

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

DUŠAN OMERČEVIĆ

**HIPERPOVEZOVANJE REALNOSTI S POMOČJO  
FOTOTELEFONA**

MAGISTRSKO DELO

mentor: prof. dr. Aleš Leonardis

LJUBLJANA, 2010

Št.: 134-MAG-RI/2010

Datum: 22. 09. 2010



**Dušan OMERČEVIĆ**, univ. dipl. inž. rač. in inf.

## Ljubljana

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Hiperpovezovanje realnosti s pomočjo fototelefona**

### **Hyperlinking Reality Using a Camera Phone**

Tematika naloge:

Leta 2008 je bilo po celem svetu v uporabi več kot tri milijarde prenosnih telefonov. Čeprav moderni prenosni telefoni omogočajo zahtevno procesiranje in hitre podatkovne komunikacije, se še vedno uporabljajo predvsem za zvočno komunikacijo med ljudmi. Ena glavnih ovir za bolj razširjeno uporabo prenosnih telefonov za dostop do Interneta in drugih podatkovnih omrežij, so neustrezni uporabniški vmesniki prenosnih telefonov, saj tradicionalne vhodne enote, kot so miška in tipkovnica, niso primerne za prenosne naprave. Ker ima večina današnjih prenosnih telefonov vgrajen fotoaparatus, bi lahko uporabnik, namesto tipkanja ključnih besed na majhni in nepripravni tipkovnici telefona, preprosto zajel sliko okolice, na kateri bi se avtomatično dodale hiperpovezave do zanimivih predmetov in tako na enostaven način dostopal do željenih informacij. Obstoječe metode računalniškega vida še niso dovolj robustne za delovanje v okviru omejitev prenosnih telefonov, zato so novejša raziskava usmerjene v to, kako jih prilagoditi za delovanje v različnih okoljih in pogojih.

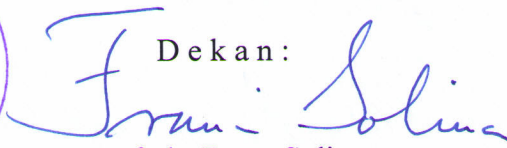
V okviru magistrske naloge proučite uporabo računalniškega vida za izboljšanje uporabniških vmesnikov prenosnih telefonov. Pri tem se osredotočite na algoritme računalniškega vida, ki temeljijo na lokalnih invariantnih značilnicah. Izpostavite probleme obstoječih načinov uporabe lokalnih invariantnih značilnic, ki sledijo iz posebnosti zgradbe in uporabe prenosnih naprav, ter raziščite možnosti nadaljnjih izboljšav z novimi pristopi. V okviru magistrske naloge predlagane metode implementirajte ter jih z ustreznimi eksperimenti ovrednotite glede na obstoječe metode tega področja.

Mentor:

  
prof. dr. Aleš Leonardis



Dekan:

  
prof. dr. Franc Solina





UNIVERSITY OF LJUBLJANA  
FACULTY OF COMPUTER AND INFORMATION SCIENCE

DUŠAN OMERČEVIĆ

**HYPERLINKING REALITY USING A CAMERA  
PHONE**

MASTER THESIS PROPOSAL

Supervisor: Aleš Leonardis, Ph.D.

LJUBLJANA, 2010





# Povzetek

Koncept uporabniškega vmesnika “Hiperpovezovanje realnosti s pomočjo fototelefona”, ki ga predstavljamo v tej nalogi, poskuša rešiti enega izmed ključnih izzivov s katerimi se soočajo uporabniški vmesniki namenjeni prenosnim napravam, to je, izbor in prikaz dejanj, ki ustrezajo trenutnemu uporabnikovemu kontekstu. V predstavljenemu konceptu uporabnik namesto tipkanja ključnih besed na majhni in nepripravni tipkovnici telefona preprosto zajame sliko okolice, nakar se predmetom na sliki avtomatično dodajo hiperpovezave do zanimivih informacij. Naša metoda najprej poišče referenčne panoramske slike, ki prikazuje isti prizor kot uporabnikova slika, in nato preslika na uporabnikovo sliko informacije, ki so bile predhodno označene na referenčnih panoramskih slikah. Dodajanje hiperpovezav do informacij predmetom na uporabnikovi sliki in prikaz tako dopolnjene slike na (večkratni) dotik občutljivem zaslonu fototelefona omogoča uporabniku preprost dostop do željenih informacij. Uporabniku lahko dodatno ponudimo še informacijo o njegovem položaju in usmeritvi, kar lahko predstavlja dopolnilo vgrajenemu globalnemu položajnemu sistemu.

Koncept uporabniškega vmesnika, ki je predstavljen v tej nalogi, sta omogočili nova metoda iskanja ujemanja med visoko-dimenzionalnimi značilnicami, ki temelji na konceptu smiselnih najbližjih sosedov, in nova metoda približnega iskanja najbližjih sosedov, ki je desetkrat hitrejša od izčrpnega iskanja celo v visoko-dimenzionalnih prostorih, pri čemer je približek blizu točnemu iskanju najbližjega sosedu. Naša nova metoda iskanja ujemanja med visoko-dimenzionalnimi značilnicami izboljša uspešnost metod za iskanje ujemanja med slikami na podlagi lokalnih invariantnih značilnic, medtem ko nova metoda približnega iskanja najbližjih sosedov približuje predstavljen sistem interaktivnosti v realnem času.

Koncept uporabniškega vmesnika za prenosne naprave “Hiperpovezovanje realnosti s pomočjo fototelefona” potrebuje za delovanje nabor predhodno zajetih referenčnih panoramskih slik na katerih so predmeti označeni in povezani z informacijami. Nabor slik iz Gradca obsega 107 referenčnih panoramskih slik, ki so bile posnete iz natančno izmerjenih položajev, medtem ko je bila usmeritev panoramskih slik izračunana naknadno s pomočjo tehnik računalniškega vida, čemur je sledila ročno preverjanje. Na vsaki referenčni panoramski sliki smo s pomočjo računalniškega programa za dodajanje hiperpovezav označili nekaj deset stavb, napisov, spomenikov in drugih uporabniku zanimivih predmetov.

**Ključne besede:** iskanje ujemanja med slikami na podlagi lokalnih invariantnih značilnic, ujemanje stereo slik, dopolnjena resničnost, določanje položaja na podlagi slike, smiselni najbližji sosedje, iskanje ujemanja med visoko-dimenzionalnimi značilnicami, približno iskanje najbližjih sosedov, redko kodiranje s prekopolno množico baz



# Abstract

A user interface concept for camera phones, called “Hyperlinking Reality via Camera Phones”, that we present in this thesis, provides a solution to one of the main challenges facing mobile user interfaces, that is, the problem of selection and visualization of actions that are relevant to the user in his current context. Instead of typing keywords on a small and inconvenient keypad of a mobile device, a user of our system just snaps a photo of his surroundings and objects in the image become hyperlinks to information. Our method commences by matching a query image to reference panoramas depicting the same scene that were collected and annotated with information beforehand. Once the query image is related to the reference panoramas, we transfer the relevant information from the reference panoramas to the query image. By visualizing the information on the query image and displaying it on the camera phone’s (multi-)touch screen, the query image augmented with hyperlinks allows the user intuitive access to information. In addition, we provide the user with information about his position and orientation, thus augmenting the built-in GPS.

The user interface concept presented in this thesis is enabled by a novel high-dimensional feature matching method based on the concept of meaningful nearest neighbors and a novel approximate nearest neighbors search method that provides a ten-fold speed-up over an exhaustive search even for high dimensional spaces while retaining excellent approximation to an exact nearest neighbors search. Our novel high-dimensional feature matching method improves effectiveness of image matching methods which are based on local invariant features, while the speed-up provided by the novel approximate nearest neighbors search method, brings our system closer to interactivity in real-time

The “Hyperlinking Reality via Camera phones” mobile user interface concept requires a data set of reference panoramas that are collected and annotated with information beforehand. The Graz Urban Image data Set consists of 107 reference panoramas shot from accurately measured positions, while the camera orientations were acquired in post-processing stage using computer vision techniques followed by manual verification. On each reference panorama a few dozens of buildings, logos, banners, monuments, and other objects of interest to the user were annotated using the hyperlinks annotation tool.

**Keywords:** Image matching using local invariant feature, Wide baseline stereo matching, Augmented reality, Image based localization, Meaningful Nearest Neighbors, High-dimensional feature matching, Approximate Nearest Neighbors Search, Sparse coding with an overcomplete basis set





Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrske naloge je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.





# Acknowledgements

First, I would like to acknowledge the help of Ondrej Drbohlav. Besides teaching me many secrets of writing great scientific papers, Ondrej wrote most of the Section 3.2.1 thus making our ideas about the concept of meaningful nearest neighbors much more comprehensible.

Then, I would like to thank Roland Perko for all his help and fruitful discussions. His ideas, support and encouragement are spread throughout this thesis.

I am very grateful also to all other members, past and present, of Visual Cognitive Systems (VICOS) lab Danijel Skočaj, Matjaž Jogan, Luka Fürst, Aleš Štimec, Barry Ridge, Luka Čehovin, Marko Mahnič, Jurij Šorli, Matej Kristan, and Alen Vrečko.

The research presented in this thesis has been supported by EU FP6-511051-2 project MOBVIS. I would like to thank Lucas Paletta for exemplary management of the project and all other project members, Alexander Almer, Gerald Fritz, Christin Seifert, Patrick Luley, Katrin Amlacher, Aleš Leonardis, Matjaž Jogan, Roland Perko, Marko Mahnič, Jurij Šorli, Alireza Tavakoli Targi, Kristy Sim, Eric Hayman, Mårten Björkman, Bernt Schiele, Ulrich Steinhoff, Tâm Huynh, Kristof Van Laerhoven, and Linde Vande Velde, for a great collaboration. Most especially I would like to thank Jan-Olof Eklundh for all his support and encouragement, and for teaching us the Skitgubbe card game.

Acquisition of reference data images of Ljubljana, Darmstadt and Graz was done in collaboration with Roland Perko and Ulrich Steinhoff. The acquisition could not be done without the help of Bojan Stopar, Dejan Grigillo, Albin Mencin, Asobi d.o.o., Luka Fürst, Jurij Šorli, Jason Catchpole, Konstantinos Konstantinidis, Steven Reynold, Tâm Huynh and Katrin Amlacher.

Mostly, I am thankful to prof. dr. Aleš Leonardis for guidance over the course of my Master's studies. It has been an honor to be his student and I sincerely hope he will continue to guide me through the process of my studies and help me prepare myself for entering into academia.

Magistrsko delo je posvečeno ata Francu. Njegova skromnost mi je vzor.

Poljubčkov sto mojim najljubšim puncam Flori, Juno in Tini :)



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contributions . . . . .	2
1.2	Outline of the Thesis . . . . .	4
<b>2</b>	<b>Related Work</b>	<b>5</b>
2.1	Mobile applications of computer vision . . . . .	5
2.2	Local invariant features . . . . .	5
2.3	Feature matching . . . . .	6
2.4	Approximate nearest neighbors search . . . . .	7
<b>3</b>	<b>Method</b>	<b>9</b>
3.1	Local invariant features . . . . .	10
3.1.1	Local invariant features detector . . . . .	10
3.1.2	Local invariant features descriptor . . . . .	12
3.2	Feature matching . . . . .	13
3.2.1	The concept of meaningful nearest neighbors . . . . .	16
3.2.2	Approximate nearest neighbors search . . . . .	20
3.3	Estimation of geometric relations . . . . .	28
3.3.1	Epipolar geometry estimation . . . . .	28
3.3.2	In search of true correspondences . . . . .	29
3.3.3	Deriving the prior probability of a true correspondence . . . . .	31
3.3.4	Sampling the space of possible hypothesis . . . . .	31
3.3.5	Structure estimation . . . . .	32
3.3.6	Detection of pure rotation . . . . .	32
3.4	Transfer of hyperlinks . . . . .	33
3.5	Visualization of results . . . . .	34
<b>4</b>	<b>Reference Data Set</b>	<b>37</b>
4.1	Acquiring the data set . . . . .	37
4.2	Hyperlinks annotation tool . . . . .	39
<b>5</b>	<b>Results</b>	<b>41</b>
5.1	Performance evaluation of the proposed high-dimensional feature matching method . . . . .	41
5.1.1	Recognition benchmark images . . . . .	42
5.1.2	Ljubljana urban image data set . . . . .	42



## CONTENTS

---

5.2	Performance evaluation of the proposed approximate nearest neighbors search method . . . . .	44
5.2.1	Retrieval rate . . . . .	44
5.2.2	Quality of approximation . . . . .	44
5.2.3	Speed-up . . . . .	45
5.3	Evaluation of the novel user interface concept . . . . .	47
5.3.1	System setup . . . . .	50
<b>6</b>	<b>Summary and Conclusions</b>	<b>51</b>
<b>A</b>	<b>Povzetek magistrske naloge v slovenskem jeziku</b>	<b>55</b>
A.1	Uvod . . . . .	55
A.2	Pregled področja . . . . .	56
A.3	Metoda . . . . .	57
A.4	Nabor referenčnih panoramskih slik . . . . .	58
A.5	Eksperimentalni rezultati . . . . .	60
A.6	Zaključek . . . . .	62

# List of Figures

1.1	Nokia Research’s MARA project . . . . .	2
1.2	System overview . . . . .	3
1.3	Example of the concept of meaningful nearest neighbors . . . . .	4
3.1	MSER features described by SIFT descriptor. . . . .	10
3.2	High-dimensional feature matching. . . . .	11
3.3	Estimation of geometric relations. . . . .	12
3.4	Transfer of hyperlinks. . . . .	13
3.5	Example of detected MSER features . . . . .	14
3.6	Example of non-repeatable detection of features . . . . .	15
3.7	Distributions of dot product values between a randomly selected point and all other points on a $D$ -dimensional hypersphere . . . . .	17
3.8	Distribution of dot product values between five randomly selected query features and all other (seven million) features . . . . .	18
3.9	Finding meaningful nearest neighbors . . . . .	19
3.10	Example of query features that have meaningful nearest neighbors among the features detected in one of the reference panoramas . . . . .	21
3.11	Low-dimensional illustration of sparse coding with an overcomplete basis set . . . . .	22
3.12	Explanation how sparsity can speed up nearest neighbors search . . . . .	23
3.13	A brief explanation of epipolar geometry constraint. . . . .	29
3.14	Calculated position visualized on a map where the query image was shot . . . . .	35
4.1	Positions where the reference panoramas were shot and an example of images that together make a panorama . . . . .	38
4.2	Example of query images . . . . .	38
4.3	Example of stitched reference panoramas with annotated hyperlinks . . . . .	39
4.4	Screen shot of the hyperlink annotation tool . . . . .	40
5.1	Performance of the proposed high-dimensional feature matching method compared to the vocabulary tree based matching, 1-NN matching, and augmented k-NN matching . . . . .	42
5.2	Performance of the proposed high-dimensional feature matching method compared to 1-NN matching, and augmented k-NN matching on the Ljubljana urban image data set . . . . .	43

## LIST OF FIGURES

---

5.3	Three sample query images of the Ljubljana urban image data set and the top five images retrieved by our approach . . . . .	45
5.4	Comparison of recognition performance when using the approximate and the exact nearest neighbors search . . . . .	46
5.5	Retrieval rate of the proposed approximate nearest neighbors search method	47
5.6	Query images with annotated hyperlinks . . . . .	49
A.1	Pregledna shema metode in poglavitne tehnike računalniškega vida, ki smo jih uporabili. . . . .	59
A.2	Uporabniške slike z dodanimi hiperpovezavami. . . . .	61

# List of Tables

5.1	Cumulative number of query images for which (at least) the specified number of groundtruth reference images were retrieved among five top-ranked reference images . . . . .	44
5.2	Accuracy of image based localization . . . . .	48
A.1	Točnost določanja položaja na podlagi slike. . . . .	62



# Chapter 1

## Introduction

Already in the year 2008 there were more than three billion mobile phones in use throughout the world [31]. Even though the modern mobile phones possess substantial processing and data communication possibilities, they are still predominantly used only for voice communication between people. One of the major obstacles for using mobile phones to also access information available on the Internet and other data networks are the inadequate user interfaces on mobile phones. Besides small displays, the major problem of mobile user interfaces are input devices. Traditional input devices of desktop computers such as keyboards and mice are not suitable for mobile devices due to their space requirements, while some other techniques such as voice control are not commonly accepted by users [33].

Today, the most popular user interface concept for accessing information on mobile devices is navigation among the limited number of actions that the user can select using a keypad, a joystick, or more recently, a (multi-)touch screen [33]. The main challenge of this concept is deciding on what actions to present to the user and how to visualize them. Due to the abundance of information available, different user requirements, and limited information about user's context, selection of actions that are relevant to the user in his current context is a difficult and challenging problem that is a critical success factor of mobile user interfaces.

An interesting alternative to traditional mobile user interfaces was investigated in Nokia Research's MARA (Mobile Augmented Reality Applications) project [29]. In this project a mobile phone equipped with accelerometers, compass and a GPS was used as a window to an augmented reality environment in which users could access information by pointing the phone's camera in the direction of an interesting object (see Figure 1.1 for an example). Additional information about the object was accessible as a hyperlink overlaid over the video stream taken by the phone's camera and hence the concept got a name "hyperlinking reality via phones" [20]. If we are able to attribute actions to objects that surround the user, then, suddenly, all the objects in the environment become action triggers that a user can activate by a physical action of pointing the camera phone towards the object. Moreover, a photo of the environment on the camera phone's (multi-)touch screen becomes a natural interaction device allowing intuitive access to information. Therefore, the concept of "hyperlinking reality via phones" solves one of the main problems of the "action navigation" mobile user interface concept, that is, the selection and visualization of relevant

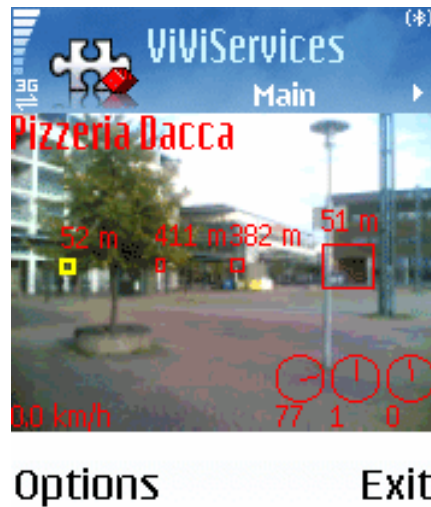


Figure 1.1: Nokia Research’s MARA project resulted in a prototype camera phone, equipped with sensors and software, that could superimpose virtual information and hyperlinks onto a real-world scene. Credit: Nokia Research Center

actions. With the proliferation of mobile devices equipped with a GPS and a compass, the MARA project’s approach to mobile augmented reality has become mainstream and widely accepted by the users [11]. The recent examples of such applications are Layar [32] and Wikitude [71].

The major drawback of the MARA project’s approach is its dependence on the accurate information about absolute location and orientation of the camera phone. For example, the median accuracy of location information provided by the GPS is only 10 meters [56], which is insufficient for accurate superimposition of hyperlinks in a video stream, resulting in poor user experience. In this thesis we propose an alternative to the MARA project’s approach. Instead of relying on the knowledge about absolute location and orientation of the camera phone, we employ computer vision techniques to identify object(s) that a user is pointing to, thus implementing the same user interface concept as the MARA project but using a different technology.

## 1.1 Contributions

The main contribution of this thesis is a user interface concept for camera phones, called “Hyperlinking Reality via Camera Phones”, that provides a solution to one of the main challenges facing mobile user interfaces, that is, the problem of selection and visualization of actions that are relevant to the user in his current context. Instead of typing keywords on a small and inconvenient keypad of a mobile device, a user of our system just snaps a photo of his surroundings and objects in the image become hyperlinks to information (see Figure 1.2 for an example).

The second contribution of this thesis is a novel high-dimensional feature matching

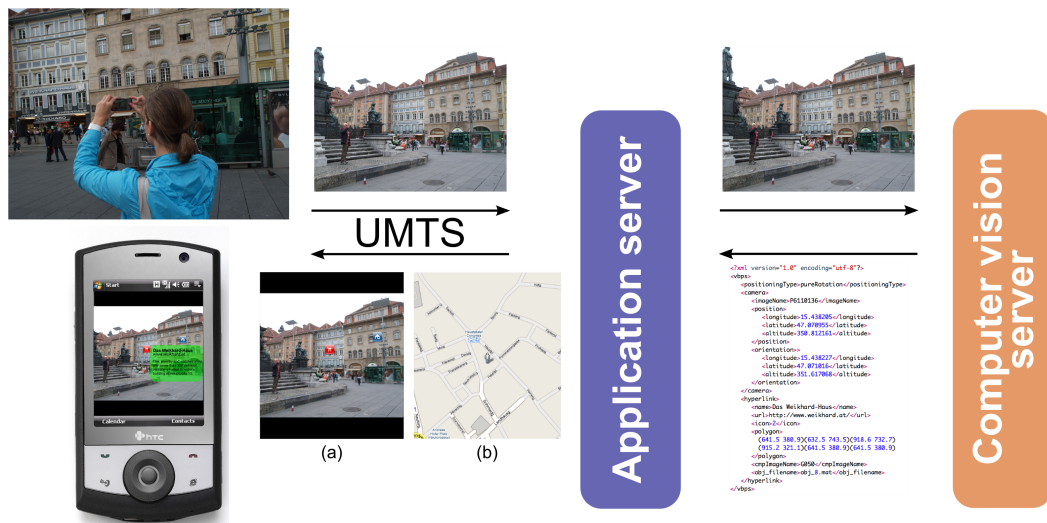


Figure 1.2: To access information about objects in his surroundings, a user just snaps a photo (top left image) and objects on the image become hyperlinks to information (bottom left image) that a user can access by simply tapping an icon. The user's photo is first sent to an application server over a UMTS network and then forwarded to a server which, by employing computer vision techniques, identifies relevant hyperlinks and calculates the user's position. The result is sent back to the application server in XML form, where the hyperlinks are depicted as icons on the photo (image a) while the calculated position is visualized on a map (indicated by icon of a man in image b), and finally displayed on the camera phone. The camera phone used in our experiments is the HTC Touch Cruise. See Chapter 5 for more information about the system setup.

method based on the concept of meaningful nearest neighbors. A nearest neighbor is considered meaningful if it is sufficiently close to a query feature such that it is an outlier to a background feature distribution (see Figure 1.3 for an example). Our novel high-dimensional feature matching method improves performance of image matching methods which are based on local invariant features, thus enabling the novel user interface concept for camera phones that is presented in this thesis.

The third contribution of this thesis is a novel approximate nearest neighbors search method that provides a ten-fold speed-up over an exhaustive search even for high dimensional spaces while retaining excellent approximation to an exact nearest neighbors search. The speed-up provided by this method, brings our system closer to interactivity in real-time, which is one of the three properties of a proper augmented reality system (see Chapter 6).

Our work has been published at international conferences [48, 56, 25, 49] and in a computer science journal [50].

The video presentation of "Hyperlinking Reality via Camera Phones" concept is available at <http://vicos.fri.uni-lj.si/HypR/>



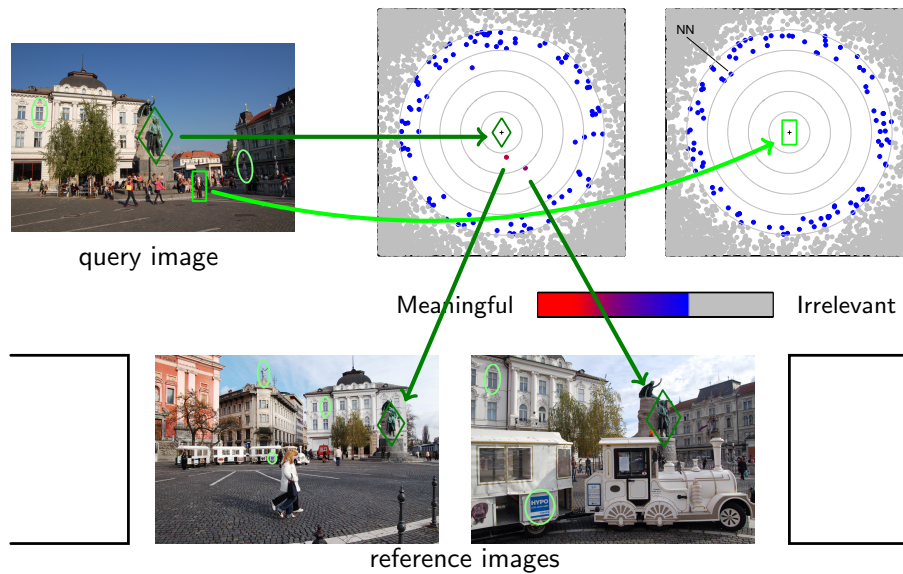


Figure 1.3: An example of the concept of meaningful nearest neighbors. Given a query image, we would like to retrieve reference images depicting the same scene. Therefore, local invariant features (marked as rhomb, ellipse and rectangle) are first detected in the query image and then the visual content of detected features is described in the form of high-dimensional vectors. For each feature detected in the query image we search for the most similar features in the reference images. Only features that are substantially more similar to the query feature than the rest of features detected in the reference images are considered meaningful and are used to vote for the respective reference image.

## 1.2 Outline of the Thesis

In the next chapter we give a short description of related work on mobile applications of computer vision and we refer to the existing computer vision techniques based on local invariant features. In Chapter 3 we provide a detailed description of our system. The “Hyperlinking Reality via Camera phones” mobile user interface concept requires a data set of reference panoramas that are collected and annotated with information beforehand. Therefore, in Chapter 4, we describe the process of reference data set acquisition. As part of the user acceptance study [25], we have evaluated the performance of our system in a real-world scenario. We present the results of the evaluation in Chapter 5. Finally, in Chapter 6, we provide thesis’ summary and conclude with a discussion of benefits that interactivity in real-time would bring to our system.

## Chapter 2

# Related Work

### 2.1 Mobile applications of computer vision

Nowadays, there are very few mobile phones on the market without a built-in camera. It is estimated that the total number of these, so called camera phones in the world, has in 2008 surpassed one billion items [39]. Therefore, it is natural to consider using the camera as an input device for querying information on mobile phones [52, 72, 70, 14, 61]. Among these systems, the most similar to ours is the system of [61, 14]. Contrary to our system, most of the processing in the system of Takacs et al. [61] is done on the mobile phone itself by pushing subset of features continuously from the server to the mobile phone. At current bandwidth of wireless networks, the processing on the mobile phone provides for faster system response time at the expense of quality of results. While our system is not interactive in real-time, the system of Takacs et al. is not registered in 3D and as such is also not a proper augmented reality system (see Chapter 6).

As stated by Henrysson [23, page 2] in his excellent PhD thesis, the tight coupling of camera and CPU gives mobile phones unique input capabilities where real-time computer vision is used to enable new interaction metaphors and link physical and virtual worlds. In his thesis, Henrysson presents the state of the art in camera phone augmented reality and describes several augmented reality applications running on a camera phone, the most famous of them being AR Tennis. As stated before, using non-vision sensors for augmented reality has some serious drawbacks due to the limited accuracy of such sensors. Therefore, Henrysson used marker-based computer vision algorithms for tracking and adapted them for camera phones, thus enabling real-time interaction with the system. Marker-based tracking requires placement of special recognizable codes in the environment for easy detection using camera phones. In outdoor environments, marker-based tracking is not an option due to the difficulties in preparing the environment and wide area use [23].

### 2.2 Local invariant features

In our work, we are not limited by the requirements of augmented reality on real-time interaction, though we would not object to such a feature in our system (see Chapter 6 for a

discussion about benefits of real-time interaction for the user). Instead we have used more powerful, but also more computationally demanding computer vision techniques based on local invariant features that were pioneered by Schmid and Mohr [53], and Lowe [35], and that are used for solving many computer vision problems, including image retrieval [53, 46, 12], object recognition [35, 16], wide baseline matching [3, 38, 67], building panoramas [8], image based localization [74] and video data mining [55]. In these applications, local invariant features are detected independently in each image and then the features of one image are matched against the features of other images by comparing respective feature descriptors. The matched features can subsequently be used to indicate the presence of a particular object, to vote for a particular image, or as tentative correspondences for epipolar geometry estimation.

## 2.3 Feature matching

The elementary methods for matching local invariant features such as threshold-based matching and nearest neighbor(s) matching, treat all features equally, while more sophisticated methods take into account that some of the local invariant features are more distinctive than others. Among the more sophisticated methods, Baumberg's method [3] tries to compensate for variable distinctiveness of local invariant features by identifying the closest and second-closest neighbor of a query feature. The closest neighbor is selected as an *unambiguous* match only if it is much closer than the second-closest neighbor. A similar concept was used also by Lowe [35].

Another approach to matching is inspired by text retrieval methods and uses entropy weighting to explicitly account for variable distinctiveness of local invariant features. Sivic and Zisserman [55] used vector quantization to partition the set of feature descriptors into disjoint subsets which they termed *visual words*. They consider all features associated with a particular visual word as matches to each other. Recently, Nistér and Stewénus [46] designed a hierarchical vector quantization method with strong emphasis on speed of matching. The resulting visual vocabulary tree, containing up to several million leaves, enabled them to partition the set of feature descriptors into a much larger number of subsets than was the case with the method of Sivic and Zisserman. The individual feature descriptors were matched implicitly, by comparing paths of feature descriptors down the vocabulary tree. A hierarchical structure similar to the vocabulary tree was also used in a vocabulary-guided pyramid match method of Grauman and Darrell [19] for computing an approximate partial matching between two *sets* of feature vectors.

None of the traditional methods have met our criteria that a good feature matching method should provide a sufficient number of votes for matching reference images and that it should provide a set of tentative correspondences with sufficient number and percentage of true correspondences. That is why we have developed a novel high-dimensional feature matching method based on the concept of meaningful nearest neighbors that we present in this thesis.

## 2.4 Approximate nearest neighbors search

A major advantage of the classical approach to feature matching over other approaches is its simplicity and conceptual clarity. By performing explicit nearest neighbors search, the problem of feature matching reduces to a problem of selection of relevant distance metric and a problem of identification of true matches within a small set of nearest neighbors. As noted already by [9] for the case of indexing visual shapes and later by [40] for the case of local invariant features, the low-dimensional feature descriptors (i.e., represented by vectors with up to ten elements) do not perform as well as high-dimensional ones (i.e., represented by vectors with more than 25 elements [44]). Consequently, the most successful feature descriptors, such as SIFT [35], Shape context [6], and PCA-SIFT [30], describe the visual appearance of features with high-dimensional vectors. While there exist several efficient (i.e., logarithmic in number of data points) nearest neighbors search algorithms for the case of low-dimensional data points (see [10, 24] for recent surveys), no algorithms are known that can identify exact nearest neighbors of points in high dimensional spaces that are any more efficient than exhaustive search [35, 7, 26, 13]. Because exhaustive search is too slow for many applications that process high-dimensional data points, several approximate methods were proposed that trade accuracy for speed, i.e. they might miss some of the nearest neighbors while providing substantial speed-up over exhaustive search.

Among the earliest approximate nearest neighbors search methods is the method of [1]. It uses balanced box-decomposition (BBD) tree in order to hierarchically decompose the space of data points into  $D$ -dimensional rectangles whose sides are orthogonal to the coordinate axes. The nearest neighbors search begins in the rectangle that contains the query point and proceeds by enumerating the rectangles in increasing order of distance from the query point. The search can be terminated when the closest point seen so far is not much farther from the query point (as specified by approximation parameter  $\epsilon$ ) as the next rectangle being enumerated. By performing early termination of the search this algorithm provides significant improvements over exhaustive search in dimensions as high as 20.

A similar approach was used also in Best-Bin-First (BBF) algorithm of [5]. The BBF algorithm uses a KD-tree [18] with a modified search ordering that searches bins of the KD-tree in the order of their closest distance from the query point. By cutting off further search after a specific number of the nearest bins have been explored (200 in implementation described in [35]), the BBF algorithm provides a speed-up over exhaustive search by two orders of magnitude for data points of moderate dimensionality (i.e. 10 - 20 dimensions).

An alternative approach to nearest neighbors search is the Projection search of [44]. By projecting data points to a smaller number of dimensions and by limiting the search in each projected dimension to a small range, the Projection search achieves a linear speed-up over exhaustive search.

Another approach to nearest neighbors search is locality-sensitive hashing (LSH) of [27]. A locality-sensitive hashing function has the property that two points hash to the same bucket with much higher probability if they are close, than if they are far apart. The nearest neighbors of the query point are identified by exhaustively evaluating data points that hash to the same bucket as the query point. By combining several independent locality-sensitive hashing functions an arbitrary high accuracy can be achieved but with query time and storage

requirements growing linearly with the number of hashing functions used.

Because none of these existing approaches to approximate nearest neighbors search seemed capable of providing sufficient speed-up in high-dimensional spaces while retaining adequate quality of the approximation, we have developed a novel approach based on sparse coding with an overcomplete basis set that we present in this thesis.

## Chapter 3

# Method

As stated in the Introduction, the goal of our work is to enable a novel user interface for mobile devices. Instead of typing keywords on a small and inconvenient keypad of a mobile device, a user just snaps a photo of his surroundings and objects on the image become hyperlinks to information. We will call the photo snapped by the user a query image. Our method commences by matching the query image to the reference panoramas depicting the same scene that were collected and annotated with information beforehand (see Chapter 4 for a detailed description of the reference panorama dataset). Once the query image is related to the reference panoramas, we transfer the relevant information from the reference panoramas to the query image. By visualizing the information on the query image and displaying it on the camera phone's (multi-)touch screen, the query image augmented with hyperlinks allows the user intuitive access to information.

The approach we have chosen in order to bring the “hyperlinking reality” functionality to camera phones is image matching using local invariant features. In image matching a query image (or a part of it) is matched against the reference images (or their parts) in order to relate the query image with a subset of reference images depicting the same scene or objects. The establishment of relations between the query and the reference images enables transfer of information from the reference images to the query image. For example, if we know the position and camera orientation of the reference image and if we have estimated geometry relating the query and the reference image, we can estimate the position and camera orientation of the query image up to a scale ambiguity [22].

The dominant framework of using local invariant features for image matching is the approach of [53]. Their approach starts with (1.a) a detection and (1.b) a description of a set of local features in an image, followed by (2.) a matching of similar local structures in two images and (3.) rejection of incorrect relations between the images. The main advantage of the approach of Schmid and Mohr and of using local invariant features in general is that it can handle substantial viewpoint and photometric changes and that it tolerates substantial clutter and occlusions [66].

Consistent with the framework of Schmid and Mohr, the first step of our method is detection and description of local invariant features in the query image snapped by the user (see Figure 3.1). In the second step, the detected features are matched against the fea-

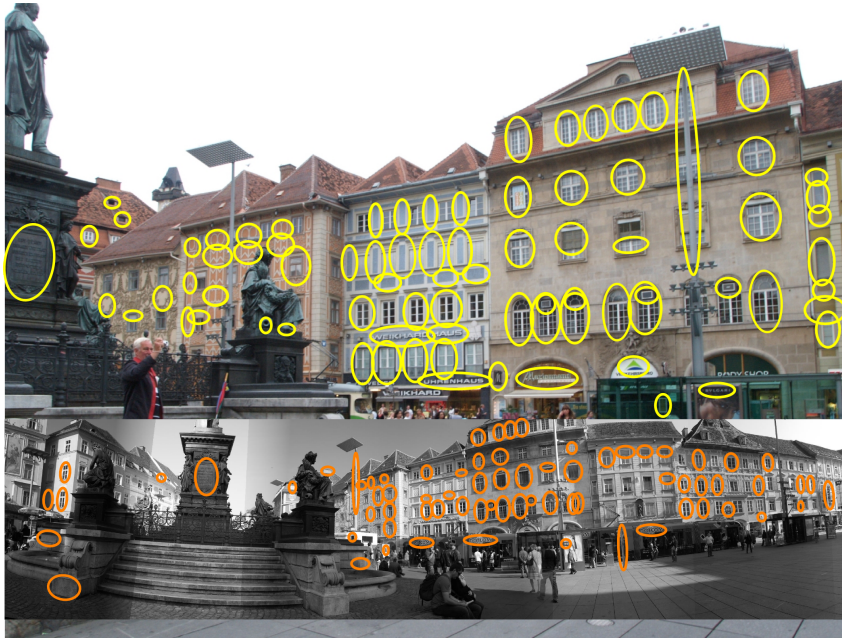


Figure 3.1: The first step of our method: MSER features [38] described by SIFT descriptor [35]. The query image is in color, while the reference panorama is greyscale.

tures detected beforehand in the reference panorama images in order to identify reference panoramas depicting the same scene (see Figure 3.2). In the third step, we try to estimate geometric relations between the query image and the subset of reference panoramas depicting the same scene (see Figure 3.3). In the fourth step, the hyperlinks defined on the reference panoramas by users or administrators are transferred on the query image (see Figure 3.4). In the last step, the hyperlinks are annotated on the query image and displayed on the camera phone’s (multi-)touch screen.

The next subsections give details about our method.

### 3.1 Local invariant features

In our framework we use local invariant features as a starting point for the process of identification of image regions depicting the same scene in the two images. The local invariant features are features of intermediate complexity, which means that they are distinctive enough to determine likely matches in a large database of features but are sufficiently local to be insensitive to clutter and occlusion [34].

#### 3.1.1 Local invariant features detector

We have chosen Maximally Stable Extremal Regions (MSER) [38] as features used in our framework. These features represent connected image regions of similar light intensity with

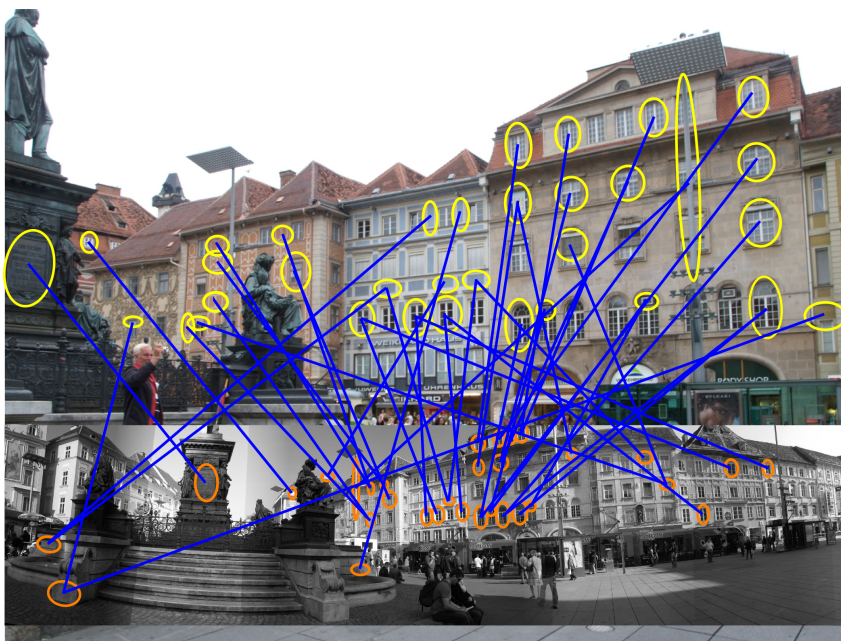


Figure 3.2: The second step of our method: High-dimensional feature matching [48]. The query image is in color, while the reference panorama is greyscale.

well-defined borders. Due to the abundance of such regions in man-made environments (e.g., letters, signs, banners) they are the most suitable type of features for applications targeted at urban settings. MSER features have also performed best in several performance evaluations of local invariant features [41, 42].

In order to describe the MSER feature in a canonical form [41], an ellipse is fitted to the detected maximally stable extremal region. The local invariant features approach assumes that the visual structure is locally planar and small compared to the distance from the camera to the visual structure, and therefore the possible viewpoint changes can be well approximated by an affine transformation of an ellipse. See Figure 3.5 for an example of detected MSER features.

MSERs represent connected image regions of similar light intensity with well-defined borders. If a visual structure is occluded by some object in the foreground that splits the visual structure in two or more parts, then the visual structure is represented not by one but by several features. Image matching based on local invariant features requires one-to-one matching between features and therefore cannot put in correspondence the single feature detected in the first image with multiple features representing the same visual structure in the second image. An example is given in Figure 3.6 where this problem arises due to the appearance of electric wires in the image. Similar effect is caused also by branches and twigs of trees in winter times, and by tree leaves in other seasons [51], fences, and all other types of objects that only partially occlude background objects.





Figure 3.3: The third step of our method: Estimation of geometric relations [22]. The query image is in color, while the reference panorama is greyscale.

### 3.1.2 Local invariant features descriptor

The visual content of each detected MSER feature is characterized by computing a SIFT descriptor [35] for the respective elliptical image region.

We assume that the MSER feature detector has produced a set of features that are (reasonably well) registered to the respective visual structures but the image of the visual structure might have been taken under different illumination or from a different viewpoint. There is a consensus in the computer vision community [40, 66] that the SIFT descriptor [35] is best at providing invariance to illumination and viewpoint changes, and tolerance to slight misregistrations between the feature and the respective visual structure, while being distinctive enough to determine likely matches in a large database of features. The SIFT descriptor is computed in three steps. In the first step, an elliptical image region is normalized to the circular one and then the (circular) image region is rotated in the direction of dominant gradient orientation, providing invariance to viewpoint changes. In the second step, gradient location is quantized in  $4 \times 4$  location grid and the gradient angle is quantized into eight orientations, resulting in 128-dimensional descriptor vector. The trilinear interpolation that is used to distribute the value of each gradient sample into adjacent histogram bins, provides tolerance to slight misregistrations between the feature and the respective visual structure. In the final step, the descriptor vector is normalized in order to provide invariance to illumination changes.

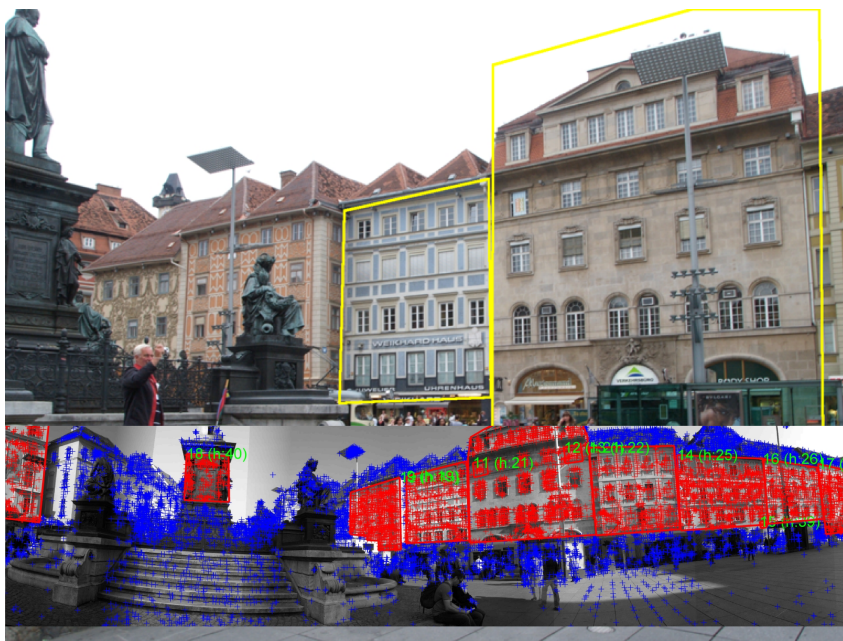


Figure 3.4: The fourth step of our method: Transfer of hyperlinks. The query image is in color, while the reference panorama is greyscale with annotated objects and detected features (shown as plus signs) depicted. The red plus signs indicate features that are included in a description of some object. The red polygon indicates the extent of the object within the reference panorama and the numbers next to the polygon are object and hyperlink identifier, respectively.

## 3.2 Feature matching

The first step of our method has provided us with a large number of features detected in the reference panoramas and much smaller, but still substantial, number of features detected in the query image. The goal of the feature matching is to relate each query feature with none, one, or several features detected in the reference panoramas. Each match translates into a vote for a particular reference panorama, and might become a tentative correspondence between image regions in the query image and a reference panorama, respectively. For the voting to be successful, a sufficient number of votes should go to the matching reference images, while only a smaller number of votes can go to unrelated reference images. Similarly, the set of tentative correspondences should include a sufficient number and percentage of true correspondences for the successful estimation of relation between the query image and the reference panorama using the hypothesize-and-test approach [17].



Figure 3.5: An example of detected MSER features. Top: input image. Bottom: detected MSER features. For the sake of clarity, only every fourth MSER feature is shown.





Figure 3.6: In the top image letter W is represented by a single MSER feature. In the bottom image the letter W is cut in several pieces by electric wires. While these wires have negligible effect on human recognition of the letter W, they have profound effect on automatic image matching. Due to the wires, the letter W is no longer a single connected image region of similar light intensity but is instead represented by five MSER features with each of the five features representing only a portion of the letter. Image matching based on local invariant features requires one-to-one matching between features and it cannot comprehend that the letter W in the bottom image is actually represented by five MSER features.

### 3.2.1 The concept of meaningful nearest neighbors

The elementary methods for matching local invariant features such as threshold-based matching and nearest neighbor(s) matching, treat all features equally, while more sophisticated methods [3, 35] take into account that some of the local invariant features are more distinctive than others. Another approach to matching is inspired by text retrieval methods and uses entropy weighting to explicitly account for variable distinctiveness of local invariant features [55, 46, 19]. None of the traditional methods have met our criteria that a good feature matching method should provide a sufficient number of votes for matching reference images and that it should provide a set of tentative correspondences with sufficient number and percentage of true correspondences. That is why we have developed a novel high-dimensional feature matching method based on the concept of meaningful nearest neighbors that was first presented in [48].

In this section we first demonstrate the effects pertinent to high-dimensional spaces that have significant implications for feature matching. Subsequently we explain the concept of meaningful nearest neighbors, and present the method which we use to implement this concept.

#### Simulated example

Let us consider a set  $\mathcal{F}$  of features which are distributed uniformly on the intersection  $\mathcal{S}_D$  of the positive  $D$ -dimensional hyperquadrant and the surface of a unit  $D$ -dimensional sphere (thus for  $\mathbf{x} \in \mathcal{S}_D$ , there holds that  $\|\mathbf{x}\| = 1$  and all the components of  $\mathbf{x}$  are positive). Given an arbitrary feature point  $\mathbf{f} \in \mathcal{F}$ , we are interested in the distribution of values of dot products of  $\mathbf{f}$  with the rest of the features (throughout this thesis, to capture similarity of a pair of unit vectors  $\mathbf{x} \in \mathcal{S}_D$ ,  $\mathbf{y} \in \mathcal{S}_D$  we use dot products  $\mathbf{x} \cdot \mathbf{y}$ ). This distribution  $p(r)$  is, for a given  $\mathbf{f} \in \mathcal{F}$ , written as

$$p(r) = \frac{dP(\mathbf{f} \cdot \mathbf{g} \leq r)}{dr}, \quad \mathbf{g} \in \mathcal{F} \setminus \mathbf{f}, \quad (3.1)$$

where  $P(\cdot)$  denotes probability. Figure 3.7 shows examples of these distributions for three different dimensionalities.

We can see that for  $D = 3$ , the shapes of the distributions vary greatly for the five different reference points used. For dimensionality  $D = 15$ , the differences are far less pronounced. Importantly though, another effect enters: negligible distribution value in direct vicinity of a query feature, and a rapid growth of the distribution as the distance increases (i.e. dot product falls). For  $D = 128$ , the shape of the distributions for the five different reference points used is virtually the same, while the initial growth of the distribution is indeed even faster compared to the previous case. In the following paragraph we show that effects observed here for higher dimensionalities ( $D = 15$  and  $D = 128$ ) are maintained to a large extent even when the features are *not* distributed uniformly.

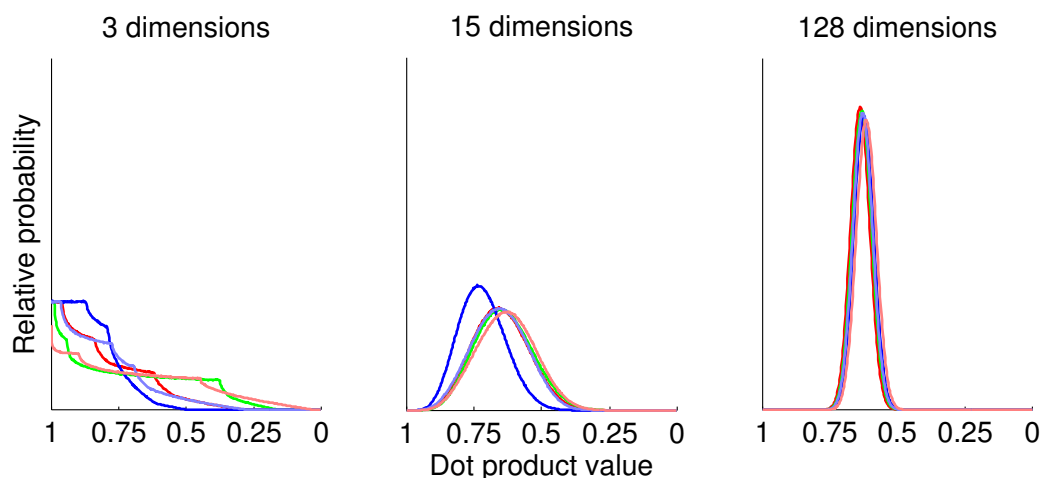


Figure 3.7: Distributions of dot product values between a randomly selected point and all other points on a  $D$ -dimensional hypersphere (positive hyperquadrant only) for 3, 15, and 128 dimensional spaces. Each of the diagrams shows the distribution for five randomly selected points.

### SIFT features set

Here we consider an important case [40] of Scale Invariant Feature Transform (SIFT) descriptor features [35]. Space populated by a set of SIFT features extracted from real scenes differs from the above example in that it can safely be claimed that it is not uniform (it is a matter of fact that non-uniformity of the data is what makes nearest neighbor approaches work). The dimensionality of SIFT features is  $D = 128$ . In Figure 3.8, distributions  $p(r)$  of dot products for five different, randomly chosen SIFTs are shown. The set of SIFT features was extracted from benchmark data set provided by Stewénius and Nistér [58]. The previously discussed high-dimensional effects are clearly retained. In particular, the distributions start with very low values, and exhibit rapid growth as the dot product values fall. The distribution curves are smoothly shaped, and although some of the curves are bimodal, the shapes of distributions are at least qualitatively similar.

### Finding meaningful nearest neighbors

We regard the distribution of dot product values between a query feature and the rest of the features as being composed of the true matches distribution, and the background distribution of irrelevant match candidates (background distribution, for short; see Figure 3.9).

We call a nearest neighbor *meaningful* if it is an outlier to the background distribution. We do not make hard decisions on which nearest neighbors are outliers; instead we assign each neighbor a weight which takes into account how likely it is that the feature is an outlier. We proceed as follows:

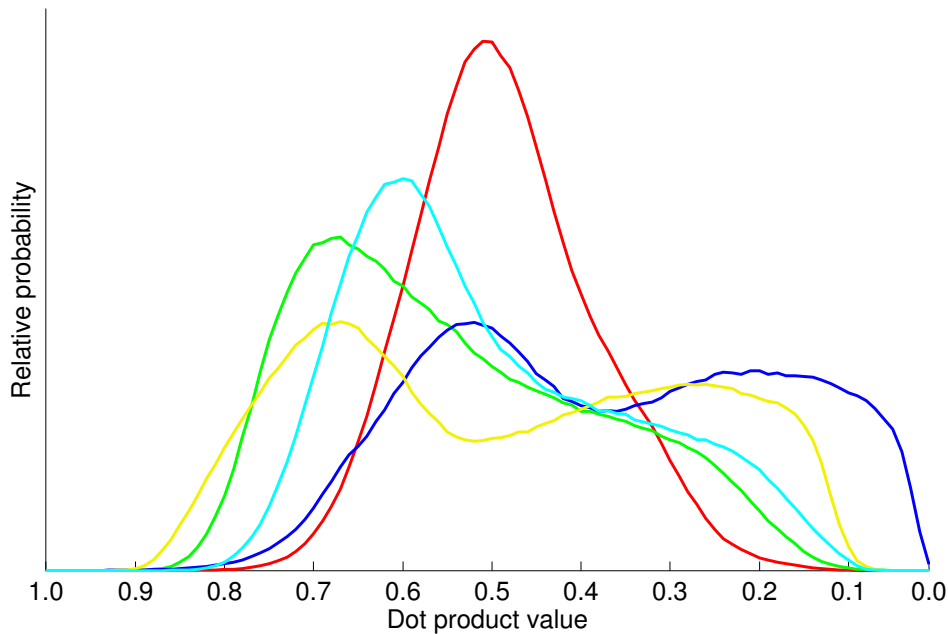


Figure 3.8: Distribution of dot product values between five randomly selected query features and all other (seven million) features extracted from recognition benchmark images [58].

1. For a given query feature, we take an extended neighborhood given by its  $k$  nearest neighbors. A working assumption is that the majority of these  $k$  neighbors are from the background distribution.
2. The growth of the background distribution within the extended neighborhood is modeled by an exponential distribution. Such choice is motivated by the good fit of the model to the data.
3. Nearest neighbors are weighted using a measure which takes into account the absolute similarity to the query and the likeliness of a neighbor being an outlier to the background distribution, and therefore a true match.

In the following, we discuss the three steps in detail.

### Extended neighborhood

Extended neighborhood is given by  $k$  nearest neighbors of the query point. The only requirement on the cardinality of this set (i.e. value of  $k$ ) comes from the consideration that it should enable estimation of the background distribution. Thus, it should be much larger than the expected number of true matches. In regards to that,  $k = 100$  was used in our applications as we expected around 10 true matches per query. For a given query point  $q$ , we denote the nearest neighbors  $v_i, i = 1, \dots, k$ . We assume that these are sorted in increasing distance from the query (i.e. descending order of dot product value).

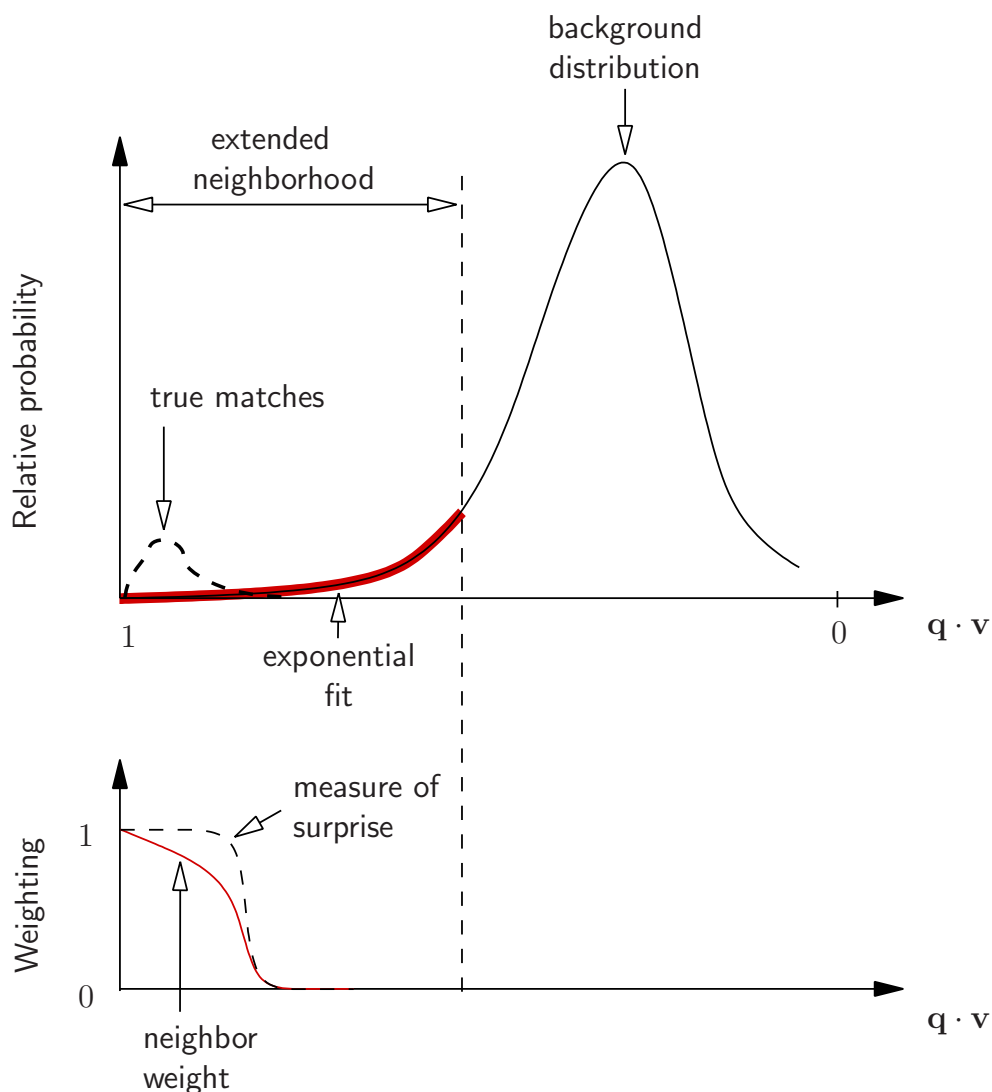


Figure 3.9: Finding meaningful nearest neighbors. The top graph shows two components of the distribution of dot product values between a query feature  $\mathbf{q}$  and a feature vector set  $\{\mathbf{v}_i\}$ . The distribution is composed of a background distribution of irrelevant match candidates, and true matches distribution (note that the true matches distribution is displayed overscaled for sake of clarity). The feature vector set is first restricted to an extended neighborhood of  $\mathbf{q}$  given by  $k$  nearest neighbors. Exponential fit is then applied in the extended neighborhood, with the goal of estimating a simple parametric model of the background distribution (thick red solid curve). Subsequently (see the bottom graph), each feature vector is assigned a measure of surprise (see Eq. 3.5) which quantifies the likeliness of a feature to be an outlier to the background distribution, and therefore, a true match. Each of the nearest neighbors is then assigned a final weight which is a combination of similarity to a query feature and the measure of surprise (see Eq. 3.6).



**Exponential fit**

The dot product values are transformed as

$$s_i = \frac{\mathbf{q} \cdot \mathbf{v}_i - \mathbf{q} \cdot \mathbf{v}_k}{1 - \mathbf{q} \cdot \mathbf{v}_k}, \quad i = 1, \dots, k, \quad (3.2)$$

to normalize the range of similarities to  $[0, 1]$ . Subsequently, the distribution of  $s_i$ 's is fitted by an exponential distribution

$$p(s) = \lambda e^{-\lambda s}. \quad (3.3)$$

The parameter  $\lambda$  is found in closed form, using the maximization of log-likelihood of occurrence of  $s_i$ 's given  $\lambda$ , resulting in

$$\lambda = (\bar{s})^{-1}, \quad (3.4)$$

where  $\bar{s}$  is the mean of  $s_i$ 's.

**Scoring the nearest neighbors**

All nearest neighbors are assigned a weight  $c_i$  which takes into account two determining factors: the similarity  $s_i$  (defined in Eq. 3.2) of a query feature to the neighbor, and the likeliness  $t_i$  of the neighbor being a true match. We employ the following function for assigning the value of  $t_i$ :

$$t_i = (1 - e^{-\lambda s_i})^k, \quad (3.5)$$

which expresses the likeliness of the value  $s_i$  being an outlier to the assumed exponential distribution and is directly related to measures of surprise [4]. The final weight  $c_i$  is computed as

$$c_i = s_i t_i. \quad (3.6)$$

When applied to image matching, meaningful nearest neighbors are independently weighted for each query feature. The sum of weights then defines the similarity of a reference image to the query image.

In Figure 3.10, an example is presented of query features that have meaningful nearest neighbors among the features detected in one of the reference panoramas.

**3.2.2 Approximate nearest neighbors search**

The high-dimensional feature matching method presented in the previous section uses nearest neighbors search to identify the extended neighborhood of a given query feature. To make search for nearest neighbors more efficient, we present in this section an approximate nearest neighbors search method inspired by the work of Olshausen and Field [47]. By employing sparse coding with an overcomplete basis set, similarity of vectors can be measured not only by similarity of projections to basis vectors but also through a similar subset of bases selected for vector representation. See Figure 3.11 for a low-dimensional illustration of sparse coding with and overcomplete basis set, and Figure 3.12 for an explanation how sparsity can speed up nearest neighbors search.



Figure 3.10: The top image depicts a random selection of 50 query features (yellow ellipses) out of 1829 features detected in this query image. The bottom image depicts 50 query features that have meaningful nearest neighbors with the highest weight among the features detected in one of the reference panoramas. Note, that the selected query features in the bottom image are more likely to represent stable visual structures such as windows, ornaments and letters, and are therefore more suitable for the image matching.

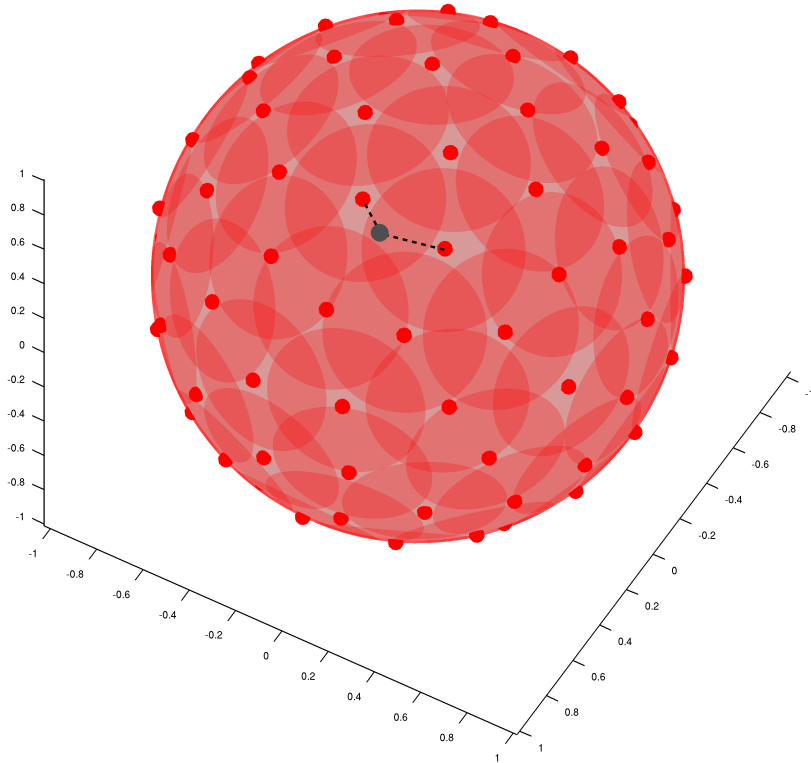


Figure 3.11: Low-dimensional illustration of sparse coding with an overcomplete basis set.  $D$ -dimensional dataspace is populated by an overcomplete basis set  $\{\mathbf{b}_m, m = 1 \dots M\}$  (red dots), where the number of overcomplete bases  $M$  is much larger than the effective dimensionality  $D$  of the dataspace. A data point  $\mathbf{v}$  (a black dot) is represented using distances to all bases which it activates ( $\#$  activated bases  $\ll D$ ). Representation  $\hat{\mathbf{v}} \in \mathbb{R}^M$  of  $\mathbf{v} \in \mathbb{R}^D$  using overcomplete basis set is therefore sparse, i.e., most of the values in  $\hat{\mathbf{v}}$  are zero.

In our approach to nearest neighbors search vectors representing query and input data points are first sparsely projected to an overcomplete basis set. The resulting sparse query and input vectors are in effect compared in the coordinate system defined only by the bases that the sparse vectors have in common. If the sparse query and input vectors are sufficiently similar in this custom-tailored coordinate system, then also the original query and input data points are compared, following a filter-and-refine approach to nearest neighbors search. While exhaustive search needs  $\mathcal{O}(DN)$  operations to identify nearest neighbors of a  $D$ -dimensional query data point among  $N$  input data points, sparse coding enables efficient implementation whose computational complexity is only  $\mathcal{O}(\log N \sqrt{DN} + DL)$ . The number of data points  $L$  selected for exhaustive search in the refinement step can be very small compared to the total number of data points  $N$  while still providing for a very good approximation to an exact nearest neighbors search.

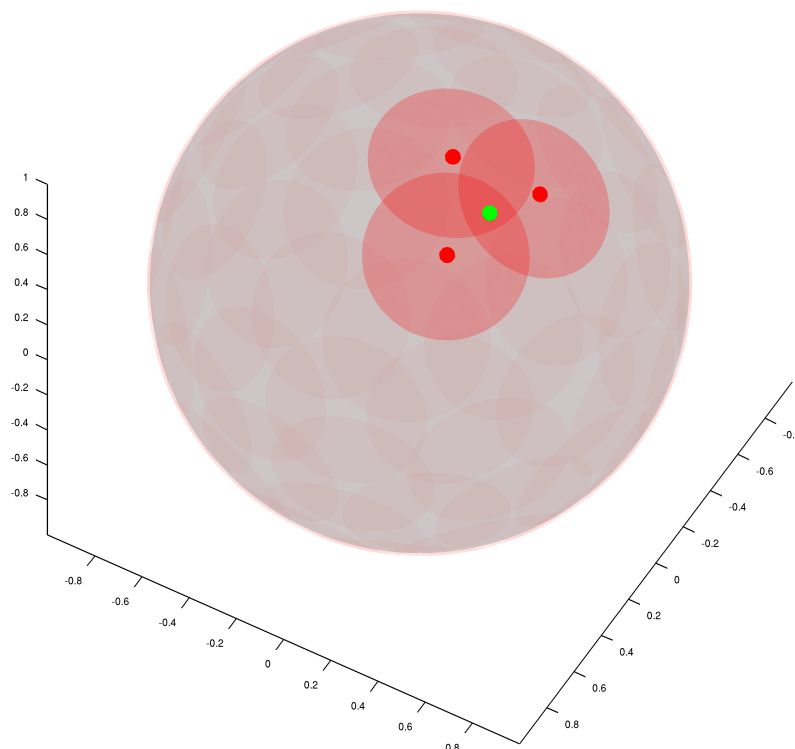


Figure 3.12: A query data point (a green dot) activates just three out of many overcomplete bases. Data points that activate (i.e., are sufficiently close to) some of these three overcomplete bases are candidates for the nearest neighbors of the query data point. The more overcomplete bases they activate, the higher the probability that they are the true nearest neighbors of the query data point.

### Sparse coding

The proposed approximate nearest neighbors search method operates on a set

$$V = \{\mathbf{v} : \mathbf{v} \in \mathbb{R}^D, \|\mathbf{v}\| = 1\}, \quad (3.7)$$

of unit-normalized  $D$ -dimensional vectors representing data points of a particular domain. In addition we assume, that the query and input data points are drawn from the same distribution. As a measure of similarity between two data points it uses dot product between respective vectors in  $V$ .

For the sparse coding, we assume to have an overcomplete basis set

$$B = \left\{ (\mathbf{b}, \rho) : \mathbf{b} \in \mathbb{R}^D, \|\mathbf{b}\| = 1, \rho \in \left[0, \frac{\pi}{2}\right] \right\}, |B| = M. \quad (3.8)$$

A basis is defined as a pair of a vector  $\mathbf{b}$  and a scalar  $\rho$ . The vector  $\mathbf{b}$  specifies the position of the basis in the data space, while the scalar  $\rho$  defines the activation radius of the basis. We say that a basis  $(\mathbf{b}, \rho) \in B$  is *activated* by a data point if the data points' vector  $\mathbf{v}$  is within the angle  $\rho$  from the vector  $\mathbf{b}$ , i.e., if

$$\mathbf{b} \cdot \mathbf{v} > \cos \rho, \quad (\mathbf{b}, \rho) \in B, \mathbf{v} \in V. \quad (3.9)$$

In order to uniquely index overcomplete bases in  $B$  we define a bijective mapping

$$(\mathbf{b}_m, \rho_m) : 1, \dots, M \mapsto B. \quad (3.10)$$

Sparse coding of a vector  $\mathbf{v} \in V$  is done by projecting it to the set of overcomplete bases  $B$ . Values of the projections to the activated bases are retained, while others are set to zero. Elements of the resulting sparse vector  $\tilde{\mathbf{v}}$  are

$$\tilde{v}_m = \begin{cases} \mathbf{b}_m \cdot \mathbf{v} - \cos(\rho_m) & \mathbf{b}_m \cdot \mathbf{v} > \cos(\rho_m) \\ 0 & \text{otherwise.} \end{cases} \quad (3.11)$$

where

$$(\mathbf{b}_m, \rho_m) \in B, m = 1, \dots, M. \quad (3.12)$$

The non-zero elements of the sparse vector  $\tilde{\mathbf{v}}$  have the cosine of the basis' activation radius  $\rho_m$  deducted, following a similar intuition as in [28], that the value of the projection of a data point to a basis' vector has a strong impact on the probability that the two nearby data points are represented by the same basis.

### Filter-and-refine

Given a set

$$I \subset V, |I| = N, \quad (3.13)$$

of  $N$  unit-normalized  $D$ -dimensional vectors representing input data points, the filtering of data points is done by computing cosine similarity

$$h_s = \frac{\tilde{\mathbf{q}} \cdot \tilde{\mathbf{v}}}{\|\tilde{\mathbf{q}}\| \cdot \|\tilde{\mathbf{v}}\|}, \quad \mathbf{v} \in I, \quad (3.14)$$

between sparsely coded query data point  $\mathbf{q} \in V$  and all the input data points in  $I$ .

In the refinement step, the nearest neighbors of the query data point are identified by performing exhaustive search only among the  $L$  input data points with the highest value of the similarity  $h_s$ .

### Efficient implementation of sparse matrix-sparse vector multiplication

By sparsely projecting vectors representing data points to an overcomplete basis set, we transform the problem of efficient nearest neighbors search into a problem of efficient sparse matrix-sparse vector multiplication of the form

$$\tilde{\mathbf{R}} \frac{\tilde{\mathbf{q}}}{\|\tilde{\mathbf{q}}\|} = \mathbf{h}_s, \quad (3.15)$$

where

$$\tilde{\mathbf{R}} = \begin{bmatrix} \tilde{\mathbf{v}}_1^T / \|\tilde{\mathbf{v}}_1^T\| \\ \tilde{\mathbf{v}}_2^T / \|\tilde{\mathbf{v}}_2^T\| \\ \vdots \\ \tilde{\mathbf{v}}_N^T / \|\tilde{\mathbf{v}}_N^T\| \end{bmatrix} \quad (3.16)$$

is a  $N$  by  $M$  sparse matrix of stacked, sparsely encoded and then unit-normalized vectors representing input data points.

Our method for efficient sparse matrix-sparse vector multiplication builds upon the transpose approach of [21]. In the transpose approach the sparse matrix is represented column-wise, where each column is a list of pairs (index, value) and only the indices of the non-zero components are kept. The transpose approach takes advantage of the fact that only the columns of the sparse matrix that correspond to the non-zero elements of the sparse vector contribute to the end result of the multiplication.

Using the transpose approach, we further transform the problem of efficient sparse matrix-sparse vector multiplication into a problem of efficient merging of columns, i.e. lists of pairs (index, value), that correspond to non-zero values of the sparse query vector  $\tilde{\mathbf{q}}$ . As in [21] we compute successive refinements of  $\mathbf{h}_s$  iteratively, starting with all values in  $\mathbf{h}_s$  set to zero and then adding non-zero elements in the columns corresponding to non-zero values of  $\tilde{\mathbf{q}}$ , one column after another.

In order to analyze computational complexity of the transpose approach to efficient sparse matrix-sparse vector multiplication, we must make two requirements on the over-complete basis set  $B$  (see Section 3.2.2 for a description of how to acquire such an over-complete basis set).

1. Sparse coding of both query and input data points should result in sparse vectors with sparsity  $s$ , i.e., with  $s$  non-zero elements, on the average.
2. Probability that a basis  $(\mathbf{b}, \rho) \in B$  is activated by a data point, should be  $s/M$ .

When calculating similarities  $\mathbf{h}_s$  using Eq. (3.15), the first requirement ensures that only  $s$  columns of  $\tilde{\mathbf{R}}$  need to be considered, while the second requirement guarantees that there are  $(s/M) \cdot N$  non-zero elements in a column, on the average. Therefore, the computational complexity of the transpose approach to efficient sparse matrix-sparse vector multiplication is

$$\mathcal{O}\left(\frac{s^2 N}{M}\right). \quad (3.17)$$

### Complexity analysis

The presented approximate nearest neighbors search method consists of three parts. Firstly, a  $D$ -dimensional vector representing a query data point is projected to an overcomplete set of  $M$  bases resulting in a sparse vector  $\tilde{\mathbf{q}}$  with only  $s$  out of  $M$  non-zero elements, on the average. Secondly, sparse coding of both query and input data points enables efficient calculation of similarity  $\mathbf{h}_s$  using the transpose approach to sparse matrix-sparse vector multiplication method presented in the previous section. Finally, in the refinement step,  $L$  data

points with the highest similarity  $h_s$  are exhaustively compared with the query data point. The computational complexity of the method expressed in big O notation as a function of  $M$  is therefore

$$\mathcal{O}(f(M)), f(M) = DM + \frac{s^2 N}{M} + DL. \quad (3.18)$$

Function  $f(M)$  reaches a minimum when its first derivative equals zero

$$f'(M) = 0 \Rightarrow D - \frac{s^2 N}{M^2} = 0 \Rightarrow M = s\sqrt{\frac{N}{D}}, \quad (3.19)$$

which tells us what is the optimal cardinality  $M$  of the overcomplete basis set in order to achieve the lowest computational complexity for a given set of parameters  $s$ ,  $D$ , and  $N$ . When the optimal number of overcomplete bases is used, the computational complexity of the presented approximate nearest neighbors search is

$$\mathcal{O}\left(s\sqrt{DN} + DL\right). \quad (3.20)$$

Because the typical values of  $s$  have order  $\mathcal{O}(\log N)$ , the above equation reduces to

$$\mathcal{O}\left(\log N\sqrt{DN} + DL\right). \quad (3.21)$$

### Learning overcomplete basis set

Learning of overcomplete basis set starts with an initial set of overcomplete bases

$$B_0 = \{(\mathbf{b}, \rho) : (\mathbf{b}, \rho) \in B, \mathbf{b} \in X_+^D, X_+ \sim |N(0, 1)|, \rho = 0\},$$

$$|B_0| = M, \quad (3.22)$$

with vector component randomly drawn from the points on the intersection of the positive  $D$ -dimensional hyperquadrant and the surface of a unit  $D$ -dimensional sphere, while the initial activation radius  $\rho$  is set to 0. We achieve uniform sampling from the surface of the unit hypersphere by generating and normalizing a  $D$ -dimensional vector composed of independent standard normal variates [37].

At each learning step  $i$  we draw randomly a data point  $\mathbf{v}$  from the input set  $I$  and sparsely project it to the current set of overcomplete bases. resulting in a set

$$A_i = \{(\mathbf{b}, \rho) : (\mathbf{b}, \rho) \in B_i, \mathbf{b} \cdot \mathbf{v} > \cos(\rho)\} \quad (3.23)$$

that is composed of bases activated by the data point  $\mathbf{v}$ . The set of overcomplete bases is updated at each learning step using the update rule

$$B_{i+1} = \left\{ \left( \frac{\mathbf{b} + \Delta\mathbf{b}_\alpha + \Delta\mathbf{b}_\beta}{\|\mathbf{b} + \Delta\mathbf{b}_\alpha + \Delta\mathbf{b}_\beta\|}, \rho - \gamma \right) : (\mathbf{b}, \rho) \in A_i \right\} \cup \left\{ (\mathbf{b}, \rho + p_a \gamma) : (\mathbf{b}, \rho) \in (B_i \setminus A_i) \right\}. \quad (3.24)$$

For the activated bases, the update rule changes the position in the data space of a particular basis in direction specified by the sum of values of  $\Delta \mathbf{b}_\alpha$  and  $\Delta \mathbf{b}_\beta$ , and decreases the activation radius of the basis by the value of parameter  $\gamma$ . For the non activated bases, the position of a basis in the data space does not change, while the activation radius is increased by the  $p_a \gamma$ .

In Section 3.2.2 we have defined two requirements on the overcomplete basis set  $B$ . The first requirement, that sparse encoding of query and input data points should result in sparse vectors with sparsity  $s$  is enforced by changing the positions of the bases, where the direction and the magnitude of the change in the position of a basis is controlled by the sum of values of  $\Delta \mathbf{b}_\alpha$  and  $\Delta \mathbf{b}_\beta$ . The value

$$\Delta \mathbf{b}_\alpha = \frac{\alpha}{|A|} (\mathbf{v} - \mathbf{b}), \quad (3.25)$$

represents *attraction*, i.e., the change in the position of a basis towards the position of the data point  $\mathbf{v}$ . The magnitude of the attraction is controlled by the parameter  $\alpha$  and it is higher if the part of the data space where the data point  $\mathbf{v}$  resides is not well covered with the overcomplete basis set, thus forcing the set of overcomplete bases to span the entire data space. The value

$$\Delta \mathbf{b}_\beta = \beta \sum_{\substack{(\hat{\mathbf{b}}, \hat{\rho}) \in A_i, \\ \hat{\mathbf{b}} \neq \mathbf{b}}} \left( 2 \sin \frac{\rho + \hat{\rho}}{2} - \|\mathbf{b} - \hat{\mathbf{b}}\| \right) \frac{\mathbf{b} - \hat{\mathbf{b}}}{\|\mathbf{b} - \hat{\mathbf{b}}\|}. \quad (3.26)$$

represents *lateral inhibition* which incurs a penalty on the frequent co-activation of bases by the same data point, thus encouraging sparse representation. The lateral inhibition changes the position of a basis in the direction away from the position of frequently co-activated basis. The magnitude of lateral inhibition is controlled by the parameter  $\beta$  and it decreases with the distance between the two frequently co-activated bases.

The second requirement on the overcomplete basis set  $B$  defined in Section 3.2.2 that a basis  $(\mathbf{b}, \rho) \in B$  should be activated by the data point  $\mathbf{v}$  with the probability  $s/M$  is enforced by using a Monte Carlo method. In this method, on every learning iteration, activation radii of activated bases are decreased by the parameter  $\gamma$ , while the activation radii of non-activated bases are increased by  $p_a \gamma$  where

$$p_a = \frac{s}{M + s}. \quad (3.27)$$

The principle behind this method is that a basis should be activated by every  $M/s$ -th randomly drawn data point, if the probability of activation of this basis would really be  $s/M$ , resulting in a stationary value of the basis' activation radius. But if the actual probability of the activation is lower than  $s/M$ , the activation radius would increase resulting also in the increased probability of the activation of the basis, and vice versa.



### 3.3 Estimation of geometric relations

The meaningful nearest neighbors of the query features are first used as weighted votes for respective reference panoramas. The reference panoramas with the most votes are considered as potentially matching, i.e., they depict the same scene as the query image. For each potentially matching reference panorama, we try to estimate its geometric relation to the query image using the meaningful nearest neighbors and the epipolar geometry constraint. The reference panoramas for which estimation of geometric relations is unsuccessful are rejected.

We know the exact position and camera orientation of the matching reference panoramas, and hyperlinks to interesting information have been annotated on them (see Chapter 4 for details about the data acquisition and annotation process). The information about the position and the camera orientation is used to triangulate position and camera orientation of the query image [56], while the hyperlinks annotated on the matching reference panoramas are transferred to the query image.

#### 3.3.1 Epipolar geometry estimation

The epipolar geometry relates two views of the same scene independently of scene structure, and only depends on the cameras' internal parameters and relative pose [22] (see Figure 3.13 for a brief explanation of epipolar geometry constraint).

The epipolar constraint can be expressed in the form of a fundamental matrix  $F$ ,

$$\mathbf{x}'F\mathbf{x} = 0, \quad (3.28)$$

or an essential matrix  $E$ ,

$$\mathbf{x}'E\mathbf{x} = 0, \quad (3.29)$$

that can be estimated if a sufficient number of correspondences is known [22]. We will refer to the set of correspondences with just the sufficient number of correspondences as a minimum set. The estimation of the fundamental matrix requires the minimum set of seven correspondences, while the estimation of the essential matrix requires a minimum set of only five correspondences (using so called *five-point algorithm* [45]). Due to reasons that will become apparent in the next section, we prefer the algorithm with the smallest cardinality of the minimum set. Therefore in our method we use an implementation of the five-point algorithm by [57]. In addition, the five-point algorithm works even if all the correspondences in the minimum set lie on a plane (e.g., a facade of a building). Such a configuration is very common in urban environments and it is not tolerated by other algorithms for epipolar geometry estimation.

The five-point algorithm requires that the internal parameters of the camera are known. In case of camera phones, the most important internal parameter is the focal length. Due to the small form factor, most of the camera phones on the market today have a fixed focal length (i.e. no zoom) that can be acquired from the manufacturers, while the camera phones with zoom store information about the focal length in the EXIF header of the image. We can assume standard values for other internal parameters of the camera [22].

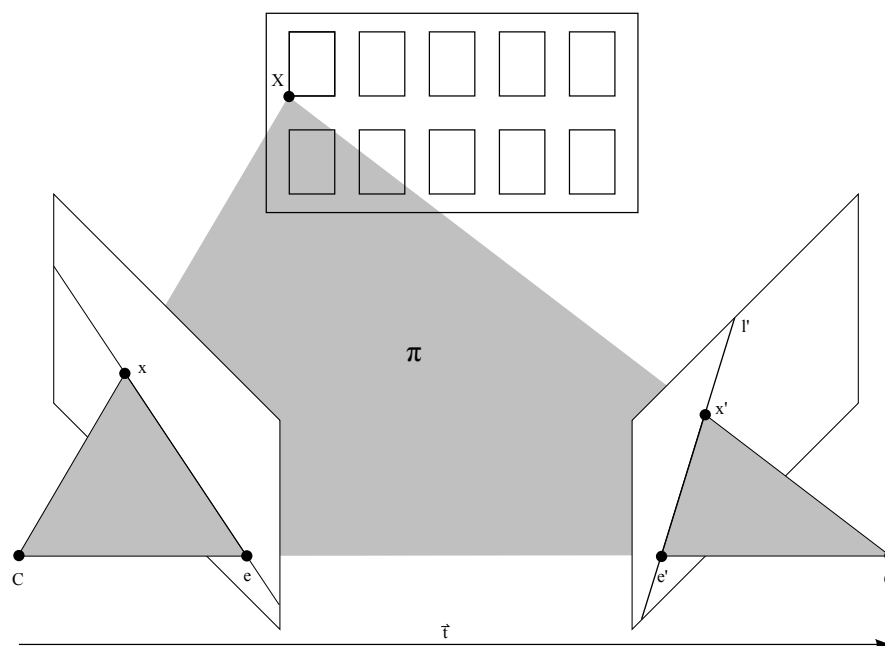


Figure 3.13: A point  $X$  in 3-space is imaged in two views, at  $x$  in the first, and  $x'$  in the second (a pair of  $x$  and  $x'$  is called a correspondence). The epipolar constraint states that if we know the epipolar geometry and the position of  $x$  in the first view then we can limit the search for the position of  $x'$  in the second view to a line  $l'$ . The line  $l'$  is called epipolar line and the plane  $\pi$  that includes the image points  $x$  and  $x'$ , space point  $X$ , and camera centres is called epipolar plane. Two epipolar planes intersect in the line called a baseline. Intersection of the baseline with the image plane is called the epipole.

### 3.3.2 In search of true correspondences

Existing matching algorithms cannot guarantee that all correspondences are true correspondences, i.e., that they are projections of the same structure in 3D world [73], so we resort to a hypothesize-and-test approach [17] in order to find a hypothesis that is the most consistent with the tentative correspondences set. In the hypothesize-and-test approach we first construct a hypothesis and then we test it on the tentative correspondences set. The hypothesis that is the most consistent with the tentative correspondences set is selected as the representative one. Given unlimited time, we could try all possible hypotheses and therefore the success of the hypothesize-and-test approach would depend solely on how successful we are at identifying the correct hypothesis by testing it on the tentative correspondences set. In real applications we do not have unlimited time, and that is why we must have a strategy for selective sampling of the space of possible hypothesis. The first and still dominant model that follows the hypothesize-and-test approach is RANdom SAMpling and Consensus (RANSAC) paradigm of [17]. In RANSAC, sampling of the space of possible hypothesis is done by repeated random sampling of the correspondences set, where each

random sample is used for calculation of a hypothesis and the criteria for selection of the best hypothesis is the number of inliers, i.e., correspondences that are consistent with the hypothesis. All correspondences in the random sample must be true correspondences in order for the hypothesis calculated from the sample to be correct and therefore it is preferable that the random sample is the minimum set, i.e., the set of correspondences with just the sufficient number of correspondences for calculation of the hypothesis.

RANSAC assesses a hypothesis by simply counting the number of inliers. The more principled way would be to calculate the posterior probability  $p(M_h|\mathcal{C})$  of hypothesis  $M_h$  given a correspondences set  $\mathcal{C}$  with  $n$  elements. The posterior probability  $p(M_h|\mathcal{C})$  cannot be measured directly, therefore Torr and Zisserman have introduced MLESAC [63], in which hypotheses are scored using the likelihood

$$p(\mathcal{C}|M_h) = \frac{p(M_h|\mathcal{C})p(\mathcal{C})}{p(M_h)}, \quad (3.30)$$

where the prior probability of the correspondences,  $p(\mathcal{C})$ , is a constant.  $p(\mathcal{C}|M_h)$ . Assuming uniform prior  $p(M_h)$  and constant  $p(\mathcal{C})$ , the likelihood  $p(\mathcal{C}|M_h)$  is directly proportional to  $p(M_h|\mathcal{C})$ .

Calculation of  $p(\mathcal{C}|M_h)$  is problem dependent. MLESAC was developed for estimation of the fundamental matrix from feature correspondences and it is therefore applicable also to our problem of estimation of the essential matrix. In MLESAC, an assumption is made that the likelihood  $p(\mathcal{C}|M_h)$  depends on the probability of residual error  $r_{hi}$  of each correspondence given the hypothesis  $M_h$  and that the probability of each residual is independent, where residual error is defined as the distance between the observed and the anticipated position of the feature in the image. Then,

$$p(\mathcal{C}|M_h) = \prod_{i=1}^n p(r_{hi}|M_h). \quad (3.31)$$

According to MLESAC, conditional probability of observing a residual  $r_{hi}$  given the hypothesis  $M_h$  is

$$p(r_{hi}|M_h) = \left( \frac{1}{2\pi\sigma^2} e^{-r_{hi}^2/2\sigma^2} \right) p(v_i) + \left( \frac{1}{w} \right) (1 - p(v_i)), \quad (3.32)$$

which is a mixture model of a Gaussian and uniform model for the case of true and false correspondences, respectively, and the parameters  $w$  and  $\sigma$  should be chosen based on the selection of the feature detector and class of imagery. The probability  $p(v_i)$  expresses the prior probability that the  $i$ -th correspondence is a true correspondence. According to MLESAC, conditional probability of observing a residual  $r_{hi}$  given the hypothesis  $M_h$  can be expressed as a mixture model of a Gaussian and uniform model for the case of true and false correspondences, respectively, with a probability  $p(v_i)$  expressing the prior probability that the  $i$ -th correspondence is a true correspondence.

In MLESAC, the prior probability of a true correspondence  $p(v_i)$  is assumed uniform within the correspondences set (i.e.,  $p(v_i) = p(v)$ ) and is, in principle, assumed to be independent of the hypothesis. Torr and Zisserman also do not assume any prior knowledge of  $p(v)$  and they suggest estimating  $p(v)$  for each hypothesis separately.

### 3.3.3 Deriving the prior probability of a true correspondence

In [62], the authors argue that MLESAC’s probabilistic approach to random sampling and consensus could be improved if an estimate of the prior probability of a true correspondence  $p(v_i)$  was available. In the feature matching step of our method, we have first identified and then weighted the meaningful nearest neighbors. Experimentally we have observed that the weight attributed to the meaningful nearest neighbor is well correlated with the probability that the query feature and the meaningful nearest neighbor form a true correspondence. Therefore, in our method, we use the weight attributed to the meaningful nearest neighbors in order to calculate the prior probability of a true correspondence  $p(v_i)$ .

Due to the presence of repetitive visual structures, a reference feature can be matched to more than one query feature. In addition, to reduce the chance of missing a true correspondence, we allow matching of a query feature with more than one reference feature from a single image (i.e., *soft matching*). To express that, due to the uniqueness constraint [59], at most one correspondence per feature can be correct, the prior probability of the  $k$ -th correspondence of a query feature  $i$  with  $n_i$  potentially matching reference features with scores  $s_{ik}$  and validities  $v_{ik}$  ( $k = 1, \dots, n_i$ ), is calculated as in [62]

$$\begin{aligned}
 p(v_{ik} | s_{i1}, \dots, s_{in_i}) = & \\
 & \frac{p(v_{ik} | s_{ik}, n_i) \prod_{j \neq k}^{n_i} p(\bar{v}_{ij} | s_{ij}, n_i)}{\sum_l^{n_i} \left[ p(v_{il} | s_{il}, n_i) \prod_{j \neq l}^{n_i} p(\bar{v}_{ij} | s_{ij}, n_i) \right] + \prod_j^{n_i} p(\bar{v}_{ij} | s_{ij}, n_i)}
 \end{aligned} \tag{3.33}$$

In this equation, the numerator gives the probability that the  $k$ -th correspondence of the query feature  $i$  is correct and all other correspondences of this query feature are incorrect. The denominator normalizes the numerator by the sum of probabilities of all correspondences for a given query feature, and the possibility that none are correct.

### 3.3.4 Sampling the space of possible hypothesis

Original RANSAC’s strategy to draw minimum sets from the correspondences set uniformly at random is feasible only when the correspondences set includes more than 50% of true correspondences [64]. If there are less than 50% of true correspondences, the time needed (i.e., the number of hypotheses) to draw the minimum set with only true correspondences included with sufficient probability (e.g., of at least 95%) becomes too long and therefore not acceptable to the user.

In Guided-MLESAC of [62], instead of filling the minimum set uniformly at random, the elements in the minimum set are chosen by a Monte-Carlo method according to  $p(v_i)$ . While this enables identification of the true hypothesis also for cases when the correspondences set includes only 30% of true correspondences, it is still not powerful enough for our target application. Due to the presence of repetitive structures, the correspondences sets

that we have observed in our application usually include only between 10% and 15% of true correspondences. Historically, the random sampling strategy was chosen primarily because there was no reliable measure of validity of a correspondence, and secondarily, to avoid interdependencies between correspondences included in the minimum set (e.g., all features lying on a single plane or a small part of the image). In our method, the weight attributed to the meaningful nearest neighbor can be used as a measure of correspondence validity. Therefore, instead of random sampling strategy, we employ a deterministic sampling strategy that constructs minimum sets by permuting the top  $l$  tentative correspondences with the highest weight. The number of hypotheses thus generated is  $\binom{l}{l-r}$ , where  $r$  is the cardinality of the minimum set.

While we have indeed observed interdependencies between correspondences included in the minimum set that would not be present in the case of a random sampling strategy, the interdependencies did not noticeably diminish the success of geometry estimation. In our opinion, any sampling strategy that includes information about probability of a correspondence being a true correspondence based on the visual similarity of respective image regions, will sample some parts of the scene with higher probability than others due to the nature of camera motion. For example, if the camera moves in the direction of a facade A and parallel to the facade B, then the images of the features detected on facade A will undergo only a similarity transformation (i.e., only position and scale will change), while the images of the features detected on the facade B will undergo a full perspective transformation that is much harder to model.

### 3.3.5 Structure estimation

The epipolar geometry constraint does not differentiate between physically possible configurations and configurations in which some features are behind the camera, thus violating the so called *cheirality constraint* [22]. Tentative correspondences that are consistent with the hypothesized epipolar geometry but are violating the cheirality constraint may impede success of geometry estimation by giving support to incorrect hypotheses. In order to remove the influence of such correspondences, we first decompose the essential matrix into four possible combinations of rotation and translation of the first camera relative to the second camera using the method described in [36]. For each of the four combinations, we estimate the position of the features in 3D world by intersecting rays back-projected from the two cameras' centers through the images of features in both images, respectively, using the method described in [36]. The combination of the camera rotation and translation with the most features in a physically viable configuration (i.e., in front of the camera) is chosen, and the correspondences that violate the cheirality constraint are removed from the set of true correspondences for the hypothesis that is being tested.

### 3.3.6 Detection of pure rotation

If the distance between the first and the second camera position is small compared to the distance to the scene, the calculation of the translational component of the camera motion is unreliable. Typically this happens, when the user has shot the query image from (almost)

the same position as the reference panorama. In order to detect such a situation, we check if the rotational matrix  $R$  alone is sufficient to explain the relation between positions of features in the first and second image, respectively, as expressed by

$$\mathbf{x}' = R\mathbf{x}. \quad (3.34)$$

We estimate the rotational matrix  $R$  with the same hypothesize-and-test approach as used for the estimation of the epipolar geometry, but this time we use the Singular Value Decomposition (SVD) to compute the rotational matrix  $R$  from the minimum set of just three correspondences. The correspondences set from which the minimum set is drawn includes only correspondences that are consistent with the estimated epipolar geometry. If the rotation alone is sufficient to explain a large majority (75%) of transformations of feature positions in the first and second image, respectively, we assume that the two images are shot from the same position.

### 3.4 Transfer of hyperlinks

The previous step of our method has provided us with the geometry that relates the query image and matching reference panoramas. In addition, the set of tentative correspondences was purged to include only true correspondences, i.e., correspondences that are consistent with the estimated geometric relations. This enables us to transfer information from the reference panoramas to the query image.

Among the information that we have annotated on the reference panoramas beforehand (see Chapter 4 for details) are also hyperlinks to some interesting information about buildings, logos, banners, monuments and other objects depicted on the reference panoramas and that we expect to also be present on the query image shot by the user. A hyperlink is defined by a polygon that indicates the position of an object of interest within the reference panorama. The features that are detected within the polygon are therefore describing the visual appearance of the object and can serve as an indication of an object's presence.

The transfer of a hyperlink commences by identifying true correspondences that include features of the object. If there is an insufficient number of such correspondences (less than four) we assume that the object is not present in the query image. Otherwise, we verify that the configuration of positions of features in the reference panorama is compatible with the positions of the features in the query image. For that, we assume that the objects in question are planar (or close to planar), and that the compatibility of configurations can be verified using the homography constraint, as expressed by

$$\mathbf{x}' = H\mathbf{x}. \quad (3.35)$$

We estimate the homography matrix  $H$  with the same hypothesize-and-test approach as used for the estimation of the epipolar geometry, but this time we use an algorithm that estimates homography from a minimum set of just three correspondences and known epipolar geometry [22, 60]. In addition, we reject hypotheses for which the homography matrix  $H$  is (almost) singular [68] and hypotheses that do not preserve the polygon orientation. An (almost) singular homography matrix indicates projection of a plane in the first image to a

line in the second image (the local invariant features cannot be detected on a line), while a change of the polygon orientation indicates that either the hypothesis is wrong or the plane is transparent.

The transfer of hyperlinks is done for each matching reference panorama separately. In case, there is more than one hyperlink detected for a particular object, we choose the hyperlink with higher credibility and reject the others.

### 3.5 Visualization of results

The estimated homography matrix  $H$  is used to project the polygon that defines the hyperlink from the reference panorama to the query panorama (see Figure 3.4 for an example). If the projected polygon is too big (e.g. in case of a building) to be fully visible in the query image, it is cut accordingly, so that it does not extend past the borders of the query image.

Finally, we put an icon that indicates a hyperlink to the user in the center of the projected polygon. Once the user taps the icon on his camera phone's (multi-)touch screen, an information pane is displayed with some interesting story about the object, as depicted in Figure 1.2.

In addition, since we know the geometry relating the query image and the nearby reference panoramas, and we know the exact location and orientation from where reference panoramas were shot, we can triangulate the user's position and orientation with an accuracy comparable to GPS [56] (see Figure 3.14 for an example and Chapter 5 as well as [56] for an evaluation of the accuracy of image based localization). Such an image based localization system can augment GPS, or in some circumstances where GPS performs poorly, even replace it.

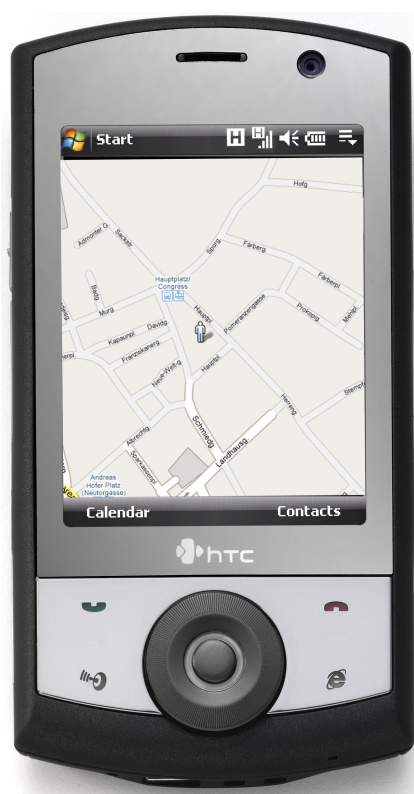


Figure 3.14: The calculated position visualized on a map (indicated by icon of a man) where the query image from Figure 3.1 was shot. In this case, the position is estimated with accuracy of approximately 10 meters.





## Chapter 4

# Reference Data Set

The novel user interface concept presented in Chapter 3 requires a data set of reference panoramas that were collected beforehand and are annotated with information that is of interest to the user. In this chapter we present the process and the tools that we have used in order to collect such a data set.

### 4.1 Acquiring the data set

We have acquired 1284 reference images shot from 107 accurately measured camera positions using an Olympus E-1 digital camera and a five megapixel image resolution. Twelve images shot at a single position together make a panorama view covering  $360^\circ \times 56^\circ$  (see Figure 4.1 for visualization of camera positions on a map and an example of images that together make a panorama)<sup>1</sup>. The data set was acquired in the historical city center of Graz, Austria in October 2007. Reference images were collected in such a way that a large scale acquisition would be feasible.

We have acquired 184 query images using three different cameras (Olympus E-1, Canon IXUS I55, and Nokia N-90 camera phone; see Figure 4.2 for examples) together with accurately measured camera positions. Olympus E-1 is a digital SLR camera with five megapixel sensor, Canon IXUS I55 is a consumer grade digital camera with five megapixel sensor, and Nokia N-90 is a camera phone with two megapixel sensor.

The camera position and orientation of query images were selected by the test subjects that were told to shoot what they might find interesting or at the places where they would feel lost and would therefore need navigation guidance.

The cameras were positioned directly above distinctive objects on the ground (i.e. shafts) with known position provided by the municipality from their geographic information system with positioning error of less than 0.5 m. Due to the abundance of distinctive objects on the ground, this only slightly influenced the selection of camera positions. Reference images were shot with a camera mounted on the tripod. The height of the camera from the ground was 1.5 m.

---

<sup>1</sup>The data set is available online at <http://vicos.fri.uni-lj.si/GUIS107/>

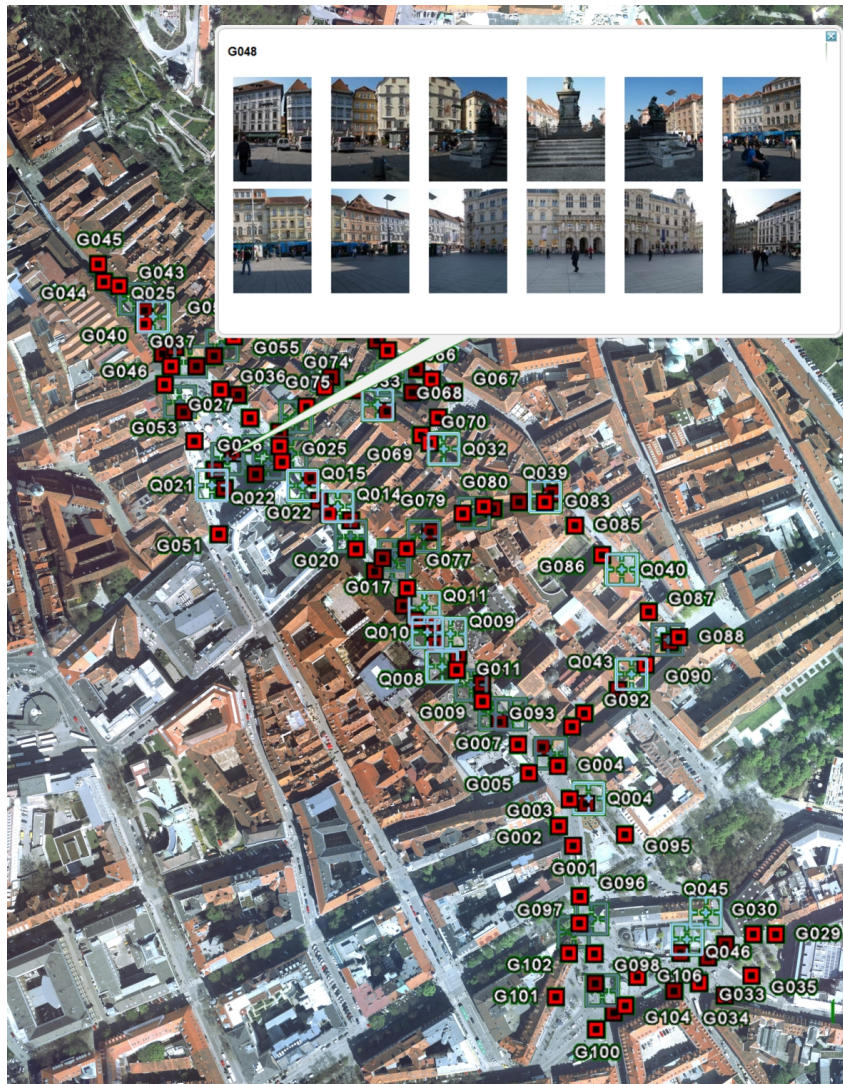


Figure 4.1: The positions where the reference panoramas were shot and an example of images that together make a panorama.



Figure 4.2: An example of query images.



Figure 4.3: An example of stitched reference panoramas with annotated hyperlinks (red polygons).

While there exist several simple and accurate techniques for acquisition of the geographical location of the camera (e.g., satellite positioning, classical surveying techniques, geographic information systems), acquisition of absolute camera orientation is more challenging. Therefore we acquired the absolute camera orientation by registering images themselves using an automatic procedure followed by manual verification. The procedure for registration (estimation of absolute orientation) of reference images proceeded in three steps. In the first step, reference images shot at the same position were stitched in a panorama using the method of [8] (see Figure 4.3 for an example). Features were detected and described in the original images and then put in the common coordinate system. The final registration was further optimized using bundle adjustment [65]. The result of panorama stitching was manually verified in order to assure that all panoramas are correctly stitched. In the second step, reference panoramas were matched against each other in order to estimate epipolar geometry relating panoramas depicting the same scene. The success of epipolar geometry estimation was manually verified by checking whether the image region around the epipoles depict the same scene in both panoramas. Finally, the absolute orientation of panoramas was calculated either by aligning epipoles and known camera positions, or by propagating absolute orientation from one panorama to another using known epipolar constraints.

## 4.2 Hyperlinks annotation tool

In order to streamline the process of annotating the reference panoramas with the hyperlinks to some interesting information about buildings, logos, banners, monuments and other objects, we have developed a hyperlink annotation tool (see Figure 4.4). The application enables the administrator (i) to annotate the image region that represent a building or some object, (ii) to add new hyperlinks, and (iii) to attribute a hyperlink to a particular object. Besides having an ID, each hyperlink has attributed a name, an icon, a short description, and an URL for further information about the object.

Using the hyperlinks annotation tool, we have annotated several hundred historic sights, shops, restaurants, cafés, ice cream parlors, bookshops, pharmacies, buildings, logos, ban-

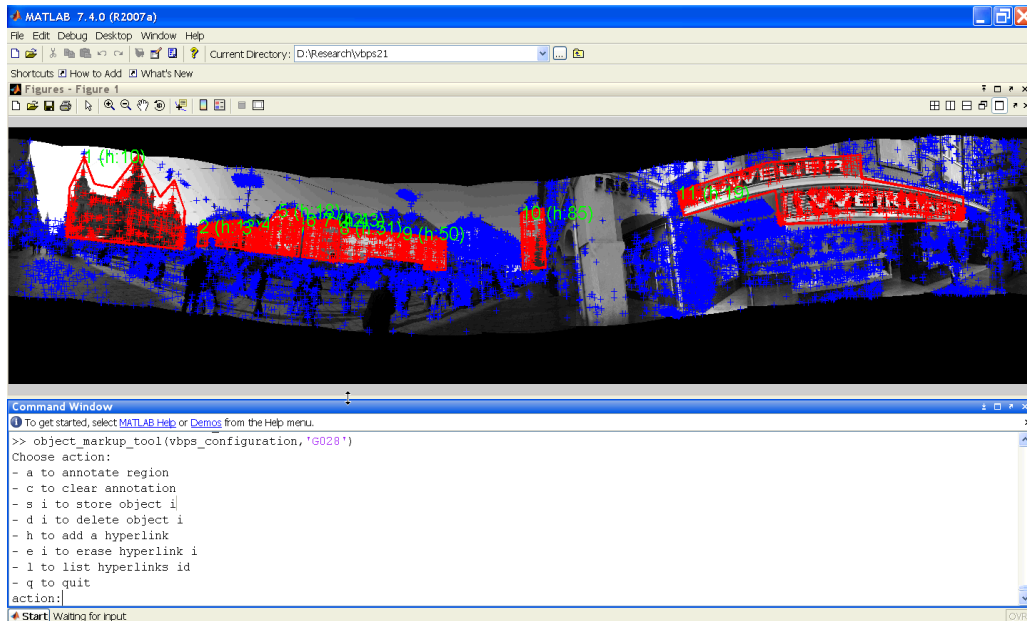


Figure 4.4: A screen shot of the hyperlink annotation tool. The top frame shows the reference panorama that is being annotated, while the bottom frame lists the actions available to the administrator. On the reference panorama, the centers of detected features are shown as plus sign. The red plus signs indicate features that are included in a description of some object. The red polygon indicates the extent of the object within the reference panorama and the numbers next to are object and hyperlinks identifiers, respectively.

ners, monuments, and other objects of interest to the user (see Figure 4.3 for annotation samples).

# Chapter 5

## Results

### 5.1 Performance evaluation of the proposed high-dimensional feature matching method

We have evaluated the performance of the proposed high-dimensional feature matching method on two challenging image data sets, a data set of recognition benchmark images provided by Stewénus and Nistér [58] and the Ljubljana urban image data set<sup>1</sup> collected by ourselves for the purpose of image based localization. Image matching was performed using local invariant features detected by the Maximally Stable Extremal Region (MSER) detector [38] and described by a Scale Invariant Feature Transform (SIFT) descriptor [35]. Besides comparing the performance of our approach to the performance of vocabulary tree based matching of Nistér and Stewénus [46], we also compared it to 1-NN matching and augmented k-NN matching.

In the *1-NN matching* only the nearest neighbor of a query feature was used to vote for the respective image. The similarity of a reference image to a query image was measured by the number of votes the reference image received.

The *augmented k-NN matching* searches for  $k$  nearest neighbors of a given query feature but uses only the nearest neighbors that are within radius  $r$  from the query feature ( $k$  and  $r$  were specifically optimized for each of the image data sets to give the best matching results). Each of these neighbors votes for a respective reference image. The number of votes collected for each reference image is divided by the square root of the number of features detected in that reference image, to account for considerable variation of number of features detected in different reference images. Such score is then used for ranking the reference images.

In experiments presented in this section we have extensively used the novel approximate nearest neighbors search method, presented in Chapter 3, that is ten-times faster than the exhaustive search even in high-dimensional spaces. In order to show the quality of approximation to an exact nearest neighbors search and to validate results of the experiments, we have repeated all the experiments of this section also by using the exact nearest neighbor

---

<sup>1</sup>Ljubljana urban image data set and performance evaluation results are available online at <http://vicos.fri.uni-lj.si/LUIS34/>



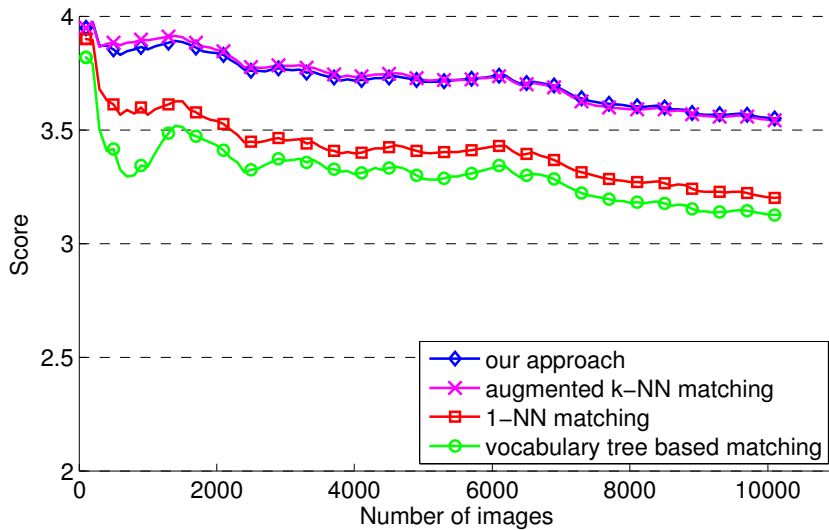


Figure 5.1: Performance of the proposed high-dimensional feature matching method compared to the vocabulary tree based matching [46], 1-NN matching, and augmented k-NN matching. The performance was evaluated using a data set of recognition benchmark images [58]. For the evaluation we used the novel approximate nearest neighbors search method. We validated results by repeating the experiment using the exact nearest neighbor search on the first 1000 images. We show in Figure 5.4 that the difference in results is negligible.

search on a partial subset of images.

### 5.1.1 Recognition benchmark images

The image data set of Stewénus and Nistér [58] contains 10200 images in groups of four that belong together, see [46] for examples. Each image is in turn matched against all other images and scoring is performed by counting how many of the four images in a block (including the query image) are found among the four best matching images. In our evaluation we used SIFT descriptor vectors provided by [58]. The result of our approach compared to other methods is presented in Figure 5.1. For the augmented k-NN matching it was experimentally established that taking 30 nearest neighbors and radius  $r$  of  $32^\circ$  gives best matching results. Our method performed much better than the vocabulary tree based matching and the 1-NN matching, while it performed approximately the same as the augmented k-NN matching that was specifically optimized for this particular image data set.

### 5.1.2 Ljubljana urban image data set

The second image data set consists of 612 reference images of an urban environment covering an area of  $200 \times 200$  square meters. At each of the 34 standpoints, 18 reference images

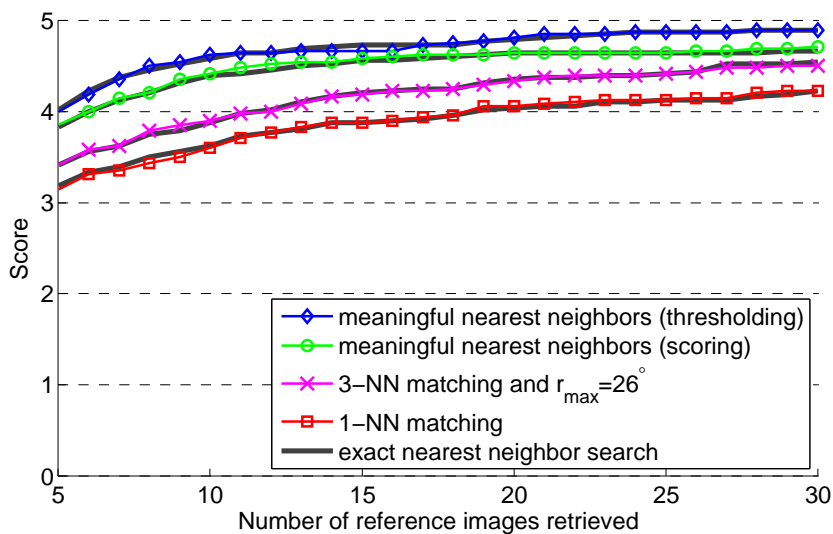


Figure 5.2: Performance of the proposed high-dimensional feature matching method compared to 1-NN matching, and augmented k-NN matching on the Ljubljana urban image data set. The curves show how many of the groundtruth reference images were retrieved among  $n$  top-ranked reference images (the score is upper bounded to 5). The dark, thick lines indicate the results of respective methods using the exact instead of the approximate nearest neighbor search. The processing of all 48 query images took 2 hours using the approximate nearest neighbors search and 19 hours using the exhaustive search.

were captured in overcast weather. Nine months later, 48 query images were captured in sunny weather. For each query image we have manually selected between 10 and 15 best matching reference images that share the most of the scene with the query image, and we took these as groundtruth reference images. The evaluation was performed by retrieving  $n$  most similar reference images and counting how many of the groundtruth reference images were among them. The score was upper bounded to 5 because of a weak boundary between reference images selected and those not selected as the groundtruth reference images but still having at least part of the scene in common with the query image. The superior performance of our approach over the other methods is presented in Figure 5.2. For the augmented k-NN matching it was experimentally established that taking three nearest neighbors and radius  $r$  of  $26^\circ$  gives best matching results.

In Table 5.1 we show the cumulative number of query images for which (at least) the specified number of groundtruth reference images were retrieved among five top-ranked reference images. We show that our approach retrieves at least two groundtruth reference images for 44 out of 48 query images (92%), while the other methods are successful for only 38 (79%) and 37 (77%) query images respectively. This case is important for image based localization because with two matching reference images retrieved from two different standpoints and assuming common ground plane it is possible to triangulate the user



Table 5.1: Cumulative number of query images for which (at least) the specified number of groundtruth reference images were retrieved among five top-ranked reference images. The right-most column shows the number of cases when not a single groundtruth reference image was correctly retrieved. There are 48 query images altogether.

method	#groundtruth reference images					
	5	4	3	2	1	none
our approach	24	33	41	44	45	3
augmented k-NN	20	30	32	38	43	5
1-NN	15	28	29	37	41	7

position [74]. The other important case for image based localization is the number of cases when not a single matching reference image can be correctly retrieved as this prohibits even approximate localization. While augmented k-NN matching could not find any matching reference image for five out of 48 query images (10%) and 1-NN matching was unsuccessful for seven query images (15%), our approach failed for only three query images (6%). As an example of performance of our approach to matching, we show in Figure 5.3 the most similar reference images retrieved for three sample query images.

## 5.2 Performance evaluation of the proposed approximate nearest neighbors search method

### 5.2.1 Retrieval rate

The retrieval rate of the proposed approximate nearest neighbors search method was estimated using 760062 SIFT descriptor vectors detected in the first 1000 recognition benchmark images of [58]. The retrieval rate was computed by comparing the nearest neighbors retrieved by an exact search method to the nearest neighbors retrieved by the approximate search method. The results presented in Figure 5.5 show that the true nearest neighbor is retrieved in the 94.2% of cases and that the retrieval rate is slightly less for second to fifth nearest neighbor. The retrieval rate for the first hundred nearest neighbors is 85.8%.

### 5.2.2 Quality of approximation

With approximate nearest neighbors search methods it is hard to tell whether the missed nearest neighbors are important or not. An approximate nearest neighbors search method can have a very low retrieval rate and still be highly useful if it retrieves all the true matches, while if it misses many true matches the high retrieval rate is to no avail. Therefore, we have compared the meaningful nearest neighbors returned by the high-dimensional feature matching method presented in this thesis, once using approximate and the other time

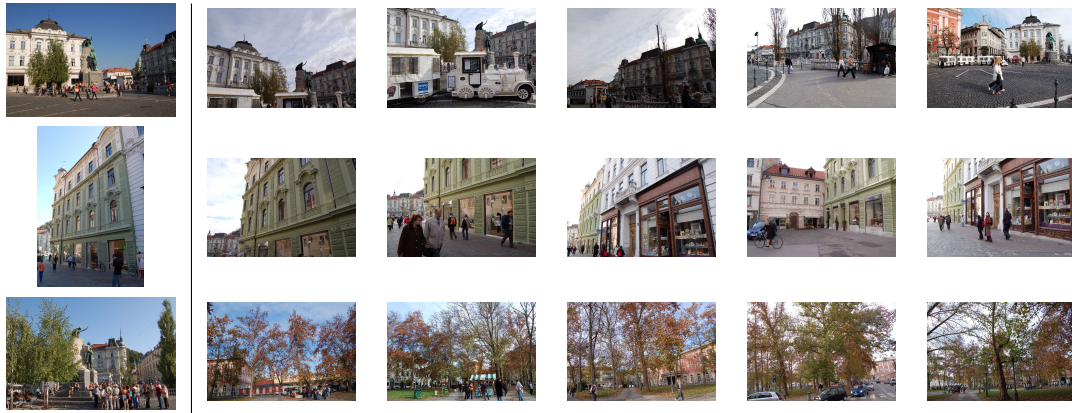


Figure 5.3: Three sample query images (the first column) of the Ljubljana urban image data set and the top five images retrieved by our approach (second to sixth columns). The first row demonstrates perfect performance; the building present in the the query image is also in all top-ranked images. The second row shows similar performance; note however that the second to last image is unsuitable for image-based localization as it depicts the other side of the building with identical facade. The third row is an example of retrieval which seems correct as far as image similarity is concerned, but since the retrieved images do not depict the same scene, this is considered a failure for image-based localization applications.

using exact nearest neighbors search. The meaningful nearest neighbors were weighted as presented in Chapter 3 and the extended query feature neighborhood was identified by searching for  $k = 100$  nearest neighbors. The results of this comparison are given in Figure 5.5. The results show that the approximate method returns most of the meaningful nearest neighbors with the highest weight. While the approximate method retrieves only 85.8% of the first hundred nearest neighbors, this does not result in substantial reduction of meaningful nearest neighbors returned. To further support this claim we have run the evaluation of the proposed matching method, 1-NN matching, and augmented k-NN matching on both, recognition benchmark images of [58] and Ljubljana urban image data set, once using the approximate and the other time using exact nearest neighbors search. From the results presented in Figs. 5.2 and 5.4 it is evident that the difference in performance between the approximate (colored, thin lines) and the exact method (dark, thick lines) is negligible.

### 5.2.3 Speed-up

The approximate nearest neighbors search method enabled us to process all 10200 recognition benchmark images of [58] in 100 hours and all 48 query images of Ljubljana urban image data set in two hours. This required cross matching of seven million SIFT descriptor vectors [58] and matching of 260000 features detected in the query images with 3.8 million features detected in the reference images of Ljubljana urban image data set. For comparison, using exhaustive nearest neighbors search we would need ten times as much, that is

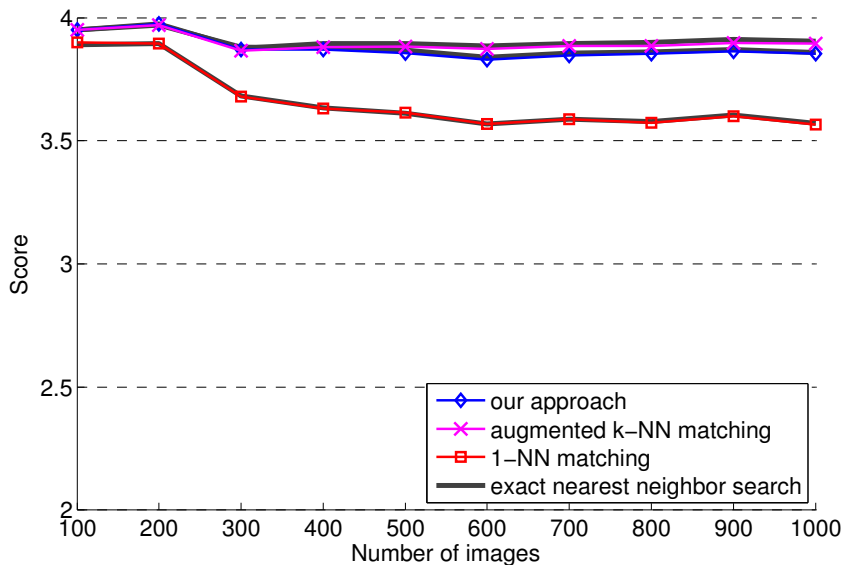


Figure 5.4: Comparison of recognition performance when using the approximate (colored, thin lines) and the exact nearest neighbors search (dark, thick lines) on the first 1000 recognition benchmark images of [58]. The results show that independent of the matching method used the difference in performance between the approximate and the exact method is negligible.

1000 hours, or 41 days, for images of [58] and 19 hours for the query images of Ljubljana urban data set. It is important to note that the performance was evaluated by comparing the execution time of our approximate nearest neighbors search method against the execution time of the highly optimized implementation of the exhaustive search that fully utilized vectorized instructions of a 2.4 GHz Intel Core 2 Duo E6600 processor. The presented approximate nearest neighbors search method preserves data locality of exhaustive search, thus fully exploiting memory hierarchies.

In addition, we have compared our method to hierarchical k-means tree of [43], multiple randomized kd-trees of [54] and exhaustive search using the evaluation framework of [43]. The results of the evaluation have shown that our method is 25% faster than the method based on hierarchical k-means tree, more than three times faster than the multiple randomized kd-trees, and more than 100 times faster than exhaustive search. The evaluation was done using the LUIS-34 dataset [48] with 3.8M SIFT descriptor vectors. Due to the requirement of the concept of meaningful nearest neighbors we have set required quality of approximation at 95% for the first nearest neighbor and at 80% for the first 100 nearest neighbors.

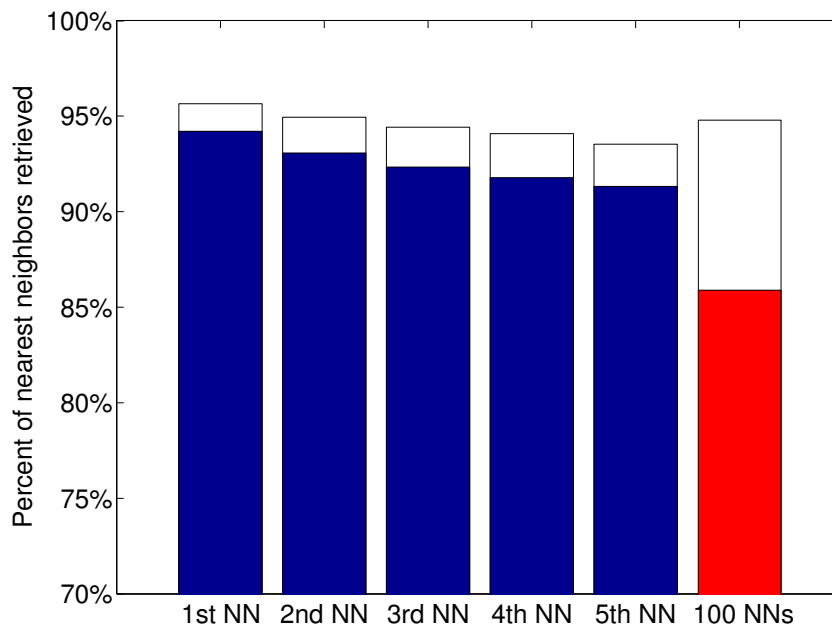


Figure 5.5: The retrieval rate of the proposed approximate nearest neighbors search method. The first five columns show retrieval rate for the first to fifth nearest neighbor respectively, while the last column shows average retrieval rate for the 100 nearest neighbors. The solid bars indicate percentage of nearest neighbors retrieved regardless whether they are meaningful or not, while the rectangles above the solid bars indicate the retrieval rate when also meaningfulness of nearest neighbors is considered. Please note that the vertical scale starts at 70%.

### 5.3 Evaluation of the novel user interface concept

The overall performance of the system presented in this thesis was evaluated within the scope of an user acceptance study [25] conducted on a sample of 16 user and with a real-world scenario. The study took place in the city of Graz, Austria, in June 2008. Each user was first given a brief explanation of the system, the setting, and what we expect him to do. After that the user was free to test the system by himself, free and independently within a predefined area. We expected the user to walk around and be curious about prominent objects in his surroundings, mostly buildings, but also shops, restaurants, cafés, ice cream parlors, bookshops, pharmacies, logos, banners, monuments, et c. The users shot 73 query images altogether (see Figure 5.6 for examples of query images)<sup>2</sup>.

In order to evaluate the performance of our system, we have measured the geographic position of each query image and counted the number of hyperlinks a user would expect to find on a particular query image. The geographic position of the query images was measured using a high-resolution aerial image with a ground sampling distance of ten centimeters.

<sup>2</sup>Also the query images are available online at <http://vicos.fri.uni-lj.si/GUIS107/>

<b>~10 m</b>	<b>~20 m</b>	<b>~30 m</b>	<b>~50 m</b>	<b>~100 m or more</b>	<b>Not positioned</b>
38	9	5	0	7	14

Table 5.2: Accuracy of image based localization. Out of 73 query images, 52 images were positioned with accuracy comparable to GPS, while the remaining 21 images were positioned incorrectly or not at all.

Due to the lack of distinctive markings on the ground, the accuracy of this method is only about five meters, but this is still better than the accuracy of the GPS receiver [56] that is built into the camera phone used in the study. We have counted the number of hyperlinks a user would expect to find on a particular query image based on the number of prominent objects in the image. Because the definition of a prominent object is rather vague and user dependent, the number of hyperlinks counted in such a way provides only an approximate measure of the success of our implementation of the “Hyperlinking Reality via Camera Phones” concept. Each of the 73 query images included at least one hyperlink, but none more than six. The average number of hyperlinks was 2.4 with a standard deviation of 1.2.

Our system automatically annotated 51 out of 73 query images with at least one hyperlink (see Figure 5.6 for examples of query images with annotated hyperlinks). The maximum number of hyperlinks per query image was six and the number of hyperlinks per query image was never larger than the number of manually counted hyperlinks. The average number of automatically annotated hyperlinks was 1.8 with standard deviation of 1.1. We manually checked all the automatically annotated hyperlinks and we did not find any incorrect annotations. There were 22 query images without any hyperlink annotated. Such an outcome would clearly be a disappointment to the user expecting access to information about objects in his surroundings. On close inspection, we found out that a reason behind 14 out of 22 failures was unsuccessful localization (see below). Namely, the transfer of relevant information (i.e. hyperlinks) from the reference panoramas to the query image is the last step of our method and therefore it fails whenever the preceding steps fail.

The success of image based localization was evaluated by comparing the calculated and measured geographic position from where a query image was shot. The results, presented in Table 5.2, show that out of 73 query images, 52 images were positioned with accuracy comparable to GPS [56] with median localization accuracy of 13.5 meters, while the remaining 21 images were positioned incorrectly or not at all. These results further support the claim of [56] that image-based localization can augment GPS, or in some circumstances where GPS performs poorly, even replace it.

The user acceptance study [25] has shown, that users reacted positively on the application and were highly motivated to take advantage of the intuitive interface, with some important remarks regarding technical features (response time, reliability), information visualization and future applications of the technology.

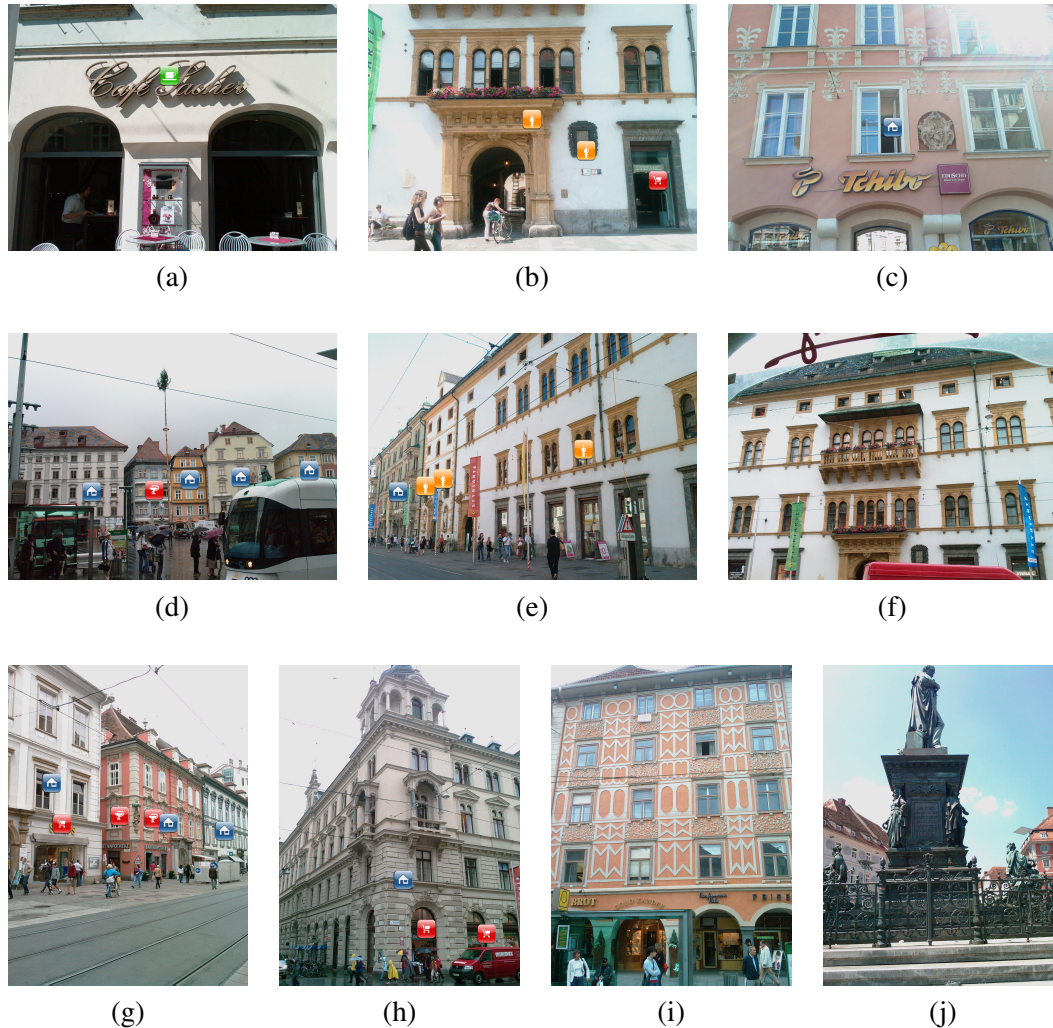


Figure 5.6: Query images with annotated hyperlinks. For query images (a), (b), (d), (e), (g), and (h) our system has automatically annotated all the hyperlinks that a user would anticipate. In the query image (c) only the building itself was hyperlinked while the “Tschibo” logo and the monument of St. Mary were not hyperlinked. Please notice, that this image was shot against the sun, a practice not uncommon among our test users. Our method has failed to annotate with hyperlinks query images (f), (i), and (j). The images (f) and (i) failed due to the dominance of repetitive structures, while the monument in image (j) has insufficient number of features detected on it and it is also not planar enough for our method to succeed.

### 5.3.1 System setup

The system was implemented as a three-tier architecture (see Figure 1.2). The client ran on a HTC Touch Cruise camera phone and it was responsible for capturing an image, sending it to the server, and displaying the results to the user using the web browser. The task of the application server was to receive the image, write the focal length into its EXIF header, send it for processing to the computer vision server, and visualize the results in the form of a web page. The focal length of the HTC was fixed and provided by the phone's manufacturer. The computer vision server took as input an image with focal length written in its EXIF header and returned the result in XML form.

The HTC Touch Cruise is a UMTS mobile phone with a Microsoft Windows Mobile operating system and a touch screen. In addition, it has a three megapixel camera of good quality with autofocus and a built-in GPS receiver. Because the phone supports high-speed downlink packet access (HSDPA), the transmission of results from the server to the mobile client took less than one second. Unfortunately, this phone does not support high-speed uplink packet access (HSUPA) and therefore the transmission of a three mega-pixel image with average size of half a megabyte is the most time consuming step in our system taking approximately 25 seconds. With a broader adoption of HSUPA, the transmission of the query image to the server should take only a couple of seconds.

The application and computer vision servers ran on a dual quad core Intel Xeon processor with 8GB of memory. On average, the query image was processed in 19.6 seconds with a standard deviation of 8.6 seconds. Because all components of our recognition pipeline are highly parallelizable, the processing time could be substantially reduced with larger number of processing cores.



## Chapter 6

# Summary and Conclusions

We have presented a novel user interface concept for camera phones based on the state-of-the-art computer vision technology, called “Hyperlinking Reality via Camera Phones”. The presented concept provides a solution to one of the main challenges facing mobile user interfaces, that is, the problem of selection and visualization of actions that are relevant to the user in his current context. Instead of typing keywords on a small and inconvenient keypad of a mobile device, a user of our system just snaps a photo of his surroundings and objects on the image become hyperlinks to information.

Our method commences by matching the query image to the reference panoramas depicting the same scene that were collected and annotated with information beforehand. Once the query image is related to the reference panoramas, we transfer the relevant information from the reference panoramas to the query image. By visualizing the information on the query image and displaying it on the camera phone’s (multi-)touch screen, the query image augmented with hyperlinks allows the user intuitive access to information. In addition, we provide the user with information about his position and orientation, thus augmenting GPS.

The novel user interface concept presented in this thesis is enabled by a novel high-dimensional feature matching method based on the concept of meaningful nearest neighbors and a novel approximate nearest neighbors search method that provides a ten-fold speed-up over an exhaustive search even for high dimensional spaces while retaining excellent approximation to an exact nearest neighbors search. Our novel high-dimensional feature matching method improves effectiveness of image matching methods which are based on local invariant features, while the speed-up provided by the novel approximate nearest neighbors search method, brings our system closer to interactivity in real-time. We consider a nearest neighbor as meaningful if it is sufficiently close to a query feature such that it is an outlier to a background feature distribution.

The presented mobile user interface concept requires a data set of reference panoramas that are collected and annotated with information beforehand. The Graz Urban Image data Set consists of 107 reference panoramas shot from accurately measured positions, while the camera orientations were acquired automatically using computer vision techniques followed by manual verification. On each reference panorama a few dozen buildings, logos,



banners, monuments, and other objects of interest to the user were annotated using the hyperlinks annotation tool. In the near future, we envision collecting the data sets of reference panoramas for many more cities in an industrial fashion, by augmenting the current process of mobile mapping imagery acquisition of digital cartography vendor companies, such as TeleAtlas. Alternatively, we could also use street-level imagery, such as provided by Google [69] or Microsoft.

The performance of the system presented in this thesis was evaluated using 73 query images acquired within the scope of a study that validated acceptance of the system by users [25]. The results of the evaluation show that our method successfully annotated with at least one hyperlink 70% of the query images, with two thirds of the failures being caused by unsuccessful localization. The image-based localization positioned 71% of the query images with accuracy comparable to GPS, i.e., with a median localization accuracy of 13.5 meters, further supporting the claim of [56] that image-based localization can augment GPS, or in some circumstances where GPS performs poorly, even replace it.

The system presented in this thesis is not a proper augmented reality system yet. A proper augmented reality system [2, 23] should (1) combine real and virtual, (2) be interactive in real-time, and (3) be registered in 3D. While our system does combine real and virtual, and it is registered in 3D, it is not interactive in real-time. In our system, processing of a typical image is done on a server computer and it requires 19 seconds on average using two four-core Intel processors, while a proper interactivity would require at least five frames per second and processing done on the camera phone itself in order to avoid lags due to data communication latency.

Users of our system would benefit greatly if it was interactive in real-time. One of the concerns of the users participating in the user acceptance study [25] was that they do not have feedback to inform them whether or not there exist hyperlinks for a given object. If our system was capable of real-time interactivity, users could look around with a camera phone acting as a looking glass indicating hyperlinks present in their environment. By pressing a button, the user could first freeze the current view and then explore available hyperlinks. As was shown in [23, page 53], users prefer to freeze the current view in order to assume a more natural pose for interaction with the system. In addition, a description of a hyperlink could be shown, if the user keeps an icon indicating the hyperlink in the center of the view for an extended time.

A natural way for speeding-up our system and bringing it closer to interactive frame rates would be to couple it with tracking using a video stream (e.g., [15]), accelerometers, compass, and GPS, while our system would provide for initialization, drift prevention using periodic re-initialization, and fine-grain registration. A major obstacle that should be addressed before coupling our system with tracking is that typical video frame image quality and resolution in today's camera phones is insufficient for a good performance of our system.

Finally, we would like to note, that the major advantage of our approach compared to other approaches to augmented reality is simplicity of content creation since there is no need for the content creator to have detailed knowledge of a 3D environment [23, page 59]. In our system, a hyperlink can be attached to an object by simply selecting features detected on the object.

There are several important challenges that we did not investigate yet but would be important for a commercial application of our technology. Some of these challenges are how to integrate multiple and overlapping users' updates, how to prevent malicious updates, and when to allow removal of objects from the objective maps. In addition, the current implementation of the hyperlinks annotation tools requires a desktop computer. How to implement such a tool also on a camera phone is more of an engineering problem than a scientific challenge.



## Appendix A

# Povzetek magistrske naloge v slovenskem jeziku

### A.1 Uvod

Že leta 2008 je bilo po celem svetu v uporabi več kot tri milijarde prenosnih telefonov [31]. Čeprav moderni prenosni telefoni omogočajo zahtevno procesiranje in hitre podatkovne komunikacije, se še vedno uporabljajo predvsem za zvočno komunikacijo med ljudmi. Ena glavnih ovir za bolj razširjeno uporabo prenosnih telefonov za dostop do Interneta in drugih podatkovnih omrežij so neustrezni uporabniški vmesniki prenosnih telefonov, saj tradicionalne vhodno/izhodne enote kot so zaslon, miška in tipkovnica niso najbolj primerne za prenosne naprave, medtem ko nekaterih drugih možnosti, kot je upravljanje z glasom, uporabniki niso splošno sprejeli [33].

Danes najbolj priljubljena zasnova uporabniškega vmesnika za dostop do informacij na prenosnih napravah je odločanje med omejenim naborom možnosti, ki jih lahko uporabnik izbere s pomočjo tipkovnice, igralne palice, oziroma, v zadnjem času, s pomočjo zaslona občutljivega na (večkratni-)dotik [33]. Poglavitni izziv tega pristopa je, katere možnosti predstaviti uporabniki in kako jih prikazati. Zaradi obilice informacij, ki so na voljo, in omejenega poznavanja uporabnikovega konteksta, je izbor možnosti, ki utrezajo uporabniku v njegovem trenutnem kontekstu, zahteven in izzivalen problem, ki je bistven element uspeha uporabniškega vmesnika namenjenega prenosnim napravam.

Zanimivo nadomestilo tradicionalnim uporabniškim vmesnikom na prenosnih napravah so preučevali pri Nokia Research v okviru projekta MARA (Mobile Augmented Reality Applications – Aplikacije prenosne dopolnjene resničnosti) [29]. V okviru tega projekta so prenosni telefon opremljen s pospeškometrom, kompasom in globalnim položajnim sistemom uporabili kot okno v okolje dopolnjene resničnosti v katerem je lahko uporabnik dostopal do informacij tako, da je usmeril fotoaparatus vgrajen v telefon v smer zanimivega predmeta. Dodatne informacije o objektu so bile dostopne s pomočjo hiperpovezave dodane v video tok zajet s fotoaparatom telefona, zaradi česar je ta koncept dobil ime "hiperpovezovanje resničnosti s pomočjo telefona" [20]. Če lahko pripišemo dejanja predmetom, ki obkrožajo uporabnika, potem postanejo vsi predmeti v uporabnikovi okolici možni sprožilci

dejanj, ki jih lahko uporabnik sproži s preprostim fizičnim dejanjem usmerjanja fototelefona v smeri zanimivega predmeta. Še več, slika uporabnikove okolice prikazana na (večkratni) dotik občutljivem zaslonu fototelefona postane naravna interaktivna naprava, ki omogoča intuitiven dostop do informacij. Torej, koncept "hiperpovezovanje resničnosti s pomočjo telefona" reši pglavni problem koncepta uporabniškega vmesnika za prenosne naprave "odločanje med dejanji", to je izbor in prikaz ustreznih dejanj. S splošno razširjenostjo prenosnih naprav opremljenih z globalnim položajnim sistemom in kompasom, je postal pristop k prenosni dopolnjeni resničnosti, predstavljen v projektu MARA, uveljavljen in široko sprejet med uporabniki [11]. Sveža primera uporabe pristopa projekta MARA sta storitvi Layar [32] in Wikitude [71].

Poglavitna slabost pristopa uporabljenega v projektu MARA je odvisnost od natančne informacije o absolutnem položaju in usmeritvi fototelefona. Na primer, srednja natančnost določitve položaja s pomočjo globalnega položajnega sistema je samo deset metrov [56], kar je premalo za natančno dodajanje hiperpovezav v video tok in posledično močno poslabša uporabniško izkušnjo. V kolikor bi uporabili tehnike računalniškega vida za določitev predmeta(-ov) v smer katerih kaže uporabnik, bi lahko udejanili enak koncept uporabniškega vmesnika za prenosne naprave kot v projektu MARA pri čemer ne bi potrebovali podatka o absolutnem položaju in usmeritvi fototelefona, kar bi omogočalo precej bolj natančno povezovanje fizičnega sveta z virtualnim.

V naslednjem razdelku najprej na kratko predstavimo področje uporabe računalniškega vida na prenosnih napravah in obstoječe tehnike računalniškega vida, ki temeljijo na lokalnih invariantnih značilnicah. Nato v razdelku A.3 opišemo glavne prispevke pričujoče naloge, ki so prikaz koncepta uporabniškega vmesnika za prenosne naprave, nova metoda iskanja ujemanja med visoko-dimenzionalnimi značilnicami in nova metoda približnega iskanja najbližjih sosedov. Koncept uporabniškega vmesnika za prenosne naprave "Hiperpovezovanje realnosti s pomočjo fototelefona" potrebuje za delovanje nabor predhodno zajetih referenčnih panoramskih slik na katerih so predmeti označeni in povezani z informacijami, zato v razdelku A.4 predstavimo postopek zajema referenčnih panoramskih slik. V okviru študije sprejemljivosti sistema s strani uporabnikov [25] smo ocenjevali zmogljivost našega sistema po resničnem scenariju uporabe. Rezultate ocenjevanja predstavljamo v razdelku A.5. V razdelku A.6 zaključimo poglavje z razpravo o koristih, ki bi jih prinesla našemu sistemu interaktivnost v realnem času.

## A.2 Pregled področja

Dandanes je na tržišču zelo malo telefonov brez vgrajenega fotoaparata. Ocenjuje se, da je že leta 2008 skupno število teh, tako imenovanih fototelefonov, preseglo številko ene milijarde [39]. Zato se zdi naravno razmišljanje o uporabi kamere kot vhodne naprave za iskanje informacij s pomočjo prenosnih telefonov [52, 72, 70, 14, 61]. Med navedenimi sistemi je najbolj soroden našemu sistem [61, 14]. V nasprotju z našim sistemom se večina obdelave podatkov v sistemu Takacs in sodelavcev [61] vrši na samem prenosnem telefonu in to tako, da strežnik neprestano posreduje prenosnemu telefonu podmnožico značilnic, ki se nahajajo v uporabnikovi okolici. Ob trenutni pasovni širini brezžičnih omrežij, omogoča

obdelava podatkov na prenosnem telefonu krajše odzivne čase, vendar na račun kakovosti rezultatov. Medtem, ko naš sistem ne zagotavlja interaktivnosti v realnem času, sistem Takacs in sodelavcev ni poravnan z okolico uporabnika v 3D, in torej prav tako ni pravi sistem dopolnjene resničnosti (več v razdelku A.5).

Obstoječe metode računalniškega vida še niso dovolj robustne za delovanje v okviru omejitev prenosnih telefonov, zato so novejšje raziskave usmerjene v to, kako jih prilagoditi za delovanje v različnih okoljih in pogojih. Kot najbolj obetajoč pristop se kaže pristop temelječ na lokalnih invariantnih značilnicah [53, 35], ki se uporablja v mnogih aplikacijah računalniškega vida kot so iskanje slik [53, 46, 12], prepoznavanje predmetov [35, 16], ujemanje stereo slik [3, 38, 67], izdelava panoramskih slik [8], lokalizacija na podlagi slik [74] in iskanje v video posnetkih [55]. Po tem pristopu se lokalne invariantne značilnice neodvisno zaznajo v vsaki sliki posebej, nakar se poišče ujemanje med značilnicami iz obeh slik s primerjanjem opisnikov zaznanih značilnic. Značilnice, ki se ujema v obeh slikah, se lahko uporabi kot dokaz prisotnosti določenega predmeta, kot glas za določeno sliko, ali kot morebitno korespondenco za ocenjevanje epipolarne geometrije.

Iskanje ujemanja med zaznanimi značilnicami je bistven korak pristopa, ki temelji na lokalnih invariantnih značilnicah. Ker noben od obstoječih pristopov k iskanju ujemanja, ki smo jih proučili [3, 35, 55, 46, 19], ni ustrezal naši zahtevi, da mora dobra metoda za iskanje ujemanja zagotoviti zadostno število glasov ustrezni referenčni sliki in ustrezno število korespondenc med katerimi je zadosten delež pravih korespondenc. Zaradi tega smo se odločili razviti novo metodo iskanja ujemanja med visoko-dimenzionalnimi značilnicami, ki temelji na konceptu smiselnih najbližjih sosedov in katero predstavljamo v tej nalogi.

Pomembna prednost klasičnega pristopa k iskanju ujemanje med značilnicami pred drugimi pristopi, je njegova preprostost in konceptualna jasnost. S tem, da izrecno izvedemo iskanje najbližjih sosedov, prevedemo problem iskanja ujemanja med značilnicami na problem izbora metrike za merjenje razdalje in problem določitve resnično ujemajočih se značilnic v majhni množici najbližjih sosedov. Medtem, ko za nizko-dimenzionalne prostore obstaja več učinkovitih (to je, z logaritmčno časovno zahtevnostjo) algoritmov za iskanje najbližjih sosedov (glej [10, 24] za pregled področja), za visoko-dimenzionalne prostore ni poznan noben algoritem za točno iskanje najbližjih sosedov, ki bi bil bolj učinkovit od izčrpnega iskanja [35, 7, 26, 13]. Ker tudi noben od približnih algoritmov za iskanje najbližjih sosedov ni zagotavljal ustrezne pohitritve iskanja najbližjih sosedov v visoko-dimenzionalnih prostorih, smo se odločili za razvoj nove metode približnega iskanja najbližjih sosedov, katero predstavljamo v tej nalogi.

### **A.3 Metoda**

Kot smo omenili že v uvodu, je cilj naših raziskav omogočiti izvedbo novega koncepta uporabniškega vmesnika za prenosne naprave. Namesto tipkanja ključnih besed na majhni in nepripravni tipkovnici telefona uporabnik preprosto zajame sliko okolice, nakar se predmetom na sliki avtomatično dodajo hiperpovezave do zanimivih informacij. Naša metoda najprej poišče referenčne panoramske slike, ki prikazuje isti prizor kot uporabnikova slika, in nato preslika na uporabnikovo sliko informacije, ki so bile predhodno označene na ref-

erenčnih panoramskih slikah (glej razdelek A.4 za podrobnejši opis nabora referenčnih panoramskih slik). Dodajanje hiperpovezav do informacij predmetom na uporabnikovi sliki in prikaz tako dopolnjene slike na (večkratni-)dotik občutljivem zaslonu fototelefona omogoča uporabniku preprost dostop do željenih informacij.

Pristop, ki smo ga ubrali, da smo lahko fototelefonom dodali zmožnost "hiperpovezovanja realnosti", temelji na iskanju ujemanja med slikami na podlagi lokalnih invariantnih značilnic. Pri iskanju ujemanja na podlagi lokalnih invariantnih značilnic se uporabnikova slika (oziroma njen del) primerja z referenčnimi slikami (oziroma njihovimi deli) z namenom, da se najde ujemanje med uporabnikovo sliko in podmnožico referenčnih slik, ki prikazujejo isti prizor ali predmet. Določitev razmerij med uporabnikovo in referenčnimi slikami omogoča prenos informacij nazaj iz referenčne slike na uporabnikovo sliko. Na primer, če poznamo položaj in usmeritev fotoaparata v trenutku ko smo posneli referenčne slike in smo določili geometrijska razmerja med uporabnikovo in referenčno sliko, potem lahko določimo položaj in usmeritev fotoaparata v trenutku, ko je uporabnik posnel svojo sliko [22].

Najpogosteje uporabljen način uporabe lokalnih invariantnih značilnic za iskanje ujemanja med slikami je način, ki so ga predlagali v [53]. V skladu s tem načinom najprej na vsaki sliki posebej (1.a) zaznamo in (1.b) opišemo množico lokalnih značilnic, čemur sledi (2.) iskanje podobnih lokalnih struktur v slikah in (3.) zavrnitev napačnih povezav med slikami. Glavna prednost pristopa Schmidove in Mohra oz. na splošno uporabe lokalnih invariantnih značilnic je, da lahko obravnava precejšnje spremembe zornega kota in fotometričnih sprememb med slikami in da je odporen na gnečo in zakrivanje med predmeti na sliki [66].

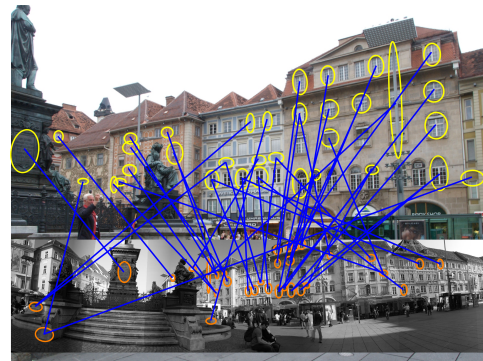
V skladu z načinom, ki sta ga predlagala Schmidova in Mohr, je prvi korak naše metode zaznavanje in opisovanje lokalnih invariantnih značilnic v uporabnikovi sliki. V drugem koraku, za vsako zaznano značilnico v uporabnikovi sliki poiščemo ustrezno značilnico med značilnicami, ki so bile predhodno zaznane v referenčnih panoramskih slikah, in tako poskušamo ugotoviti katere referenčne panoramske slike prikazujejo isti prizor kot uporabnikova slika. V tretjem koraku, poskušamo določiti geometrijske relacije med uporabnikovo sliko in podmnožico referenčnih panoramskih slik, ki prikazujejo isti prizor kot uporabnikova slika. V četrtem koraku, se hiperpovezave, ki so jih na referenčnih panoramskih slikah določili uporabniki oziroma uredniki, prenesejo na uporabnikovo sliko. V zadnjem, to je, petem koraku, se hiperpovezave dodajo predmetom na uporabnikovi sliki in se prikažejo na (večkratni-)dotik občutljivem zaslonu fototelefona. Pregledna shema metode, poglobitve tehnike računalniškega vida, ki smo jih uporabili, in vzorčni primer so prikazane na sliki A.1.

## **A.4 Nabor referenčnih panoramskih slik**

Koncept uporabniškega vmesnika, ki smo ga predstavili v razdelku A.3 potrebuje nabor predhodno zajetih referenčnih panoramskih slik katerim so dodane informacije, ki so zanimive uporabnikom. Zato smo v oktobru 2007 zajeli 1284 referenčnih slik posnetih iz 107 točno določenih položajev znotraj zgodovinskega središča mesta Gradec v Avstriji. Refer-



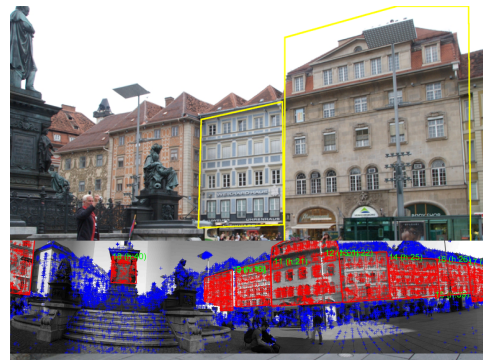
(1) Značilnice MSER [38] opisane z opisnikom SIFT [35]



(2) Iskanja ujemanja med visoko-dimenzionalnimi značilnicami [48]



(3) Ugotavljanje geometrijskih razmerij [22]



(4) Prenos hiperpovezav

Figure A.1: Pregledna shema metode in poglavitne tehnike računalniškega vida, ki smo jih uporabili. Uporabnikova slika je barvna, medtem ko so referenčne panoramske slike črno-bele. V spodnji desni sliki so na referenčni panoramski sliki označeni predmeti in zaznane značilnice (prikazane kot rdeči znaki za seštevanje). Rdeči znaki za seštevanje označujejo značilnice, ki so vključene v opis enega izmed predmetov. Rdeči mnogokotnik označuje obseg predmeta v okviru referenčne panoramske slike, medtem ko številke poleg mnogokotnika enolično označujejo predmet oziroma hiperpovezavo.

enčne slike so bile zajete na način, ki omogoča zajem v velikem obsegu. Nadalje smo s tremi različnimi fotoaparati (Olympus E-1, Canon IXUS I55 in fototelefon Nokia N-90) posneli 184 uporabniških slik, pri čemer smo za vsako uporabniško sliko izmerili točen položaj s katerega je bila posneta. Položaj in usmeritev uporabniških slik so določili poskusni uporabniki, katerim je bilo naročeno naj poslikajo, kar se jim bo zdelo zanimivo. Poleg tega je bilo poskusnim uporabnikom naročeno naj slikajo na mestih, kjer se počutijo izgubljeni in bi jim koristil vodič po mestu. Usmeritev panoramskih slik smo izračunali naknadno s pomočjo tehnik računalniškega vida. Pravilnost izračuna usmeritve panoramskih slik smo še



dodatno ročno preverili.

## A.5 Eksperimentalni rezultati

Uspešnost sistema, ki ga predstavljamo v tej nalogi, je bila preverjena v okviru uporabniške študije [25], ki smo jo izvedli po resničnostnem scenariju na vzorcu šestnajstih uporabnikov. Študija je bila izvedena junija 2008 v mestu Gradec, Avstrija. Vsakemu izmed uporabnikov smo najprej na kratko predstavili sistem in kaj se od njega pričakuje, nakar je lahko samostojno uporabljal naš sistem znotraj v naprej določenega območja. Od uporabnika smo pričakovali, da se sprehaja naokrog in je radoveden kakšne stvari ga obdajajo. Vsega skupaj so uporabniki posneli 73 slik (primeri slik, ki so jih posneli uporabniki, so prikazani na sliki A.2).

Za namen preverjanja uspešnosti našega sistema smo za vsako uporabniško sliko zabeležili točen položaj s katerega je bila posneta in prešteli število hiperpovezav, ki bi jih uporabniki pričakovali na posamezni uporabniški sliki. Vsaka izmed 73 uporabniških slik je vsebovala najmanj eno hiperpovezavo, vendar nobena več kot šest. V povprečju je imela uporabniška slika 2.4 hiperpovezave s standardnim odklonom 1.2.

Naš sistem je na 51 od 73 uporabniških slik avtomatsko dodal vsaj eno hiperpovezavo (primeri slik z dodanimi hiperpovezavami so prikazani na sliki A.2). Največje število hiperpovezav za posamezno uporabniško sliko je bilo šest in število hiperpovezav na posamezni uporabniški nikoli ni presegllo pričakovano število hiperpovezav. V povprečju je imela uporabniška slika 1.8 hiperpovezave s standardnim odklonom 1.1. Ročno smo preverili vse avtomatsko dodane hiperpovezave in ugotovili, da med njimi ni nobene napačne povezave. Za 22 uporabniških slik sistem ni dodal nobene hiperpovezave. Takšen rezultat bi zagotovo razočaral uporabnika, ki pričakuje dostop do informacij o predmetih iz njegove okolice. Podrobnejši pregled razlogov za neuspeh je pokazal, da je razlog za 14 od 22 neuspehov neuspešno določanje geografskega položaja. Prenos ustreznih informacij, to je, hiperpovezav, iz referenčnih panoramskih slik na uporabnikovo sliko je zadnji korak v naši metodi, zato neuspeh v katerem od predhodnih korakov posledično pomeni, da je tudi dodajanje hiperpovezav neuspešno.

Uspešnost določanja položaja na podlagi slike je bila preverjena s primerjavo izmerjenega in izračunanega položaja s katerega je bila določena uporabniška slika posneta. V tabeli A.1 so prikazani rezultati, ki kažejo, da je bil za 52 od 73 uporabniških slik položaj določen s točnostjo, ki je primerljiva z globalnim položajnim sistemom (GPS) [56] s srednjo točnostjo 13.5 metrov, medtem ko je bil položaj preostalih 21 uporabniških slik določen napačno ali neuspešno. Ti rezultati dodatno potrjujejo spoznanje iz [56], da lahko določanje položaja na podlagi slike predstavlja dobro dopolnitev GPSu, ali, v določenih razmerah kjer GPS ne deluje najbolje, celo njegovo nadomestilo.

Uporabniška študija [25] je tudi pokazala, da so uporabniki dobro sprejeli naš sistem in so z veseljem izkoriščali intuitivnost predstavljenega uporabniškega vmesnika za prenosne naprave. Uporabniki so podali nekaj pomembnih pripomb glede tehničnih značilnosti (odzivni čas, zanesljivost delovanja), načina prikaza informacij in prihodnosti uporabe predstavljene tehnologije.

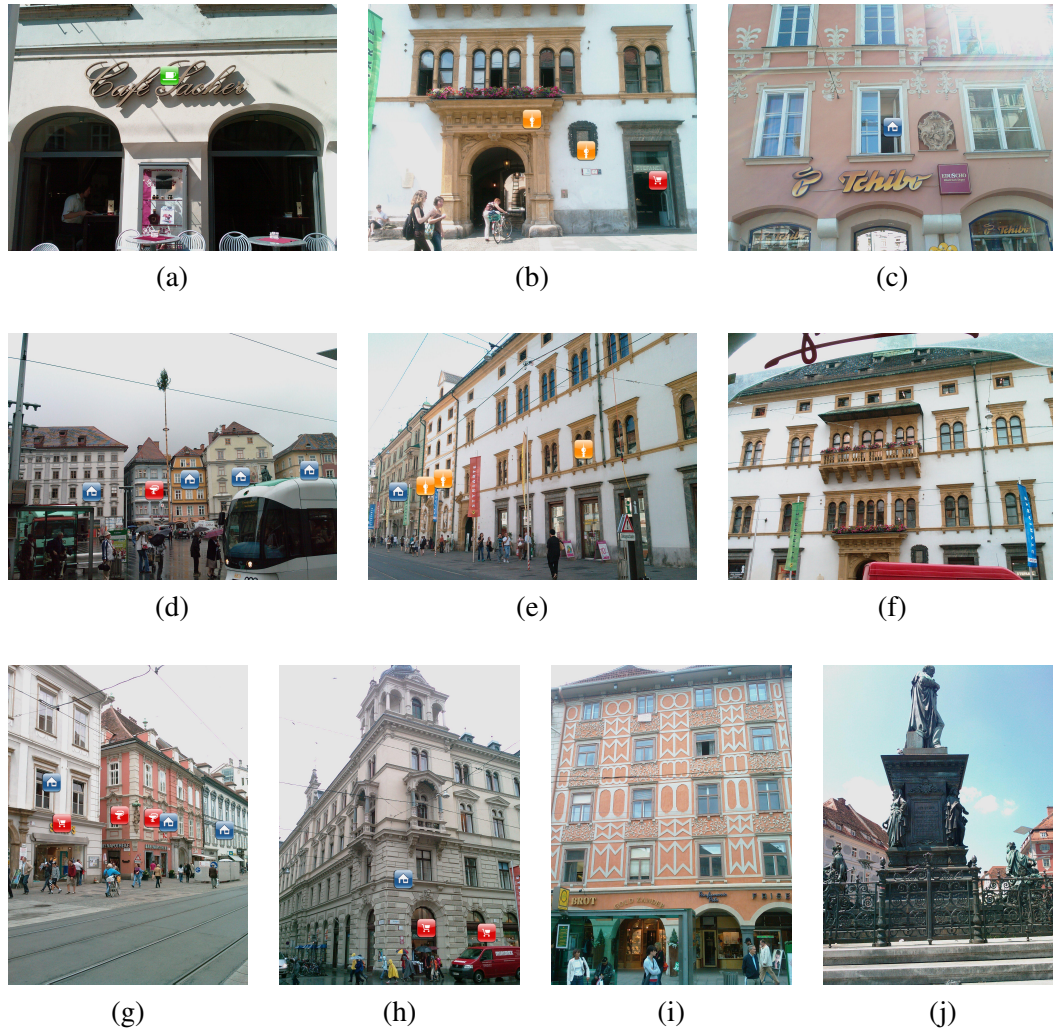


Figure A.2: Uporabniške slike z dodanimi hiperpovezavami. Za uporabniške slike (a), (b), (d), (e), (g) in (h) je naš sistem dodal vse hiperpovezave, ki jih je uporabnik pričakoval. V uporabniški sliki (c) je bila hiperpovezava dodana samo sami zgradbi, medtem ko napisu “Tschibo” in kipu sv. Marije hiperpovezavi nista bili dodani. Naj poudarimo, da je bila slika posneta v smer sonca, kar ni bila tako redka praksa naših poskusnih uporabnikov. Naša metoda je bila neuspešna pri dodajanju hiperpovezav v primeru uporabniških slik (f), (i) in (j). V primeru uporabniških slik (f) in (i) je bil razlog za neuspeh veliko število ponavljajočih se struktur, medtem ko je bilo na uporabniški sliki (j) na spomeniku zaznanih premajhno število značilnic. Poleg tega je spomenik preveč plastičen, da bi bila lahko naša metoda uspešna.

~10 m	~20 m	~30 m	~50 m	~100 m ali več	Ni določeno
38	9	5	0	7	14

Table A.1: Točnost določanja položaja na podlagi slike. Za 52 od 73 uporabniških slik je bil položaj določen s točnostjo, ki je primerljiva z GPS, medtem ko je bil položaj preostalih 21 uporabniških slik določen napačno ali neuspešno.

## A.6 Zaključek

V nalogi smo predstavili koncept uporabniškega vmesnika “Hiperpovezovanje realnosti s pomočjo fototelefona”, ki temelji na napredni tehnologiji računalniškega vida. Predstavljen koncept rešuje enega izmed ključnih izzivov s katerimi se soočajo uporabniški vmesniki namenjeni prenosnim napravam, to je, izbor in prikaz dejanj, ki ustrezajo trenutnemu uporabnikovemu kontekstu. V predstavljenemu konceptu uporabnik namesto tipkanja ključnih besed na majhni in nepripravni tipkovnici telefona preprosto zajame sliko okolice, nakar se predmetom na sliki avtomatično dodajo hiperpovezave do zanimivih informacij. Naša metoda najprej poišče referenčne panoramske slike, ki prikazuje isti prizor kot uporabnikova slika, in nato preslika na uporabnikovo sliko informacije, ki so bile predhodno označene na referenčnih panoramskih slikah. Dodajanje hiperpovezav do informacij predmetom na uporabnikovi sliki in prikaz tako dopolnjene slike na (večkratni) dotik občutljivem zaslonu fototelefona omogoča uporabniku preprost dostop do željenih informacij. Uporabniku lahko dodatno ponudimo še informacijo o njegovem položaju in usmeritvi, kar lahko predstavlja dopolnilo vgrajenemu globalnemu položajnemu sistemu.

Koncept uporabniškega vmesnika, ki je predstavljen v tej nalogi, sta omogočili nova metoda iskanja ujemanja med visoko-dimenzionalnimi značilnicami, ki temelji na konceptu smiselnih najbližjih sosedov, in nova metoda približnega iskanja najbližjih sosedov, ki je desetkrat hitrejša od izčrpnega iskanja celo v visoko-dimenzionalnih prostorih, pri čemer je približek blizu točnemu iskanju najbližjega sosedu. Naša nova metoda iskanja ujemanja med visoko-dimenzionalnimi značilnicami izboljša uspešnost metod za iskanje ujemanja med slikami na podlagi lokalnih invariantnih značilnic, medtem ko nova metoda približnega iskanja najbližjih sosedov približuje predstavljen sistem interaktivnosti v realnem času.

Koncept uporabniškega vmesnika za prenosne naprave “Hiperpovezovanje realnosti s pomočjo fototelefona” potrebuje za delovanje nabor predhodno zajetih referenčnih panoramskih slik na katerih so predmeti označeni in povezani z informacijami. Nabor slik iz Gradca obsega 107 referenčnih panoramskih slik, ki so bile posnete iz natančno izmerjenih položajev, medtem ko je bila usmeritev panoramskih slik izračunana naknadno s pomočjo tehnik računalniškega vida, čemur je sledila ročno preverjanje. Na vsaki referenčni panoramski sliki smo s pomočjo računalniškega programa za dodajanje hiperpovezav označili nekaj deset stavb, napisov, spomenikov in drugih uporabniku zanimivih predmetov. Predstavljamo si, da bo v bližnji prihodnosti možen ekonomičen zajem naborov referenčnih panoramskih slik še za mnogo drugih mest.

Uspešnost sistema, ki ga predstavljamo v tej nalogi, je bila preverjena s pomočjo 73

uporabniških slik, ki so bile zajete v okviru uporabniške študije [25]. Rezultati ocenjevanja so pokazali, da je naša metoda pravilno dopolnila z vsaj eno hiperpovezavo 70% uporabniških slik, pri čemer je bila za dve tretjini napak krivo neuspešno določanje položaja. Določanje položaja na podlagi slike je pravilno določilo položaj za 71% uporabniških slik z srednjo točnostjo 13.5 metrov. Naj povemo še, da je uporabniška študija pokazala dober sprejem predstavljenega sistema s strani uporabnikov.

Sistem, ki smo ga predstavili v tej nalogi še ni pravi sistem dopolnjene resničnosti. Pravi sistem dopolnjene resničnosti [2, 23] bi moral (1) združiti resnično in virtualno, (2) omogočati interaktivnost v realnem času in (3) biti poravnan z okolico uporabnika v 3D. Naš sistem združuje resnično in virtualno in je poravnan z okolico uporabnika v 3D, vendar zaenkrat še ne omogoča interaktivnosti v realnem času. Obdelava uporabniške slike v našem sistemu se vrši na strežniku, za kar je v povprečju potrebno 19 sekund. Prava interaktivnost bi zahtevala obdelavo s hitrostjo vsaj pet slik na sekundo na samem prenosnem telefonu, da se izognemo zakasnitvam povezanim s prenosom podatkov. Uporabnikom našega sistema bi interaktivnost v realnem času prinesla veliko koristi. Ena izmed pripomb uporabnikov, ki so sodelovali v uporabniški študiji [25] je bila, da nimajo povratne informacije za katere predmete v njihovi okolici obstajajo hiperpovezave. Če bi naš sistem omogočal interaktivnost v realnem času, bi lahko uporabniki preprosto pogledali okoli sebe, pri čemer bi fototelefon postal neke vrste povečevalno steklo za dostop do informacij o predmetih v uporabnikovi okolici.



# Bibliography

- [1] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.
- [2] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [3] Adam Baumberg. Reliable feature matching across widely separated views. In *CVPR*, volume 1, pages 774–781, 2000.
- [4] M. J. Bayarri and J. Morales. Bayesian measures of surprise for outlier detection. *Journal of Statistical Planning and Inference*, 111(1–2):3–22, February 2003.
- [5] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR*, pages 1000–1006, 1997.
- [6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(4):509–522, April 2002.
- [7] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, September 2001.
- [8] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74(1):59–73, August 2007.
- [9] Andrea Califano and Rakesh Mohan. Multidimensional indexing for recognizing visual shapes. *IEEE PAMI*, 16(4):373–392, April 1994.
- [10] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.
- [11] Brian X. Chen. If you’re not seeing data, you’re not seeing. <http://www.wired.com/gadgetlab/2009/08/augmented-reality/>.
- [12] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *ICCV*, October 2007.

- [13] Kenneth L. Clarkson. Nearest-neighbor searching and metric space dimensions. In Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk, editors, *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, pages 15–59. MIT Press, 2006.
- [14] Gregory Cuellar, Dean Eckles, and Mirjana Spasojevic. Photos for information: a field study of cameraphone computer vision interactions in tourism. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3243–3248, 2008.
- [15] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, October 2003.
- [16] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 67(2):159–188, April 2006.
- [17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [18] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.
- [19] Kristen Grauman and Trevor Darrell. Approximate correspondences in high dimensions. In *NIPS 19*, pages 505–512, 2007.
- [20] Kate Greene. Hyperlinking reality via phones. *MIT Technology Review*, (11–12), 2006.
- [21] Patrick Haffner. Fast transpose methods for kernel learning on sparse data. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 385–392, 2006.
- [22] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [23] Anders Henrysson. *Bringing Augmented Reality to Mobile Phones*. PhD thesis, Linköping University, 2007.
- [24] Gisli R. Hjaltason and Hanan Samet. Index-Driven Similarity Search in Metric Spaces. *ACM Transactions on Database Systems*, 28(4):517–580, 2003.
- [25] Norman Höller, Arjan Geven, Manfred Tscheligi, Lucas Paletta, Katrin Amlacher, Patrick Luley, and Dušan Omerčević. Exploring the urban environment with a camera phone: Lessons from a user study. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile-HCI)*, September 2009.

- 
- [26] Piotr Indyk. Nearest neighbors in high-dimensional spaces. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 39. CRC Press, 2nd edition, 2004.
- [27] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Symposium on Theory of Computing*, pages 604–613, 1998.
- [28] Herve Jegou, Laurent Amsaleg, Cordelia Schmid, and Patrick Gros. Query-adaptive locality sensitive hashing. In *International Conference on Acoustics, Speech, and Signal Processing*, 2008. to appear.
- [29] Markus Kähäri and David J. Murphy. MARA - Sensor based augmented reality system for mobile imaging. <http://research.nokia.com/research/projects/mara/>, October 2006. Nokia Research Center.
- [30] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *CVPR*, June 2004.
- [31] Phil Kendall. Worldwide cellular user forecasts, 2008-2013. Technical report, Strategy Analytics Inc., June 2008.
- [32] Layar. <http://www.layar.com/>.
- [33] Christian Lindholm, Turkka Keinonen, and Harri Kiljander. *Mobile Usability: How Nokia Changed the Face of the Mobile Phone*. McGraw-Hill Professional, 2003.
- [34] David G. Lowe. Local feature view clustering for 3D object recognition. In *CVPR*, volume 1, pages 682–688, 2001.
- [35] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.
- [36] Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.
- [37] George Marsaglia. Choosing a point from the surface of a sphere. *Annals of Mathematical Statistics*, 43(2):645–646, 1972.
- [38] Jiri Matas, Ondrej Chum, Martin Urban, and Tomas Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, September 2004.
- [39] Neil Mawston. Enabling technologies: CMOS beats CCD in half-billion global camera phone market. Technical report, Strategy Analytics Inc., June 2007.
- [40] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27(10):1615–1630, October 2005.



- [41] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005.
- [42] Pierre Moreels and Pietro Perona. Evaluation of features detectors and descriptors based on 3D objects. In *ICCV*, volume 1, pages 800–807, 2005.
- [43] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. Int. Conference on Computer Vision and Applications*, 2009.
- [44] Sameer A. Nene and Shree K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE PAMI*, 19(9):989–1003, September 1997.
- [45] David Nister. An efficient solution to the five-point relative pose problem. *IEEE PAMI*, 26(6):756–777, 2004.
- [46] David Nistér and Henrik Stewénus. Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, pages 2161–2168, 2006.
- [47] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, December 1997.
- [48] Dušan Omerčević, Ondrej Drbohlav, and Aleš Leonardis. High-Dimensional Feature Matching: Employing the Concept of Meaningful Nearest Neighbors. In *ICCV*, October 2007.
- [49] Dušan Omerčević and Aleš Leonardis. Hyperlinking reality via camera phones. In *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 3515–3516, New York, NY, USA, 2009. ACM.
- [50] Dušan Omerčević and Aleš Leonardis. Hyperlinking reality via camera phones. *Machine Vision and Applications*, pages 1–14, 2010. 10.1007/s00138-010-0285-9.
- [51] Dušan Omerčević, Roland Perko, Alireza Tavakoli Targhi, Jan-Olof Eklundh, and Aleš Leonardis. Vegetation segmentation for boosting performance of MSER feature detector. In *Computer Vision Winter Workshop*, pages 17–23, Moravske Toplice, Slovenia, February 2008.
- [52] Franklin Reynolds. Camera phones: A snapshot of research and applications. *Pervasive Computing, IEEE*, 7(2):16–19, April-June 2008.
- [53] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE PAMI*, 19(5):530–535, May 1997.
- [54] Chanop Silpa-Anan and Richard Hartley. Optimised KD-trees for fast image descriptor matching. In *CVPR*, 2008.

- [55] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, 2003.
- [56] Ulrich Steinhoff, Dušan Omerčević, Roland Perko, Bernt Schiele, and Aleš Leonardis. How computer vision can help in outdoor positioning. In *European Conference on Ambient Intelligence*, volume 4794, pages 124–141. Springer LNCS, November 2007.
- [57] Henrik Stewénius, Chris Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006.
- [58] Henrik Stewénius and David Nistér. Object recognition benchmark. <http://vis.uky.edu/~stewe/ukbench/>.
- [59] Christoph Strecha, Tinne Tuytelaars, and Luc Van Gool. Dense matching of multiple wide-baseline views. In *ICCV*, October 2003.
- [60] Jianbo Su, Ronald Chung, and Liang Jin. Homography-based partitioning of curved surface for stereo correspondence establishment. *Pattern Recognition Letters*, 28(12):1459–1471, 2007.
- [61] Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bimpigiannis, Radek Grzeszczuk, Kari Pulli, and Bernd Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proceeding of the 1st ACM international conference on Multimedia information retrieval - MIR '08*, pages 427–434, October 2008.
- [62] Ben J. Tordoff and David W. Murray. Guided-mlesac: Faster image transform estimation by using matching priors. *IEEE PAMI*, 27(10):1523–1535, 2005.
- [63] Philip H. S. Torr and Andrew Zisserman. Mlesac: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.
- [64] Philip H.S. Torr and David W. Murray. Outlier detection and motion segmentation. In *Proceedings of SPIE*, September 1993.
- [65] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, 2000.
- [66] Tinne Tuytelaars. A survey on local invariant features, 2006. Tutorial at ECCV2006.
- [67] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *IJCV*, 59(1):61–85, August 2004.
- [68] Etienne Vincent and Robert Laganier. Detecting planar homographies in an image pair. In *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, pages 182–187, 2001.

- [69] Luc Vincent. Taking online maps down to street level. *Computer*, 40(12), 2007.
- [70] Jingtao Wang, Shumin Zhai, and John Canny. Camera phone based motion sensing: interaction techniques, applications and performance study. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 101–110, 2006.
- [71] Wikitude. <http://www.wikitude.org/>.
- [72] Tom Yeh, Konrad Tollmar, and Trevor Darrell. Searching the web with mobile images for location recognition. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 76–81, 2004.
- [73] Wei Zhang and Jana Kosecka. Hierarchical building recognition. *Image and Vision Computing*, 25(5):704–716, 2007.
- [74] Wei Zhang and Jana Kořecká. Image based localization in urban environments. In *International Symposium on 3D Data Processing, Visualization and Transmission*, pages 33–40, 2006.

# Izjava

Izjavljam, da sem magistrsko nalogo izdelal samostojno pod vodstvom mentorja prof. dr. Aleša Leonardisa. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Ljubljana, September 23, 2010

Dušan Omerčević