPoint," [Online]. Available: http://www.microsoft.com/mappoint/en-us/home.aspx. [Accessed 7 7 2013].

[102] M. Peterson, "A decade of maps and the internet," in *Proceedings of the XXII International Cartographic Conference*, Coruna, Spain, 2005.

[103] M. Peterson, "Trends in Internet and ubiquitous cartography," *Cartographic Perspectives no 61,* pp. 36-49, 2008.

[104] WolframAlpha, "Internet users per capita," n.d.. [Online]. Available: http://www.wolframalpha.com/input/?i=internet+users+per+capita+.

[105] Netcraft, "January 2013 Web Server Survey," Jan 2013. [Online]. Available: http://news.netcraft.com/archives/2013/01/07/january-2013-web-server-survey-2.html.

[106] M. Kroepfl, D. Buchmueller and F. Leberl, "Online Maps and Cloud-Supported Location-Based-Services Across a Manifold of Devices," in *Proceedings of the Conference of the International Society of Photogrammetry and Remote Sensing (ISPRS) Commission IV, WG IV/5*, Melbourne, 2012.

[107] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski and M. Zaharia, "A view of cloud computing," *Communications of the ACM,* vol. 53, no. 4, pp. 50-58, 2010.

[108] Gartner, "Forecast Overview: Public Cloud Services, Worldwide, 2011-2016, 4Q12 Update," 8 February 2013. [Online]. Available: http://my.gartner.com/portal/server.pt?open=512&objID=202&&PageID=5553&mode=2&in_hi_userid=2&cached=true&resId=2332215&ref=AnalystProfile.

[109] Google, "A Brief History of Google Maps," n.d.. [Online]. Available: http://www.google.com/help/maps/helloworld/behind/history.html.

[110] S. Shankland, "Navteq to supply Microsoft with 3D map data," 7 12 2009. [Online]. Available: http://news.cnet.com/8301-30685_3-10410320-264.html.

[111] S. Shankland, "Nokia, Microsoft becoming Windows Phone bedfellows," 11 2 2011. [Online]. Available: http://news.cnet.com/8301-30685_3-20031468-264.html#!.

[112] K. Hafner, S. Rai and A. Kramer, "Google Offers a Bird's-Eye View, And Some Governments Tremble," *New York Times,* 20 12 2005.

[113] J. Tariman, "Nokia's HERE to go 3D," 15 11 2012. [Online]. Available: http://hitechtabai.wordpress.com/2012/11/15/nokias-here-to-go-3d/. [Accessed 7 7 2013].

[114] Microsoft, "Bing Maps Publishes Equivalent of 100,000 DVD's of Bird's Eye Imagery," 11 6 2013. [Online]. Available: http://www.bing.com/blogs/site_blogs/b/maps/archive/2013/06/11/largest-shipment-of-bird-s-eye-100-000-dvds-of-imagery.aspx. [Accessed 1 7 2013].

[115] Google Maps, "Cars, Trikes, and More," 6 6 2013. [Online]. Available: http://maps.google.com/help/maps/streetview/learn/cars-trikes-and-more.html.

[116] Bing Maps, "Streetside - The true-to-life experience for explorers everywhere," 2011. [Online]. Available: http://www.microsoft.com/maps/streetside.aspx. [Accessed 25 6 2013].

[117] K. Komenda, "The Map API Showdown," 21 1 2011. [Online]. Available: http://www.klauskomenda.com/archives/2011/01/21/the-map-api-showdown/. [Accessed 7 7 2013].

[118] E. Mahaney, "Future of Paper Maps," [Online]. Available: http://geography.about.com/od/understandmaps/a/Future-Of-Paper-Maps.htm. [Accessed 5 7 2013].

[119] A. Tinworth, "Digital Surrey: the Ed Parsons Map Project," 26 7 2012. [Online]. Available: http://www.onemanandhisblog.com/archives/2012/07/digital_surry_the_ed_parsons_map_project.html. [Accessed 5 7 2013].

[120] S. Garfield, "Why modern maps put everyone at the centre of the world," 12 10 2012. [Online]. Available: http://www.bbc.co.uk/news/magazine-19908848. [Accessed 5 7 2013].

[121] F. Novelli, "Platform Substitution and Cannibalization: The Case of Portable Navigation Devices," in *Software Business*, Heidelberg, Springer, 2012, pp. 141-153.

[122] P. Connolly and D. Bonte, "Personal Navigation Devices," ABI Research, 2011.

[123] AAA, "AAA," [Online]. Available: http://www.aaa.com. [Accessed 5 7 2013].

[124] Rand McNally, "Rand McNally," [Online]. Available: http://www.randmcnally.com. [Accessed 5 7 2013].

[125] PRWEB, "Atlas and Map Publishers in the US Industry Market Research Report Now Available from IBISWorld," 6 9 2012. [Online]. Available: http://www.prweb.com/releases/2012/9/prweb9873905.htm. [Accessed 5 7 2013].

[126] Go-Gulf, "Smartphone Users Around the World – Statistics and Facts [Infographic]," 2012. [Online]. Available: http://www.go-gulf.com/blog/smartphone/.

[127] S. Helal, R. Bose and W. Li, Mobile Platforms and Development Environments, Morgan & Claypool, 2013.

[128] A. Schill, "LTE, WiMAX and 4G," 10 10 2012. [Online]. Available: http://www.rn.inf.tu-dresden.de/lectures/MCaMC/04_LTE_and_beyond.pdf. [Accessed 15 6 2013].

[129] L. June, "Apple replaces Google Maps with its own maps, turn-by-turn navigation and traffic info," 11 6 2012. [Online]. Available: http://www.theverge.com/2012/6/11/3076745/apple-maps-google-maps-replacement. [Accessed 7 7 2013].

[130] M. Liedtke, "Google Maps return to iPhone with new mobile app," 13 12 2012. [Online]. Available: http://news.yahoo.com/google-maps-return-iphone-mobile-053341410.html. [Accessed 7 7 2013].

[131] S. Kang, T. Kim and S. Jang, "Location-based services: enabling technologies and a concierge service model," in *Societies and Cities in the Age of Instant Access*, Springer, 2007, pp. 227-239.

[132] B. Bonetti, "Nokia City Lens brings augmented reality to Nokia Lumia," 8 5 2012. [Online]. Available: http://conversations.nokia.com/2012/05/08/nokia-city-lens-brings-augmented-reality-to-nokia-lumia/. [Accessed 7 7 2013].

[133] J. Jiang, G. Han and J. Chen, "Modelling Turning Restrictions in Traffic Networks for Vehicle Navigation System," *International Archives of Photogrammetry, Remote Sensing and Information Sciences,* vol. 34, no. 4, pp. 106-110, 2002.

[134] B. Hofmann Wellenhof, K. Legat and M. Wieser, Navigation, Wien: Springer, 2003.

[135] O. Kersting and J. Döllner, "Interactive 3D visualization of vector data in GIS," in *ACM international symposium on Advances in geographic information systems*, 2002.

[136] ESRI, "What is Raster Data," 22 9 2008. [Online]. Available: http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=What_is_raster_data?.

[137] H. Samet, Multidimensional and Metric Data Structures, San Francisco: Diane D. Cerra, 2006.

[138] R. Finkel and J. Bentley, "Quad Trees: A Data Structure for Retrieval on Composite Keys," *Acta Informatica 4,* pp. 1-9, 1974.

[139] T. Grubesic and A. Murray, "Assessing positional uncertainty in geocoded data," *Proceedings of the 24th Urban Data Management Symposium,* 2004.

[140] D. Roongpiboonsopit and H. Karimi, "Comparative evaluation and analysis of online geocoding services," *International Journal of Geographical Information Science, 24(7),* pp. 1081-1100, 2010.

[141] C. Amelunxen, "An Approach to gecoding based on volunteered Spatial Data," *Geoinformatik,* pp. 7-12, 2010.

[142] MelissaData, "MelissaData," [Online]. Available: http://www.melissadata.com/. [Accessed 7 7 2013].

[143] K. Pridal, "Tiles à la Google Maps: Coordinates, Tile Bounds and Projection," 2008. [Online]. Available: http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/.

[144] J. Kanner, "TomTom Agrees to Acquire Tele Atlas for EU2 Billion (Update10)," 23 7 2007. [Online]. Available: http://www.bloomberg.com/apps/news?pid=newsarchive&sid=agT1Po33faG4&refer=home.

[145] J. Erwing, "Nokia to Pay $8.1 Billion for Navteq," 1 10 2007. [Online]. Available: http://www.businessweek.com/stories/2007-10-01/nokia-to-pay-8-dot-1-billion-for-navteqbusinessweek-business-news-stock-market-and-financial-advice.

[146] V. Gough, "Updating the Maps of the United Kingdom, Germany, Finland and Sweden," 8 12 2011. [Online]. Available: http://google-latlong.blogspot.com/2011/12/updating-maps-of-united-kingdom-germany.html.

[147] NavTeq, "Server-Based Applications (Europe)," 18 5 2007. [Online]. Available: http://contracts.onecle.com/telenav/navteq-license-6-2007-05-18.shtml.

[148] "Crowdsourcing," n.d.. [Online]. Available: http://en.wikipedia.org/wiki/Crowdsourcing.

[149] C. Heipke, "Crowdsourcing geospatial data," *ISPRS Journal of Photogrammetry and Remote Sensing, 65(6),* pp. 550-557, 2010.

[150] "Creative Commons," [Online]. Available: http://creativecommons.org/.

[151] M. Haklay, "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets.," *Environment and planning. B, Planning & design 37(4),* p. 682, 2008.

[152] B. Ciepłuch, R. Jacob, P. Mooney and A. Winstanley, "Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps," *Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resuorces and Enviromental Sciences,* p. 337, 2010.

[153] D. Paine and J. Kiser, Aerial Photography and Image Interpretation, Third Edition, Hoboken, New Jersey: John Wiley & Sons Inc., 2012.

[154] M. Brown and D. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision,* vol. 74, no. 1, pp. 59-73, 2007.

[155] T. Reichenbacher and S. De Sabbata, "Geographic relevance: different notions of geographies and relevancies," *SIGSPATIAL Special, 3(2),* pp. 67-70, 2011.

[156] WikiPaintings, [Online]. Available: www.wikipaintings.org.

[157] R. Miau, "blogspot.com," [Online]. Available: http://manitobatripruikaiwismer.blogspot.com/2013_06_30_archive.html.

[158] "Wikipedia, Broken Chair," [Online]. Available: http://en.wikipedia.org/wiki/Broken_Chair.

[159] B. Epshtein, E. Ofek, Y. Wexler and P. Zhang, "Hierarchical photo organization using geo-relevance," in *Proceedings of ACM Sig-Spatial Conference 2007*, 2007.

[160] I. Nourbakhsh, R. Sargent, A. Wright, K. Cramer, B. McClendon and M. Jones, "Mapping disaster zones," *Nature,* vol. 439, pp. 787-788, 2006.

[161] Bing Maps, "New Hurricane Sandy app uses High-Resolution Imagery," 26 11 2012. [Online]. Available: http://www.bing.com/blogs/site_blogs/b/maps/archive/2012/11/26/new-hurricane-sandy-app-uses-high-resolution-imagery.aspx.

[162] R. Gonzalez and R. Woods, Digital Image Processing - Second Edition, Upper Saddle River, New Jersey: Pretence-Hall, 2002.

[163] "Photosynth," [Online]. Available: htp://www.photosynth.net/.

[164] Spiegel, "Privacy Concerns: German Towns Saying 'Nein' to Google 'Street View'," 29 9 2008. [Online]. Available: http://www.spiegel.de/international/germany/privacy-concerns-german-towns-saying-nein-to-google-street-view-a-581177.html.

[165] A. Kashyap, "Microsoft Nixes Bing Streetside Imagery Offline In Germany After Privacy Complaints," 12 5 2012. [Online]. Available: http://news.ebrandz.com/microsoft/2012/5497-microsoft-nixes-bing-streetside-imagery-offline-in-germany-after-privacy-complaints.html.

[166] "Flickr," [Online]. Available: http://www.flickr.com/.

[167] "Photobucket," [Online]. Available: http://www.photobucket.com/.

[168] "Panoramio," [Online]. Available: http://www.panoramio.com/.

[169] "Facebook," [Online]. Available: http://www.facebook.com/.

[170] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz and A. Wiechert, "Point Clouds: Lidar versus 3D Vision," *Photogrammetric Engineering and Remote Sensing,* vol. 76, no. 10, pp. 1123-1134, 2010.

[171] J. Thurgood, M. Gruber and K. Karner, "Multi-ray matching for automated 3D object modeling," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 35,* pp. 1682-1777, 2004.

[172] C. Zach, T. Pock and H. Bischof, "A Globally Optimal Algorithm for Robust TV-L," *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007,* pp. 1-8, 2007.

[173] N. Haala and M. Rothermel, "Dense Multi-Stereo Matching for High Quality Digital Elevation Models," *Photogrammetrie-Fernerkundung-Geoinformation,* vol. 4, pp. 331-343, 2012.

[174] C. Strecha, O. Kueng and P. Fua, "Automatic Mapping from Ultra-Light UAV Imagery," *infoscience.epfl.ch,* 2012.

[175] A. Irschara, V. Kaufmann, M. Klopschitz, H. Bischof and F. Leberl, "Towards fully automatic photogrammetric reconstruction using digital images taken from UAVs," in *ISPRS TC VII Symposium—100 Years ISPRS*, 2010.

[176] B. Reitinger, M. Sormann, L. Zebedin, B. Schachinger, M. Hoefler, R. Tomasi and M. Gruber, "ULTRAMAP V3–A REVOLUTION IN AERIAL PHOTOGRAMMETRY," in *International Society of Photogrammetry and Remote Sensing*, Melbourne, 2012.

[177] W. Schickler and A. Thorpe, "Operational procedure for automatic true orthophoto generation," *International Archives of Photogrammetry and Remote Sensing,* vol. 32, pp. 527-532, 1998.

[178] J. Rau, N. Chen and L. Chen, "True orthophoto generation of built-up areas using multi-view images," *Photogrammetric Engineering and Remote Sensing,* vol. 68, no. 6, pp. 581-588, 2002.

[179] G. Zhou, W. Chen, J. Kelmelis and D. Zhang, "A comprehensive study on urban true orthorectification," in *IEEE Transactions on Geoscience and Remote Sensing*, 2005.

[180] M. Nielsen, "True Orthophoto Generation," Technical University of Denmark, Lyngby, 2004.

[181] D. Greig, B. Porteous and A. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society Series B, 51,* p. 271–279, 1989.

[182] C. Brenner, N. Haala and D. Fritsch, "Towards fully automated 3D city model generation," Automatic Extraction of Man-Made Objects from Aerial and Space Images, (III), 2001.

[183] L. Zebedin, J. Bauer, K. Karner and H. Bischof, "Fusion of feature-and area-based information for urban buildings modeling from aerial imagery," in *Computer Vision–ECCV*, 2008.

[184] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation," *International journal of computer vision,* vol. 99, no. 1, pp. 69-85, 2012.

[185] P. Meixner, F. Leberl and M. Brédif, "3D roof details by 3D aerial vision," in *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

[186] C. Iovan, "Detection and Characterisation of Vegetation in Urban Areas from High Resolution Aerial Imagery," L'University Pierre et Marie Curie, Paris, 2009.

[187] Microsoft, "UltraCamEagle - Technical Specifications," 2011. [Online]. Available: http://download.microsoft.com/download/7/4/3/743EFD09-258B-4BFA-8D56-3148C60DD137/UCAMTechnicalDocuments/UltraCamEagle-Specs.pdf.

[188] R. Hartley and R. Gupta, "Linear pushbroom camera," in *Computer Vision—ECCV'94*, 1994.

[189] D. Yocky, "Multiresolution wavelet decomposition image merger of Landsat thermatic mapper and SPOT panchromatic data.," *Photogrammetric Engineering & Remote Sensing, 62(9),* p. 1067–1074, 1996.

[190] W. Shi, C. Zhu and X. Yang, "Multi-band wavelet for fusing SPOT panchromatic and multispectral images," *Photogrammetric Engineering & Remote Sensing,69(5),* p. 513–520, 2003.

[191] R. Perko, Computer Vision for Large Format Aerial Cameras - Dissertation, Graz, 2004.

[192] Sat Imaging Corp, "Satellite Imaging Sensors," [Online]. Available: www.satimagingcorp.com/satellite-sensors.html. [Accessed 23 06 2013].

[193] NASA, "Sensor Specifications Eros-A," [Online]. Available: http://geo.arg.nasa.gov/sge/health/sensors/eros.html. [Accessed 23 6 2013].

[194] P. Cheng and C. Chappel, "DEM Generation Using QuickBird Stereo Data," *PCI Geomatics,* pp. 36-38, 3 2006.

[195] Vexcel Corporation, "Vexcel Apex Ground Station," 2005. [Online]. Available: http://www.infoserve.co.jp/doc/VEXCEL/apexGS.pdf. [Accessed 24 6 2013].

[196] F. Bignalet-Cazalet, S. Baillarin, D. Greslou and C. Panem, "Automatic and generic mosaicing of satellite images," in *2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Honolulu, 2010.

[197] M. Gianinetto and M. Scaioni, "Automated geometric correction of high-resolution pushbroom satellite data," *Photogrammetric engineering and remote sensing,* vol. 74, no. 1, p. 107, 2008.

[198] Aerometrex, "From the size of a mountain to the size of a 50 cents coin," 4 4 2012. [Online]. Available: http://aerometrex.com.au/blog/?p=427.

[199] A. Hirano, R. Welch and H. Lang, "Mapping from ASTER stereo image data: DEM validation and accuracy assessment," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 57, no. 5, pp. 356-370, 2003.

[200] M. Gruber and R. Ladstätter, "Geometric issues of the digital large format aerial camera UltraCamD," in *International Calibration and Orientation Workshop EuroCOW*, 2006.

[201] Blom, "Blom and Microsoft renew and extend agreement valued at USD 10 - 14 million," 4 3 2010. [Online]. Available: http://www.blomasa.com/news/blom-and-microsoft-renew-and-extend-agreement-valued-at-usd-10---14-million.html.

[202] M. Lemmens, "Digital Aerial Cameras - New Developments," GIM, 28 3 2011. [Online]. Available: http://www.gim-international.com/issues/articles/id1695-Digital_Aerial_Cameras.html. [Accessed 24 6 2013].

[203] Microsoft, "Microsoft Acquires Vexcel Corp., a Worldwide Leader in Imagery and Remote Sensing Technology," 4 5 2006. [Online]. Available: http://www.microsoft.com/en-us/news/press/2006/may06/05-04VexcelPR.aspx.

[204] D. Riley, "Google Acquires ImageAmerica," 21 7 2007. [Online]. Available: http://techcrunch.com/2007/07/21/google-acquires-imageamerica/.

[205] DigitalGlobe, "Ortho Vision Premium," [Online]. Available: http://img.docstoccdn.com/thumb/orig/85968589.png. [Accessed 24 6 2013].

[206] Podoski Consulting, "History of Aerial Photography," 6 6 2013. [Online]. Available: http://www.papainternational.org/history.asp.

[207] M. Gruber and F. Leberl, "High Quality Photogrammetric Scanning for Mapping," in *China International Geoinformatics Industry, Technology and Equipment Exhibition*, Beijing, 2000.

[208] G. Petrie and S. Walker, "Airborne Digital Image Technology: A New Overview," *The Photogrammetric Record 22(119),* pp. 203-225, 2007.

[209] PhotoScience, "Aerial LiDAR," PhotoScience, [Online]. Available: http://www.photoscience.com/services/aerial-lidar. [Accessed 25 6 2013].

[210] M. Lemmens, "Airborne Lidar Scanners," 2 3 2011. [Online]. Available: http://www.gim-international.com/issues/articles/id1667-Airborne_Lidar_Scanners.html. [Accessed 15 6 2013].

[211] A. Large and G. Herritage, "Laser Scanning - Evolution of the Discipline," in *Laser Scanning for the Environmental Sciences*, Wiley, 2009, pp. 1-20.

[212] C. Lloyd and P. Atkinson, "Deriving DSMs from LiDAR data with kriging," *International Journal of Remote Sensing,* vol. 23, no. 12, pp. 2519-2524, 2002.

[213] M. Bartels, H. Wei and D. Mason, "DTM generation from LIDAR data using skewness balancing," in *18th International Conference on Pattern Recognition, ICPR*, 2006.

[214] A. Kobler, N. Pfeifer, P. Ogrinc, L. Todorovski, K. Oštir and S. Džeroski, "Repetitive interpolation: A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain," *Remote Sensing of Environment,* vol. 108, no. 1, pp. 9-23, 2007.

[215] N. Haala and C. Brenner, "Generation of 3D city models from airborne laser scanning data," in *EARSEL Workshop on LIDAR remote sensing on land and sea*, Tallinn, 1997.

[216] H. Maas and G. Vosselmann, "Two algorithms for extracting building models from raw laser altimetry data," *ISPRS Journal of photogrammetry and remote sensing,* vol. 54, no. 2, pp. 153-163, 1999.

[217] G. Vosselman, "Fusion of laser scanning data, maps, and aerial photographs for building reconstruction," in *IEEE International Conference Geoscience and Remote Sensing Symposium, IGARSS'02*, 2002.

[218] C. Frueh, R. Sammon and A. Zakhor, "Automated texture mapping of 3D city models with oblique aerial imagery," in *2nd International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT*, 2004.

[219] F. Prandi, C. Achille, R. Brumana, F. Fassi and L. Fregonese, "LiDAR and Pictometry Images - Integrated Use for 3D Model Generation," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* p. 37, 2008.

[220] R. Ruck and O. Smith, "KS-127A long range oblique reconnaissance camera for RF-4 aircraft," in *In 24th Annual Technical Symposium.*, 1980.

[221] G. Petrie, "Systematic Oblique Aerial Photography," 15 9 2008. [Online]. Available: http://www.petriefied.info/Petrie_Croatia_Multiple_Oblique_Camera_Systems2.pdf.

[222] Y. Wang, S. Schultz and F. Giuffrida, "Pictometry's Proprietary Airborne Digital Imaging System and its Application in 3D City Modelling," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 37, pp. 1065-1069, 2008.

[223] Microsoft Vexcel, "New Oblique Microsoft UltraCam Osprey Introduced," 26 03 2013. [Online]. Available: http://www.gim-international.com/news/remote_sensing/photogrammetry/id7283-new_oblique_microsoft_ultracam_osprey_introduced.html.

[224] P. Mishra, E. Ofek and G. Kimchi, "Validation of Vector Data using Oblique Images," in *ACM Sigspatial Conference*, 2008.

[225] P. Müller, G. Zeng, P. Wonka and L. Van Gool, "Image-based procedural modeling of facades," *ACM Transactions on Graphics,* vol. 26, no. 3, p. 85, 2007.

[226] Adam Technology, "UAV," 2013. [Online]. Available: http://www.adamtech.com.au/3dm/UAV.html.

[227] Microdrones GmbH, "MD4-200 Unmanned Aerial Vehicle (UAV) Specifications and Data Sheet," 30 7 2012. [Online]. Available: http://www.unmanned.co.uk/autonomous-unmanned-vehicles/uav-data-specifications-fact-sheets/md4-200-unmanned-aerial-vehicle-uav-specifications-and-data-sheet/. [Accessed 27 6 2013].

[228] "Pix4D," [Online]. Available: http://www.pix4d.com/.

[229] D. Küng, C. Strecha, A. Beyeler, J. Zufferey, D. Floreano, P. Fua and F. Gervaix, "The Accuracy of Automatic Photogrammetric Techniques on Ultra-Light UAV Imagery," in *Proceedings of the International Conference on Unmanned Aerial Vehicle in Geomatics (UAV-g),* Zuerich, 2011.

[230] R. Jeronimo, "Pix4D, la mejor herramienta en fotogrametria digital," 29 6 2012. [Online]. Available: http://www.gonostopografia.com/2012/06/pix4d-la-mejor-herramienta-en-fotogrametria-digital/.

[231] L. Zebedin, A. Klaus, B. Gruber-Geymayer and K. Karner, "Towards 3D map generation from digital aerial images," *ISPRS Journal of Photogrammetry and Remote Sensing, 60(6),* pp. 413-427, 2006.

[232] M. Gruber and B. Reitinger, "UltraCamX and an New Way of Photogrammetric Processing," in *ASPRS Conference*, Portland, 2008.

[233] M. Cheves, "The Sky Is No Longer The Limit—UltraCam User Group Meeting," 19 3 2012. [Online]. Available: http://www.lidarnews.com/content/view/8900/198/.

[234] G. Cristóbal, P. Schelkens and H. Thienpont, Optical and Digital Image Processing, Weinheim: Wiley-VCH, 2011.

[235] M. Kroepfl, Pulse Pattern Generator for Digital CCD Camera - Implementation by Using a CPLD - Diploma Thesis, Graz: Campus 02, 2003.

[236] R. Perko, A. Klaus and M. Gruber, "Quality comparison of digital and film-based images for photogrammetric purposes," in *International Society of Photogrammetry and Remote Sensing*, Istanbul, 2004.

[237] F. Leberl and M. Gruber, "Flying the new large format digital aerial camera Ultracam," in *Photogrammetric Week*, 2003.

[238] M. Sonka, V. Hlavac and R. Boyle, Image Processing, Analysis and Machine Vision, Brooks/Cole Publishing Company, 1998.

[239] R. Ladstaedter and M. Gruber, "Geometric Aspects Concerning the Photogrammetric Workflow of the Digital Aerial Camera UltraCam-X," in *Conference of the International Society of Photogrammetry and Remote Sensing (ISPRS)*, Beijing, 2008.

[240] M. Hirschmugl, M. Ofner, J. Raggam and M. Schardt, "Single tree detection in very high resolution remote sensing data," *Remote Sensing of Environment,* vol. 110, no. 4, pp. 533-544, 2007.

[241] Wikipedia, "False color," [Online]. Available: http://en.wikipedia.org/wiki/False-color. [Accessed 25 6 2013].

[242] M. Kroepfl, M. Gruber and E. Kruck, "Geometric Calibration of the Digital Large Format Aerial Camera UltraCamD," in *Proceedings of the Conference of the International Society for Photogrammetry and Remote Sensing (ISPRS) Commission 1, WG 1/2*, Istanbul, 2004.

[243] GIP Aalen, "The world of BINGO," [Online]. Available: http://www.gip-aalen.de/index.htm. [Accessed 27 6 2013].

[244] M. Gruber and M. Kroepfl, "UltraCamX Calibration Report," 18 1 2007. [Online]. Available: http://www.keystoneaerialsurveys.com/pdf/UCXCalReport_30914061_V10.pdf. [Accessed 24 6 2013].

[245] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon and J. Weaver, "Google street view: Capturing the world at street level," *Computer, 43(6),* pp. 32-38, 2010.

[246] Facet Technologies, "Location-Based Services," [Online]. Available: http://www.facet-tech.com/Location_Based_Services/Virtual_Tours.htm.

[247] infoUSA, "infoUSA Adds Storefront Pictures and Geocodes of Millions of Businesses to Database," 27 7 2004. [Online]. Available:

http://www.businesswire.com/news/home/20040727005973/en/infoUSA-Adds-Storefront-Pictures-Geocodes-Millions-Businesses. [Accessed 25 6 2013].

[248] IGI, "IGI Street Mapper," [Online]. Available: http://www.igi.eu/streetmapper.html.

[249] "FARO Focus3D," [Online]. Available: http://www.faro.com/en-US/products/3d-surveying/faro-focus3d/applications-us#main.

[250] "Riegl VMX-450," [Online]. Available: http://products.rieglusa.com/category/mobile-scanners.

[251] M. Lemmens, "Terrestrial Laser Scanners," 8 2009. [Online]. Available: http://www.gim-international.com/files/productsurvey_v_pdfdocument_33.pdf. [Accessed 25 6 2013].

[252] "Applanix Pos LV 120," [Online]. Available: http://www.applanix.com/products/land/pos-lv.html. [Accessed 25 6 2013].

[253] "360 Cities," [Online]. Available: http://www.360cities.net/.

[254] Photosynth Mobile, "Mobile Panoramas," Microsoft, 28 2 2013. [Online]. Available: http://blogs.msdn.com/b/photosynth/archive/2013/02/28/photosynth-for-windows-phone-8-is-here.aspx.

[255] Microsoft Research, "Image Composite Editor," 26 5 2011. [Online]. Available: http://research.microsoft.com/en-us/um/redmond/groups/ivm/ice/. [Accessed 25 6 2013].

[256] B. Donovan, B. Frischer, M. Gross, B. Gross, E. Johnson, M. Worthy, L. Reilly, W. Rourk, K. Stuart, M. Tuite, T. Watson, S. Wells and M. Wessel, "IATH Best Practices Guide to Digital Panotamic Photography," 2007. [Online]. Available: http://www2.iath.virginia.edu/panorama/section1.html. [Accessed 26 6 2013].

[257] D. Martin and J. Martin, "Converting Large Panoramic Images to PhotoOverlays for Google Earth with 360Cities," [Online]. Available: http://www.360cities.net/panoramic-photo-tutorials/converting-panorama-images-to-photo-overlays-for-google-earth. [Accessed 26 6 2013].

[258] T. Krueger, "Spice Shop in the Souq of Aswan (Egypt)," 3 12 2013. [Online]. Available: http://photosynth.net/view.aspx?cid=b88d629d-b934-4b60-b621-12fb4bb682a3.

[259] J. Frahm, P. Fite-Georgel and E. Dunn, "State of the art and challenges in crowd sourced modeling," in *Photogrammetric Week*, 2011.

[260] "Digital Photography," [Online]. Available: http://en.wikipedia.org/wiki/Digital_photography#Applications_and_considerations.

[261] Kodak, "Kodak Milestones 1990-1999," 2012. [Online]. Available: http://www.kodak.com/ek/US/en/Our_Company/History_of_Kodak/Milestones_-_chronology/1990-1999.htm.

[262] X. Jin, A. Gallagher, L. Cao, J. Luo and J. Han, "The wisdom of social multimedia: using flickr for prediction and forecast," in *International conference on Multimedia*, 2010.

[263] A. Chitu, "Flickr Adds Geotagging," 29 8 2006. [Online]. Available: http://googlesystem.blogspot.com/2006/08/flickr-adds-geotagging.html.

[264] B. Chen, E. Ofek, D. Gedye, J. Dughi, M. Dawson and J. Podolak, "Transitioning between Top-Down Maps and Local Navigation of Reconstructed 3-D Scenes". US Patent 0187723, 4 8 2011.

[265] D. Gupta, "http://www.ditii.com/2009/10/08/photosynth-overhead-view-feature/," 8 10 2009. [Online]. Available: http://www.ditii.com/2009/10/08/photosynth-overhead-view-feature/.

[266] S. Seitz, "Visit global landmarks with photo tours in Google Maps," Google, 2012 25 4. [Online]. Available: http://google-latlong.blogspot.com/2012/04/visit-global-landmarks-with-photo-tours.html. [Accessed 2013 15 6].

[267] Photobucket, "Photobucket's 2012 Year in Review," 12 2012. [Online]. Available: http://blog.photobucket.com/photobuckets-2012-year-in-review/.

[268] E. Protalinski, "Facebook officially welcomes Instagram into its family as it passes 5 billion photo milestone," 6 9 2012. [Online]. Available: http://thenextweb.com/facebook/2012/09/06/facebook-officially-welcomes-instagram-family-passes-5-billion-photo-milestone/.

[269] E. Kern, "Instagram hits 5 billion photos shared as it closes Facebook deal," 6 9 2012. [Online]. Available: http://gigaom.com/2012/09/06/instagram-hits-5-billion-photos-shared-as-it-closes-facebook-deal/.

[270] F. Michel, "How many photos are uploaded to Flickr every day, month, year?," 20 3 2012. [Online]. Available: http://www.flickr.com/photos/franckmichel/6855169886/.

[271] A. Smith, "Facebook reaches one billion users," 4 10 2012. [Online]. Available: http://money.cnn.com/2012/10/04/technology/facebook-billion-users/index.html.

[272] S. Bennett, "Twitter On Track For 500 Million Total Users By March, 250 Million Active Users By End Of 2012," 13 1 2012. [Online]. Available: http://www.mediabistro.com/alltwitter/twitter-active-total-users_b17655.

[273] J. Sileo, "Geotag, You're It! Disabling GPS Coordinates," [Online]. Available: http://www.sileo.com/geotagging/. [Accessed 26 6 2013].

[274] D. Zielstra and H. Hochmair, "Positional accuracy analysis of Flickr and Panoramio images for selected world regions," *Journal of Spatial Science,* pp. 1-23, 2013.

[275] P. Zandbergen, " Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning," *Transactions in GIS, 13(s1),* pp. 5-26, 2009.

[276] P. Dana, "GPS Error Sources," 1 5 2000. [Online]. Available: http://www.colorado.edu/geography/gcraft/notes/gps/gps.html.

[277] G. Reitmayr and T. Drummond, "Initialization for visual tracking in urban environments," in *Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR '07)*, 2007.

[278] O. Gaggi, "Discovering local attractions from geo-tagged photos," in *28th Annual ACM Symposium on Applied Computing*, 2013.

[279] J. Vuurens, A. de Vries and C. Eickhoff, "How much spam can you take? an analysis of crowdsourcing results to increase accuracy," in *ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR'11)*, 2011.

[280] I. Ivanov, P. Vajda, J. Lee, P. Korshunov and T. Ebrahimi, "Geotag Propagation with User Trust Modeling," in *Social Media Retrieval*, London, Springer, 2013, pp. 283-304.

[281] S. Ahern, N. Naarman, R. Nair and J. Yang, "World Explorer: Visualizing Aggregate Data from Unstructured Text in Geo-Referenced Collections," in *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2007.

[282] D. Liu, X. Hua, L. Yang, M. Wang and H. Zhang, "Tag ranking," in *18th international conference on World wide web*, 2009.

[283] D. Buchmueller, M. Kroepfl and F. Schaffalitzky, "Spatial Attribute Ranking Value Index". US Patent 8429156, 23 4 2013.

[284] J. Rocchio, "Relevance Feedback in Information Retrieval," *SMART Retrieval System Experimens in Automatic Document Processing,* 1971.

[285] A. Micarelli, F. Gasparetti, F. Sciarrone and S. Gauch, "Personalized search on the world wide web," in *The adaptive WEB*, Berlin, Springer Verlag, 2007, pp. 195-230.

[286] W. Mason and D. Watts, "Financial incentives and the performance of crowds," *ACM SigKDD Explorations Newsletter,* vol. 11, no. 2, pp. 100-108, 2010.

[287] J. Rogstadius, V. Kostakos, A. Kittur, B. Smus, J. Laredo and M. Vukovic, "An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets," in *Fifth International AAAI Conference on Weblogs and Social Media*, Barcelona, 2011.

[288] Foursquare, "Earning points and badges," [Online]. Available: http://support.foursquare.com/entries/215406-earning-points-and-badges. [Accessed 26 6 2013].

[289] K. Tuite, N. Snavely, D. Hsiao, N. Tabing and Z. Popovic, "PhotoCity: Training Experts at Large-scale Image Acquisition Through a Competitive Game," in *Computer Human Interaction (CHI)*, 2011.

[290] Google, "Field Trip," 2013 16 5. [Online]. Available: https://play.google.com/store/apps/details?id=com.nianticproject.scout. [Accessed 2013 26 6].

[291] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez and C. Schmid, "Aggregating Local Image Descriptors into Compact Codes," in *Conference on Computer Vision and Pattern Recognition (CVPR '10)*, 2010.

[292] M. Weiser, " Ubiquitous computing," *Computer,* vol. 26(10), pp. 71-72, 1993.

[293] Microsoft, "Microsoft Tag," Microsoft, [Online]. Available: http://tag.microsoft.com/home.aspx. [Accessed 15 6 2013].

[294] P. Viola and M. Jones, "Robust real-time face detection," *Proceedings of International Conference on Computer Vision (ICCV),* 2001.

[295] D. Buchmueller, A. Akbarzadeh and M. Kroepfl, "Using Photograph to Initiate and Perform Action". US Patent 20130156274, 20 6 2013.

[296] OrCam, "OrCam," [Online]. Available: http://www.orcam.com/. [Accessed 7 7 2013].

[297] Google, "Google Glass," [Online]. Available: http://www.google.com/glass/start/what-it-does/. [Accessed 7 7 2013].

[298] Bing VIsion, "Windows Phone 8 and Bing," 29 10 2012. [Online]. Available: http://www.bing.com/blogs/site_blogs/b/search/archive/2012/10/29/windows-phone-8-and-bing.aspx.

[299] Item Master, "Item Master," [Online]. Available: http://www.itemmaster.com/index.htm. [Accessed 26 6 2013].

[300] Koelln, "product and image database," [Online]. Available: http://www.koelln.com/international/brand__products/product_and_image_database.html. [Accessed 26 6 2013].

[301] Scaping Web Content Databases, "ISBN Database of 12.8 Million Books (Title, Author, ISBN, Cover Images and More)," [Online]. Available: http://www.scrapingweb.com/databases/bookbyisbn-database.html. [Accessed 26 6 2013].

[302] K. Sfikas, I. Pratikakis and T. Theoharis, "3D object retrieval via range image queries based on SIFT descriptors on panoramic views," in *Eurographics 2012 Workshop on 3D Object Retrieval*, 2012.

[303] R. Ohbuchi and Y. Kurita, "Local geometry adaptive manifold re-ranking for shape-based 3D object retrieval," in *20th ACM international conference on Multimedia*, 2012.

[304] Google, "Google Goggles," 2011. [Online]. Available: http://www.google.com/mobile/goggles.

[305] T. O'Brien, "Google Goggles Solves Sudoku in Record Time," 11 1 2011. [Online]. Available: http://www.switched.com/2011/01/11/google-goggles-solves-sudoku-in-record-time/.

[306] H. Ueno, A. Kawai and H. Sato, "Development of a mobile mapping system for use in emergency gas line maintenance vehicles," *Vehicle Navigation and Information Systems Conference, 1989. Conference Record (pp. 177-184),* pp. 177-184, 1989.

[307] K. Novak, "Mobile mapping technology for GIS data collection," *Photogrammetric engineering and remote sensing, 61(5),* pp. 593-501, 1995.

[308] M. Maresch, "Linear CCD array based recordings of buildings for digital models, PhD Thesis," Graz, 1997.

[309] N. El-Sheimy, "An Overview of Mobile Mapping Systems," *FIG Working Week 2005 and GSDI-8,* 6 2005.

[310] C. Toth, "R&D of mobile LiDAR mapping and future trends," *In Proceeding of ASPRS 2009 Annual Conference,* 2009.

[311] V. Tao and J. Li, Advances in Mobile Mapping Systems, London, UK: Taylor & Francis Group, 2007.

[312] Digital Trends, "Mapping the Roads, one Off-Ramp at a Time," 14 8 2006. [Online]. Available: http://www.digitaltrends.com/features/mapping-the-roads-one-off-ramp-at-a-time/.

[313] BusinessSigns.net, "Why is a sign’s minimum stroke width so important?," 21 11 2011. [Online]. Available: http://blog.businesssigns.net/technology/why-is-a-signs-minimum-stroke-width-so-important.html.

[314] J. Xiao, T. Fang, T. Tan, P. Zhao, E. Ofek and L. Quan, "Image-based Facade Modeling," *CM Transactions on Graphics (TOG) (Vol. 27, No. 5),* p. 61, 2008.

[315] A. Torii, M. Havlena and T. Pajdla, "From google street view to 3d city models," *IEEE 12th international conference on Computer vision workshops (ICCV Workshops), 2009,* pp. 2188-2195, 2009.

[316] B. Micusik and J. Kosecka, "Piecewise planar city 3D modeling from street view panoramic sequences," *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.,* pp. 2906-2912, 2009.

[317] A. Wendel, "Facade Segmentation from Streetside Images," Graz Technical University, Graz, 2012.

[318] M. Skow and H. Tran, "Automatic Exposure Control System for a Digital Camera". US Patent 7173663 - Automatic exposure control system for a digital camera, 2007.

[319] A. Weitz, "Light Meters," 23 9 2011. [Online]. Available: http://www.bhphotovideo.com/indepth/photography/buying-guides/light-meters.

[320] V. Tao, "Mobile mapping technology for road network data acquisition," *Journal of Geospatial Engineering, 2(2),* pp. 1-14, 2000.

[321] D. Nistér, O. Naroditsky and J. Bergen, "Visual Odometry," in *Compute Vision and Pattern Recognition (CVPR '04)*, 2004.

[322] P. Cignoni, C. Montani and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics,* vol. 22, no. 1, pp. 37-54, 1998.

[323] T. Pock, C. Zach and H. Bischof, "Mumford-Shah Meets Stereo: Integration of Weak Depth Hypotheses," *Computer Vision and Pattern Recognition, 2007. CVPR'07,* pp. 1-8, 2007.

[324] Nvidia, "What is GPU Computing?," [Online]. Available: http://www.nvidia.com/object/what-is-gpu-computing.html.

[325] "SICK LMS 291," [Online]. Available: http://www.sickcn.com/media/89775/lms_datasheet.pdf.

[326] S. Thrun, D. Fox, W. Burgard and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial intelligence,* vol. 128, no. 1, pp. 99-141, 2001.

[327] Topcon, "End-to-end Solutions for Mobile Mapping," [Online]. Available: http://www.topconpositioning.com/news-events/news/press-article/end-end-solutions-mobile-mapping.

[328] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering 82 (1). Retrieved 2008-05-03,* p. 35–45, 1960.

[329] H. Malvar, L. He and R. Cutler, "High-Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images," in *International Conference of Acoustic, Speech and Signal Processing*, 2004.

[330] A. Wallace, "Microsoft unveils new Boulder data center," 18 4 2008. [Online]. Available: http://www.dailycamera.com/ci_13140752?IADID=Search-www.dailycamera.com-www.dailycamera.com.

[331] J. Zhao and J. Pjesivac-Grbovic, "MapReduce, The Programming Model and Practice," 19 6 2009. [Online]. Available: http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/papers/mapreduce-sigmetrics09-tutorial.pdf.

[332] J. Dean and S. Ghemawat, *OSDI'04: Sixth Symposium on Operating System Design and Implementation,* 2004.

[333] J. Stibel, "Google's Secret Weapon: MapReduce," 9 12 2008. [Online]. Available: http://blogs.hbr.org/stibel/2008/12/mapreduce-googles-secret-weapo.html.

[334] K. Kraus, I. Harley and S. Kyle, Photogrammetry: Geometry from Images and Laser Scans, Goettingen: Hubert & Co GmbH & Co KG, 2007.

[335] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Microsoft Research, Redmond,WA, 2010.

[336] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96,* p. 3, 2003.

[337] P. Viola and M. Jones, "Robust real-time face detection," *International journal of computer vision,* vol. 57(2), pp. 137-154, 2004.

[338] C. Papageorgiou, M. Oren and T. Poggio, "A general framework for object detection," *Sixth International Conference on Computer Vision,* pp. 555-562, 1998.

[339] J. Hsieh, S. Yu and Y. Chen, "Morphology-based license plate detection from complex scenes," *Proceedings on 16th International Conference on Pattern Recognition, 2002. Proceedings. (Vol. 3),* pp. 176-179, 2002.

[340] F. Porikli and T. Kocak, "Robust license plate detection using covariance descriptor in a neural network framework," *IEEE International Conference on In Video and Signal Based Surveillance, 2006. AVSS'06.,* p. 107, 2006.

[341] K. Deb, H. Chae and K. Jo, "Vehicle license plate detection method based on sliding concentric windows and histogram," *Journal of Computers, 4(8),* pp. 771-777, 2009.

[342] B. Zarit, B. Super and F. Quek, "Comparison of five color models in skin pixel classification," *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings,* pp. 58-63, 1999.

[343] V. Vezhnevets, V. Sazonov and A. Andreeva, "A survey on pixel-based skin color detection techniques," *Proc. Graphicon (Vol. 3),* pp. 85-92, 2003.

[344] J. Kovac, P. Peer and F. Solina, "Human skin color clustering for face detection," *The IEEE Region 8 EUROCON 2003 Computer as a Tool (2003)* , pp. 144-148, 2003.

[345] R. Haralick, S. Sternberg and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence (4),* pp. 532-550, 1987.

[346] A. Bosseau, "Mathematical Morphology - a non exhaustive overview," 21 5 2007. [Online]. Available: http://maverick.inria.fr/Membres/Adrien.Bousseau/morphology/morphomath.pdf.

[347] R. Pajarola, Y. Meng and M. Sainz, "Fast depth-image meshing and warping," *Technical Report UCI-ECE-02-02, Information & Computer Science, University of California Irvine,* 2002.

[348] "PointGrey Ladybug 5," [Online]. Available: http://ww2.ptgrey.com/spherical-vision.

[349] M. McGee, "Google Street View Reaches All 7 Continents; Yes, Even Antarctica," 30 9 2010. [Online]. Available: http://searchengineland.com/google-street-view-antarctica-all-7-continents-51953.

[350] A. Haro, "User-guided Pedestrian and Object Removal," *The Workshop on User-Centred Computer Vision (UCCV),* 2013.

[351] A. Flores and S. Belongie, "Removing pedestrians from google street view images," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp. 53-58, 2010.

[352] A. Nodari, M. Vanetti and I. Gallo, "Digital Privacy: Replacing Pedestrians from Google Street View Images," *21st International Conference on Pattern Recognition (ICPR), 2012,* pp. 2889-2893, 2012.

[353] K. Sugihara, "Some location problems for robot navigation using a single camera," *Computer Vision, Graphics, and Image Processing, 42(1),* pp. 112-129, 1988.

[354] E. Krotkov, "Mobile robot localization using a single image," *IEEE International Conference on Robotics and Automation, 1989. Proceedings,* pp. 978-983, 1989.

[355] U. Nehmzow, T. Smithers and J. Hallam, "Location recognition in a mobile robot using self-organising feature maps," University of Edinburgh, Department of Artificial Intelligence, Edinburgh, 1991.

[356] S. Se, D. Lowe and J. Little, "Global localization using distinctive visual features," *International Conference on Intelligent Robots and Systems (Vol. 1),* pp. 226-231, 2002.

[357] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.,* pp. 1403-1410, 2003.

[358] B. Williams, G. Klein and I. Reid, "Real-Time SLAM Relocalisation," *International Conference on Computer Vision (ICCV),* 2007.

[359] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007.*, 2007.

[360] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco and H. Neven, "Tour the world: building a web-scale landmark recognition engine," *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR,* pp. 1085-1092, 2009.

[361] W. Zhang and J. Kosecka, "Localization based on building recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops.*, 2005.

[362] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.

[363] G. Schindler, M. Brown and R. Szeliski, "City-scale location recognition," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR '07)*, 2007.

[364] D. Lowe, "Distinctive image features from scale-invariant key points," *International Journal of Computer Vision, 60(2), 2004,* p. 91–110, 2004.

[365] A. Irschara, C. Zach, J. M. Frahm and H. Bischof, "From Structure-from-Motion Point Clouds to Fast location Recognition," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR '09)*, 2009.

[366] L. Yunpeng, N. Snavely and D. Huttenlocher, "Location Recognition using Prioritized Feature Matching," in *European Conference on Computer Vision (ECCV)*, 2010.

[367] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *JMLR,* 2001.

[368] B. Johansson and R. Cipolla, "A system for automatic pose-estimation from a single image in a city scene," in *International Conference on Signal Processing, Pattern Recognition and Applications, 2002*, 2002.

[369] D. Robertson and R. Cipolla, "An image-based system for urban navigation," in *Proceedings of British Mashine Vision Conference (BMVC) 2004*, 2004.

[370] A. Le Bris and N. Paparoditis, "Matching Terrestiral Images Captured by a Nomad System to Images of a Reference Database for Pose Estimation Purpose," in *ISPRS 2010*, 2010.

[371] F. Schaffalitzky and A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing,* vol. 18(9), pp. 647-658, 2000.

[372] J. Košecká and W. Zhang, "Video compass," in *Computer Vision—ECCV 2002*, 2002.

[373] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond and G. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proceedings of 7th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'08), Sept. 15–18 2008.* , 2008.

[374] E. Eade and T. Drummond, "Unified loop closing and recovery for real time monocular SLAM," in *British machine vision conference*, 2008.

[375] A. Baumberg, "Reliable feature matching across widely separated views," in *Proceedings of Computer Vision and Pattern Recognition (CVPR) 2000*, 2000.

[376] M. Brown, R. Szeliski and S. Winder, "Multi-image matching using multi-scale oriented patches," in *Proceedings of Conference of Computer Vision and Pattern Recognition (CVPR) 2005, Vol. 2*, 2005.

[377] D. Steedly, C. Pal and R. Szeliski, "Efficiently Registering Video into Panoramic Mosaics," in *Proceedings of 10th IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005.

[378] R. Castle, G. Klein and D. Murray, "Video-rate Localization in Multiple Maps for Wearable Augmented Reality," in *Proceedings of 12th IEEE Symposium on Wearable Computers (ISWC) Sept 2008*, 2008.

[379] T. Lindeberg, "Scale-space theory in computer vision," Kluwer Academic Print on Demand, 1993.

[380] C. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference, pages 1988*, 1988.

[381] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision, vol 30, number 2, 1998,* 1998.

[382] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of International Conference on Computer Vision (ICCV) 2001*, 2001.

[383] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference, Vol. 1,September 2002*, 2002.

[384] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proceedins of European Conference on Computer Vision 2002*, 2002.

[385] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *International Journal of Computer Vision (IJCV) 1(59), 2004,* pp. 61-85, 2004.

[386] C. Wu, B. Clipp, X. Li, J. M. Frahm and M. Pollefeys, "Model Matching with Viewpoint Invariant Patches (VIPs)," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, 2008.

[387] A. J. Chavez, A FAST interest point detection algorithm, Master of Science Thesis, 2008.

[388] F. Fraundorfer and H. Bischof, "Evaluation of local detectors on non-planar scenes," in *Proceedings of 28th workshop of the Austrian Association for Pattern Recognition*, 2004.

[389] A. Gil, O. M. Mozos, M. Ballesta and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual SLAM," *Machine Vision and Applications, Springer-Verlag, March 2009,* 2009.

[390] D. Lowe, "Object recognition from local scale invariant features," in *Proceedings of the International Conference on Computer Vision (ICCV '99)*, Corfu, 1999.

[391] H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded up robust features," in *Proceedings of European Conference on Computer Vision (ECCV '06)*, 2006.

[392] S. Winder and M. Brown, "Learning local image descriptors," in *Proeedings of Conference on Computer Vision and Pattern Recognition (CVPR '07)*, 2007.

[393] S. Winder, G. Hua and M. Brown, "Picking the Best Daisy," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR '09)*, 2009.

[394] M. Brown, G. Hua and S. Winder, "Discriminant Learning of Local Image Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence. February 2010,* 2010.

[395] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) 2003 Vol. 2*, 2003.

[396] S. Lilja, Matching Image Pairs, Master of Science Thesis, Stockholm, Sweden, 2008.

[397] G. Reitmayr and T. Drummond, "Going out: robust model-based tracking for outdoor augmented reality," in *Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR '06)*, 2006.

[398] N. Jacobs, S. Satkin, N. Roman, R. Speyer and R. Pless, "Geolocating static cameras," in *Proceedings of IEEE International Conference on Computer Vision (ICCV), October 2007* , 2007.

[399] "ARToolkit," [Online]. Available: http://www.hitl.washington.edu/artoolkit/.

[400] D. Wagner, T. Langlotz and D. Schmalstieg, "Robust and unobtrusive marker tracking on mobile phones," in *Proc. 7th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'08), Sept. 15–18 2008*, 2008.

[401] C. Mlgaard, "Digital camera with integrated accelerometers". US Patent 6747690, 2004.

[402] TED, "Blaise Aguera y Arcas demos augmented-reality maps," TED, 2 2010. [Online]. Available: http://www.ted.com/talks/blaise_aguera.html. [Accessed 18 6 2013].

[403] I. Witten, A. Moffat and T. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, San Francisco: Morgan Kaufmann Publishers, 1999.

[404] T. Young and J. Van Vliet, "Recursive implementation of the Gaussian filter," *Signal Processing, Volume: 44, Issue: 2,* pp. 139-151, 1995.

[405] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005. (Vol. 1),* pp. 604-610, 2005.

[406] I. Jolliffe, Principal Component Analysis, New York: Sringer Verlag, 2002.

[407] T. Jaakkola, M. Diekhans and D. Haussler, "Using the Fisher kernel method to detect remote protein homologies," *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (Vol. 149),* p. 158, 1999.

[408] F. Perronnin, Y. Liu, J. Sánchez and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 3384-3391, 2010.

[409] H. Luhm, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of research and development, 1(4),* pp. 309-317, 1957.

[410] M. L. A. Lourakis, S. V. Tzurbakis, A. A. Argyros and S. C. Orphanoudakis, "Using Geometric Constraints for Matching Disparate Stereo Views 3D Scenes Containing Planes," in *Proc. of the International Conf. on Pat. Recogn. (ICPR'00), Vol. 1, Barcelona, Spain, Sep. 3-8, 2000*, 2000.

[411] J. Dennis and R. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations (Vol. 16), Society for Industrial and Applied Mathematics, 1987.

[412] E. Eade, "Gauss-Newton / Levenberg-Marquardt Optimization," 20 3 2013. [Online]. Available: http://ethaneade.com/optimization.pdf. [Accessed 18 6 2013].

[413] C. Floudas and P. Pardalos, Encyclopedia of optimization (Vol. 1), Dortrecht, NL: Kluwer Academic Publishers, 2008.

[414] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British machine vision conference* , 2002.

[415] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, 2005.

[416] Microsoft, "Windows Azure Pricing Calculator," [Online]. Available: http://www.windowsazure.com/en-us/pricing/calculator/. [Accessed 20 6 2013].

[417] Amazon, "Amazon Mechanical Turk," [Online]. Available: http://aws.amazon.com/mturk/. [Accessed 20 6 2013].

[418] M. Elinor, "Amazon's Mechanical Turk lets you make $$$, sort of," 21 9 2007. [Online]. Available: http://news.cnet.com/8301-10784_3-9782813-7.html. [Accessed 20 6 2013].

[419] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet physics doklady,* vol. 10, p. 707, 1966.

[420] "What Is a 3G Modem?," [Online]. Available: http://www.wisegeek.com/what-is-a-3g-modem.htm. [Accessed 20 6 2013].

[421] G. Wallace, "The JPEG still picture compression standard," *Communications of the ACM,* vol. 34, no. 4, pp. 30-44, 1991.

[422] J. A. Canny, "Computational Approach To Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence, 8 / 1986,* p. 679–714, 1986.

[423] P. Felzenszwalb and D. Huttenlocher, "Distance transforms," *Tech. Rep., Cornell Computing,* 2004.

[424] J. Walker, Fast Fourier Transform, Second Edition, CRC Press, 1996.

[425] A. Wendel, A. Irschara and H. Bischof, "Automatic alignment of 3D reconstructions using a digital surface model," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp. 29-36, 2011.

[426] E. Ofek, D. Hou, M. Kroepfl, B. Aguera y Arcas, S. Fynn, R. Molinari and T. Ernst, "Spatially registering user photographs". US Patent 8295589, 13 8 2012.

[427] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *8th IEEE International Symposium on Mixed and Augmented Reality, 2009. ISMAR 2009.*, 2009.

[428] R. Szeliski, "Image Based Rendering," in *Computer Vision: Algorithms and Applications*, London, Springer, 2011, pp. 543-574.

[429] M. Hachman, "New Microsoft Maps Combine Photos, Directions," 28 2 2006. [Online]. Available: http://www.extremetech.com/extreme/77200-updated-new-microsoft-maps-combine-photos-directionsN.