



Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics
Center for Machine Perception

Image Based Localization

Master thesis

Jan Knopp

Thesis Advisor:

Ing. Tomáš Pajdla, Ph.D.

Prague 2009

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Jan Knopp
Study programme: Electrical Engineering and Information Technology
Specialisation: Cybernetics and Measurement - Artificial Intelligence
Title of Diploma Thesis: Image Based Localization

Guidelines:

1. Make a review of the state of the art, in particular works [1,2,3], about image based localization and search in image databases.
2. Suggest a scene representation for localization based on local invariant descriptors of images and a method for its acquisition and image localization.
3. The method implement and test in experiments with real data.

Bibliography/Sources:


- [1] T. Goedeme, M. Nuttin, T. Tuytelaars, L. Van Gool "Omnidirectional Vision based Topological Navigation", Int. Journal on Computer Vision and Int. Journal on Robotics Research, joint special issue of IJCV and IJRR on vision and robotics, 74(3), pp. 219-236, 2007.
- [2] Chum, O. , Philbin, J. , Sivic, J. , Isard, M. and Zisserman, A. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil (2007)
- [3] Integrating Recognition and Reconstruction for Cognitive Traffic Scene Analysis from a Moving Vehicle B. Leibe, N.Cornelis, K. Cornelis, L. Van Gool. in DAGM'06 Annual Pattern Recognition Symposium, Berlin, Germany, Sept. 2006, LNCS, Vol. 4174, pp. 192--201, Springer, 2006

Diploma Thesis Supervisor: Ing. Tomáš Pajdla, Ph.D.

Valid until: the end of the winter semester of academic year 2009/2010


prof. Ing. Vladimír Mařík, CSc.
Head of Department




doc. Ing. Boris Šimák, CSc.
Head

Prague, September 3, 2008

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Jan Knopp
Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM2 – Umělá inteligence
Název tématu: Lokalizace na základě obrazu

Pokyny pro vypracování:


1. Nastudujte literaturu [1,2,3] o lokalizaci pozorovatele z obrazu a vyhledávání v obrazových databázích.
2. Navrhněte reprezentaci scény pro lokalizaci, která bude založena na lokálních invariantních deskriptorech. Cílem je reprezentovat velké scény. Navrhněte techniku získání reprezentace scény a vyhledávání polohy pozorovatele.
3. Techniku implementujte a proveďte experimenty na reálných datech.

Seznam odborné literatury:


- [1] Goedeme, T.; Nuttin, M.; Tuytelaars, T.; Van Gool, L.: Omnidirectional Vision based Topological Navigation. Int. Journal on Computer Vision and Int. Journal on Robotics Research, joint special issue of IJCV and IJRR on vision and robotics, 74(3), pp. 219-236, 2007.
- [2] Chum, O. , Philbin, J. , Sivic, J. , Isard, M. and Zisserman, A. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil (2007)
- [3] Integrating Recognition and Reconstruction for Cognitive Traffic Scene Analysis from a Moving Vehicle B. Leibe, N.Cornelis, K. Cornelis, L. Van Gool. in DAGM'06 Annual Pattern Recognition Symposium, Berlin, Germany, Sept. 2006, LNCS, Vol. 4174, pp. 192--201, Springer, 2006

Vedoucí diplomové práce: Ing. Tomáš Pajdla, Ph.D.

Platnost zadání: do konce zimního semestru 2009/2010


prof. Ing. Vladimír Mařík, CSc.
vedoucí katedry




doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 3. 9. 2008

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 19.5.2009

..... Jan Kučera
podpis

Abstract

The thesis presents the location recognition method of a query image in a dataset of images labelled with geo-location information. This is a challenging task as the imaged appearance of the query image can be very different from the appearance of images in the database due to changes in viewpoint, lighting, and partial occlusion by other objects. In addition, the query image might be captured at a different time of the day, different season or in a different year altogether.

We employ for this task the bag-of-visual-words approach with large vocabularies and fast spatial matching previously used for object retrieval in large image collections. First, the thesis reviews retrieval approaches and their relation to the localization problem. Second, we discuss a method to create the visual vocabulary using the geolocation of database images. Third, query expansion by a non-geotagged image database (such as Panoramio or Flickr) as a way to enrich the query image and a significant improvement of image enriching on extremely hard query images is presented. Fourth, an approach to improving the localization using the automatic detection and suppression of confusing and non-informative features in the geotagged database was developed. Finally, a localization of an image sequence (video) based on Bayes filtering is presented and an improvement of the localization is shown.

We experimentally evaluated image based localization performance and localization stages benefits on several real city-streets different datasets of almost 20K images. We also present the effect of choosing different detector/descriptor types and vocabulary sizes.

Resumé (Abstract in Czech)

Tato práce představuje metodu pro určení pozice neznámého obrázku pomocí databáze obrázků se známým místem získání. Jedná se o náročný problém, jehož složitost se zvětšuje se zvětšujícím rozdílem mezi dotazovaným obrázkem a obrázcích v databázi, což může být způsobeno rozdílem v úhlu pohledu, osvětlením, či částečným zakrytím jiným objektem. Neznámý obrázek může být také pořízen v jiném čase, nebo ročním období.

Pro řešení toho problému jsme použili bag-of-visual-words přístup využívající velké obrazové slovníky s rychlým párováním, toto řešení bylo již dříve použito pro hledání objektů, či obrázků, ve velkých obrázkových databázích. Práce nejprve osvětluje použití vyhledávací metody pro lokalizaci. Zadruhé je vysvětlen přístup stavění obrazových slovníků z obrázků obsahujících svou lokaci. Následně je prezentováno obohacování neznámých obrázků pomocí obecné, s neznámými pozicemi, databáze obrázků (např. Panoramio nebo Flickr). Práce ukazuje výrazné vylepšení obohacováním na některých extrémně těžkých obrázcích. Vylepšení lokalizace bylo také dosaženo pomocí detekce a následného vymazání “zmatečných” oblastí v obrázkové databázi. Lokalizace sekvence obrázků, tedy videa, založená na Bayesovu filtru je prezentována současně i s vylepšením, které přináší.

Výkonnost lokalizace je experimentálně ověřena na třech rozdílných sadách obsahujících téměř 20K obrázků. Jendou z dalších částí je také ukázka jak je výsledná lokalizace ovlivněna použitím různých detektorů/deskriptorů, nebo i velikostí obrazového slovníku.

Acknowledgments

I am very grateful to Tomáš Pajdla to supervise my work for his advices, amiability and encouragement. The thesis could never arise in the current state without his leadership.

My very thanks must go to Josef Šivic for leading me in image indexing and large-scale image retrieval approaches and for giving me a lot of important inventions and ideas. He significantly affected my work as well.

I would like to thank to number of people who supported me on the research level: Michal Havlena for a general computer vision support, Akihiko Torii for omni vision support and Ondra Chum for geometry verification support. I also want to thank to Kurt Cornelis and Mario Ausseleos who helped me when I encountered with a localization problem.

Also so many thanks to all CMP staff for their friendliness and support.

Importantly, I have to thank to my family for their support and patience.

Contents

1	Introduction	1
1.1	Problem Formulation and Motivation	1
1.2	Applications	2
1.3	Challenges of Image Based Localization	2
1.4	Overview of the Presented Approach	4
1.5	Thesis Structure	5
2	State of the Art	6
2.1	Local Features Detectors and Descriptors	6
2.1.1	D. Lowe Approach	7
2.1.2	H. Bay, T. Tuytelaars and L. van Gool Approach	8
2.1.3	J. Matas and Š. Obdržálek Approach	9
2.2	Large Scale Image Retrieval	10
2.2.1	Text Search Based Method	10
2.2.2	Hash Function Based Methods	11
2.2.3	Approaches using Tree Structures	12
2.3	Landmark Clustering & Photocollections Summarizing	13
2.4	Image Based Localization	14
2.4.1	Localization using Reconstructed Scenes	14
2.4.2	Large Scale Location Recognition	14
2.5	Comparison of Our Work with the State of the Art	15
2.5.1	Cascade for the Location Recognition	15
2.5.2	Visual Vocabulary & Detectors/Descriptors Investigation	16
2.5.3	Visual Vocabulary Construction	16
2.5.4	Suppression of Confusing Regions	16
2.5.5	Location Query Expansion	17
2.5.6	Video Localization	17
2.5.7	Implementation	17

3	Image Datasets	18
3.1	Downloading Google Street-View Images	18
3.2	Prague Omni-Images Database (POI)	18
3.3	Paris Landmarks Image Database (PL)	19
3.4	Paris Islands Image Database (PI)	20
4	Representation of City Images	21
4.1	Feature Extraction	21
4.2	Image Indexing	23
4.2.1	Problem Formulation	23
4.2.2	Overview of the Text Search Inspired Image Retrieval	24
4.2.3	K-Means Clustering to Create a Visual Vocabulary	25
4.2.4	Visual Vocabulary as the Set of Most Informative Words	26
4.3	Detecting and Suppressing Confusing Features	29
4.3.1	Local Confusion Score	29
4.3.2	Suppressing Confusing Features	30
5	Location Recognition	32
5.1	Initial Retrieval of Candidate Locations	33
5.2	Filtering by Spatial Verification	34
5.3	Verification of Top-Ranked Location	36
5.4	Location query expansion using non-geotagged images	37
5.5	Video Localization	38
5.5.1	Bayes Filtering	39
5.5.2	Localization of Upcoming Images	41
6	Experimental Evaluation	42
6.1	Gold Standard Method	42
6.2	Location Recognition	43
6.2.1	Initial Retrieval & Spatial Verification	44
6.2.2	Verification of the Top-Ranked Image	46
6.3	Geo-location Estimation	47
7	Conclusion & Future Work	48

List of Abbreviations

We need to use some mathematical symbols, here are their definitions.

Mathematical symbols:

- M** matrix. Computer vision used matrices: **H** - 3x3 homography matrix, **E** - 3x3 essential matrix, **F** - 3x3 fundamental matrix, **K** - 3x3 intrinsic camera matrix, **R** - 3x3 rotation matrix.
- v** n-dimensional vector.
- $i \in I$ a set I and i is the element of I .
- $I_a \subset I$ a set I and I_a is the subset of I .
- $P(x)$ probability of x achieving. $P(x|y)$ is condition probability, that means probability of x achieving when the y is achieved.

Chapter 1

Introduction

1.1 Problem Formulation and Motivation

The goal is to localize the query image of a particular street or building facade, something like: “Tell me where my photo was obtained?” We focus on the representing of cities by collections of geo-tagged images. The task is then to find a corresponding image from a geo-tagged image database, see Figure 1.1(b), depicting the same location as the query image, see Figure 1.1(a).

To obtain the geo-tagged image database we turn into the feature of Google Maps [map] called Google Street-view [vie] which provides panoramic 180x360 deg views for many streets. We implemented downloading of images from Google street-view to obtain complete geotagged street image database of a part of Paris which was used in this work for location recognition.



Figure 1.1: **Formulation of image based localization problem.** We downloaded Google street view images (b) with known geo locations (c). Given a query image (a) at unknown position, the goal is to find a corresponding database image (b) giving its location (c).

In addition, there are large databases of visual data such as Flickr [Fli] with 3G images in total at the end of year 2008 (they claim that 1G of images has been uploaded last year) and Google's Panoramio [Pan]. The hugest collection of images is managed at FaceBook [Fac]. They claim being uploading 28M photos every day but significant amount of images come from parties, faces, and other geo non-specific photos. We used public Flickr and Panoramio image databases to improve the location recognition.

1.2 Applications

The solution of image based localization allows the following applications:

- **Position recognition.** The goal is to say where has the query image been obtained. Such functionality might become a basis for location recognition from images taken by a mobile phone.
- **Geotag information correction.** Flickr, Panoramio etc. collected a huge number of images with geographical information. These geo tags, which are set manually, are often (20%) incorrect. We can correct this information by our technique.
- **Image clustering** is the answer to the question "Which images are similar in my database?" This clustering is a necessary part of automatic 3D reconstruction algorithms. This work was successfully used in Havlena *et al.* [HTKP09] for images similarity measuring as pre-processing of 3D reconstruction from image triplets.

1.3 Challenges of Image Based Localization

Although querying for the most similar image (image mining or image retrieval) from image collections is a hot computer vision field, many problems haven't been solved yet. In this thesis, we focus on the following challenges:

- **Extremel daytime changes.** Standard matching techniques are failing to find similarities between day images and night images, see Figure 1.2(a) for examples.
- **Partial occlusion by other objects.** In many images the location discriminative objects are occluded by trees, people, posters and so on, see Figure 1.2(b). These confusing objects make the problem more challenging.



(a) Very different illuminations.



(b) Viewpoint changes and partial occlusions by other objects.



(c) Many of nondiscriminative objects in the database.

Figure 1.2: **Image based localization challenges.**

- **Confusing objects.** When the image database is constructed under realistic conditions (as in this work), it will contain confusing objects such as trees and cars. These confusers are everywhere and significantly decrease localization performance because of their similarity with many other similar objects. Examples are shown in Figure 1.2(c).
- **Fast querying in large image databases.** This is an important computer science problem. To search in a database of millions items, sublinear running time is necessary.

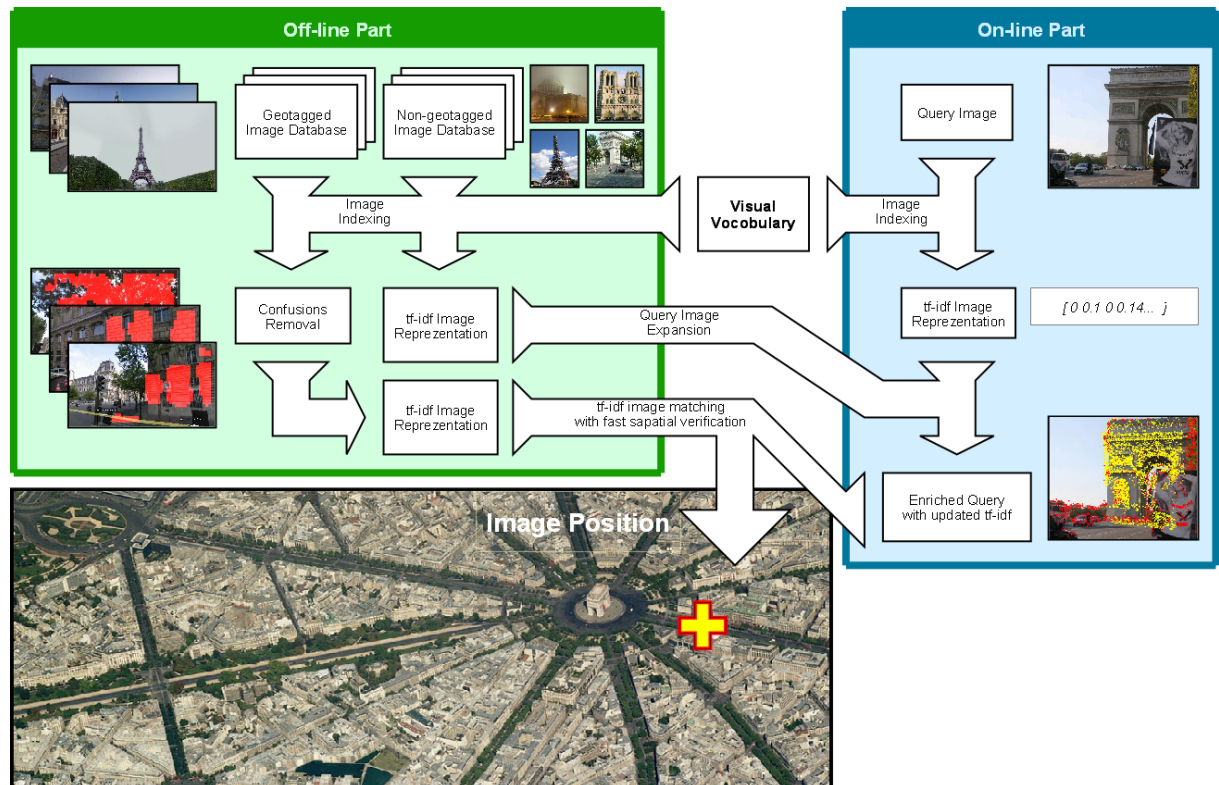


Figure 1.3: **Structure of our localization recognition method.** Figure shows database preprocessing (off-line) as well as the on-line localization of the query image.

1.4 Overview of the Presented Approach

Here, we discuss the structure of the presented localization method which is illustrated in Figure 1.3. The proposed system could be separated into two parts: (i) off-line preprocessing of database of images and (ii) on-line recognizing the location of the query image.

Firstly, we assume to have a visual vocabulary representing possible visual features. Each feature is a vector describing a small image area as corners, circles etc. The point of using the features is that the similar feature vectors describe similar image segments.

In the off-line part, geo tagged and non-geo tagged images are described by features. Each feature is then quantized by replacing it by the most similar feature from the visual vocabulary. An image is represented as a weighted bag-of-visual-words model computed from quantized features. This process is called a quantisation. With this knowledge, we can easily and quickly compute image similarity. It also allows to detect and suppress confusing features which are not informative for localization.

The goal of the on-line part is to retrieve the most similar geo tagged image containing

the visual overlap with the query image. To achieve this goal, we compute the query image features and quantise them using the visual vocabulary. After that, we aim at localization improvement by turning to a collection of non-geotagged images and the query image is expanded by using different viewpoints or different daytime images. This enriched query image is then matched to geo-tagged image database with suppressed confusing features. This matching could be separated into three stages: (i) Each database image is valued by the bag-of-visual-words image similarity with the query image. We called it **the intial retrieval process**. (ii) Then, the first n most similar images are verified by estimating matching local image geometry from the query image to the database image. It is called **the spatial verification** and it results in image resorting by the number of matching features. (iii) In the final stage, more precise verification between the top-ranked database image and the query is done. As this algorithm is more computationally expensive we used it only for the top-ranked image.

As a result, there are two possibilities. First, the top-ranked image was verified and then we know that it contains a visual overlap with the database image. Therefore, the query image was obtained at a very similar location as the top-ranked database image. Secondly, the query image was not verified and thus was not found in the database.

1.5 Thesis Structure

The thesis is organized as follows. The text continues in Chapter 2 with an overview of the state of the art of the localization methods, which include image feature detection/description, efficient searching in huge collections of images, image clustering etc. Comparison of our work with the state of the art is presented as well. Next, Chapter 3 describes three different collections of images which were used for localization experimets. Chapter 4 briefly presents the localization and describes our representation of images as well as the detection and suppression of confusing features. In Chapter 5, the localization method is described in detail. In addition, we present query expansion using non-geotagged images and localization of the video input. Chapter 6 contains experimental evaluation and demonstrates the benefits and limitations of the presented approach. In final Chapter 7, we summarize results and discuss further work.

Chapter 2

State of the Art

In this chapter we will discuss previous work. Since localization is a complicated problem touching on a variety of computer vision works, we will review several topics.

Firstly, Section 2.1 describes methods of extracting local features from the image. Then, in Section 2.2, we present the state of the art of large scale image and object retrieval. Approaches using tree structures, hashing algorithms and systems inspired by text search engines are reviewed. Thirdly, Section 2.3 reviews results in image clustering, image database summarizing and 3D modeling, which are also related to our work. In Section 2.4, several very important approaches concerning image based localization are presented. Finally, the differences between our work and the state of the art is given in Section 2.5.

2.1 Local Features Detectors and Descriptors

More than twenty years ago, Moravec [Mor83] formulated the concept of the first corner-like feature point detector. He proposed a method using small square shifting windows. Corner points were defined as points with a large intensity variation in selected directions.

Most widely used detector developed by Harris and Stephens [HS88]. They proposed keypoints detection using eigenvalues of a second moment matrix. The biggest problem of Harris corner detector is the lack of scale-invariance. After that, Lindberg [Lin98] presented Harris detector with automatic scale selection. Importantly, Mikolajczyk and Schmid [MS01, MS02] developed an approach based on an affine generalisation of the standard Harris detector. They refined Lindberg's work to create a scale-invariant feature detector called Harris-Laplace and Hessian-Laplace using convolutions with variable-shaped Gaussians.

Next, we will discuss three main modern approaches in detail.



Figure 2.1: **Example of detected feature points and maximally stable regions.**

2.1.1 D. Lowe Approach

David Lowe [Low99] aimed at speeding up the detection by using an approximation of the Laplacian-of-Gaussian (LoG). This solution is invariant to scale, rotation and translation changes. The algorithm was named SIFT (Scale Invariant Feature Transform) and it was one of the most used computer vision algorithm.

Detector [Low99] was built on searching the extremes in the space of image Difference-of-Gaussian (DoG) filter response. First, scale-space $L(x, y, \sigma)$ of an image $I(x, y)$ is produced by the convolution of the variable-scale Gaussian $G(x, y, \sigma)$ with an image $I(x, y)$,

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y). \quad (2.1)$$

Then, Difference-of-Gaussian $D(x, y, \sigma)$ can be computed from the difference of two scales separated by a constant factor k ,

$$D(x, y, \sigma) = \left(G(x, y, k\sigma) - G(x, y, \sigma) \right) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma), \quad (2.2)$$

to produce scale-space images and to find the extremes that represent significant image areas [Low99]. They are represented by their x, y coordinates and the scale σ and named as keypoints.

Descriptor [Low04]. By assigning a consistent orientation to each keypoint based on local image properties, the keypoint descriptor can be constructed as relative to the orientation and therefore invariance to image rotation can be achieved. An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint. Peaks in the orientation histogram correspond to the dominant direction of local gradients.

The local image gradients are measured to correspond the keypoint scale. Every description is then a 128-dimension vector.

Various improvements of the SIFT detector and descripto have been proposed. Ke and Sukthankar [KS04] used PCA of the gradient image. PCA-SIFT has only the 36-dimensional descriptor which is faster for matching but reduces the accuracy [MS04]. The same paper [MS04] presented SIFT also with the 36-dimensional descriptor, called GLOH, that have similar accuracy as the SIFT, but is computationally too expensive. The last refinement of the SIFT, called SURF [BTVG06], is used in our work. We describe it bellow in detail.

2.1.2 H. Bay, T. Tuytelaars and L. van Gool Approach

H. Bay, T. Tuytelaars and L. van Gool focused on how to speed up the SIFT detector and descriptor, see last Section 2.1.1. Their algorithm is called Speed Up Robust Features (SURF) [BTVG06]. The main idea is in using the integrate image,

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i < x} \sum_{j=0}^{j < y} I(i, j), \quad (2.3)$$

which allows to compute an approximation of the second-order derivative of the image in less than 16 memory accesses independently from the position and scale.

Detector is based on the Hessian matrix and it was called the Fast-Hessian Detector. It finds extrema in the scale-space defined by the determinant of the approximation of the Hessian matrix,

$$H_{approx}(x, y, \sigma) = \begin{pmatrix} D_{xx}(x, y, \sigma) & D_{xy}(x, y, \sigma) \\ D_{xy}(x, y, \sigma) & D_{yy}(y, y, \sigma) \end{pmatrix}, \quad (2.4)$$

where D_{xx} is the second partial derivative of the image in the x -direction. D_{xy} and D_{yy} respectively.

Descriptor computes firstly the orientation assignment. This orientation for each feature point is estimated by computing the sum of all responses within a sliding window in a feature point neighbourhood defined by the scale.

Then, in the neighbourhood square region of the feature point, the descriptor vector v for each 4x4 subregion is computed,

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right), \quad (2.5)$$

where d_x and d_y are the first derivatives in each subregion. Directions of d_x and d_y are defined in relation to the computed feature point orientation. This results in a 64-dimensional descriptor vector. The integrate image was used for fast derivative computation.

SURF is almost ten times quicker than SIFT with similar results. This is a significant advantage of SURF but it is only an approximation of SIFT. Therefore, for the matching problem, SURF gives better results for two extreme viewpoint changes or illumination changes. However, we observed that SURF gave the best localization performance in our situation, see Section 4.1.

2.1.3 J. Matas and Š. Obdržálek Approach

This approach uses interest regions instead of interest points. These regions are defined by a closed loop in the image which involves more information than a keypoint with the scale.

Detector is trying to find maximally stable extremal regions (MSER) [MCUP02]. These *extremal regions* are defined as the area including either higher or lower intensity than the area out of the region. MSER are obtained by tresholding the image. We start with the monotonic change of image intensities and changing the intensity step by step. We are looking for compact regions which are maximally stable for the intensity changes. The algorithm for detection of regions is nicely demonstrated by the web animation¹ by Henrik Stewenius.

Descriptor [OM02b] The affine frames are obtained by an affine invariant construction on robustly detected maximally stable extremel regions of data-dependent shape.

In addition, many works as Nister *et al.* [NS06], Sivic *et al.* [SZ03] etc. successfully describe MSER by the popular SIFT [Low04] descriptor.

Also, it is good to noted that MSER is not the only approach how to detect regions. Mikolajczyk *et al.* [MTS⁺05] presented a comparison of several different region detectors (IBR, EBR and Silent Regions). They found that MSER outperforms in the all other region detectors.

¹http://www.vis.uky.edu/~stewe/animations/animation_mser.gif



Figure 2.2: **Object retrieval.** Example of Video Google [SZ03] approach, shows retrieved images from the Ground Hog Day movie with highlighted tie used as the query object.

2.2 Large Scale Image Retrieval

Retrieving an image/object from a huge number of images is a challenging task that has become a very popular computer vision problem in recent years. Figure 2.2 illustrates an example of the object retrieval result.

2.2.1 Text Search Based Method

J. Sivic and A. Zissermann [SZ03] published an image search technique inspired by Google web search. Their approach is based on feature descriptor vector assignment into a vocabulary of k words which allows to describe the image as the tf-idf weighted vector $\mathbf{v} = (t_1 \dots t_j \dots t_k)^T$. After that, searching an image database means a comparison of these tf-idf vectors. The tf-idf image representation is shown in Figure 4.1, where is also described a modification for position recognition.

tf-idf Weighting & Original Video Google Retrieval: The intuition behind the tf-idf weighting is that tf-idf weights words occur more often in a particular image higher and downweights words that appear often in database, because they do not help to discriminative between different images. After that, weighting of word w for an image i is defined as,

$$t_j = \frac{n_{wi}}{n_i} \log \frac{N}{N_{wj}}, \quad (2.6)$$

where n_{wi} is the number of occurrences of the word w in the image i , n_i is the total number of words in the image i , N is the number of database images and N_w is the number of images containing word w_j .

At the retrieval stage, each database image is ranked by the cosine of angle between database image vector \mathbf{v}_d and a query image vector \mathbf{v}_q which is efficiently computed as the dot product,

$$f_d = \mathbf{v}_d \cdot \mathbf{v}_q = \frac{\mathbf{v}_q^T \mathbf{v}_d}{\|\mathbf{v}_q\|_2 \|\mathbf{v}_d\|_2}. \quad (2.7)$$

Query Expansion: O. Chum *et al.* [CPS⁺07] presented a method how to improve image/object search by a common text retrieval method named query expansion. The approach uses the dataset images containing the same objects as the query image. The query image is then expanded by these database images. This method improve the retrieval system significantly as papers [CPS⁺07, PCI⁺08] shows. The success of the method is based on two key elements. First one is that the image database contains images of the same object. Second element is that we enhance the query image after the careful verification filtering of non-relevant images.

The paper presents and compares several different expansion models. They show that the **recursive average query expansion** is a very efficient method where top $m < 50$ verified results returned from the standard search engine are selected and new tf-idf query \mathbf{v}_{avg} is then formed by taking the average of the original query \mathbf{v}_0 tf-idf vector and the m results,

$$\mathbf{v}_{avg} = \frac{1}{m+1} \left(\mathbf{v}_0 + \sum_{i=1}^m \mathbf{v}_i \right). \quad (2.8)$$

In addition, the union of the verified image features are taken to enrich the query image. The expansion is recursively applied to generate queries, where each new iteration uses the last computed \mathbf{v}_{avg} as the new \mathbf{v}_0 for the expansion.

Soft Assignment: James Philb *et al.* [PCI⁺08] did not construct tf-idf by a hard assignment to only one nearest neighbour visual word, but by more. Given the feature descriptor vector, the assignment is computed as the weighted distances to close visual words. Although this method really improves retrieval performance, the improvement is not too significant as the Query Expansion [CPS⁺07].

Vocabulary Tree: David Nisteur *et al.* [NS06] presented the method based on the tree structure. To construct the vocabulary tree the feature descriptors were hierarchically quantized in a tree structure and then the image similarity is computed as the similarity of the way in this tree. Note that the time cost of this approach is the same as standard Video Google retrieval due to the approximate nearest neighbourhood searching using KD-tree or hierarchical k-means.

2.2.2 Hash Function Based Methods

Although locality-sensitive hashing [IM98] has perfect theoretical performance properties, a standard implementation would be still unacceptably slow. Ke *et al.* [KH04] shows

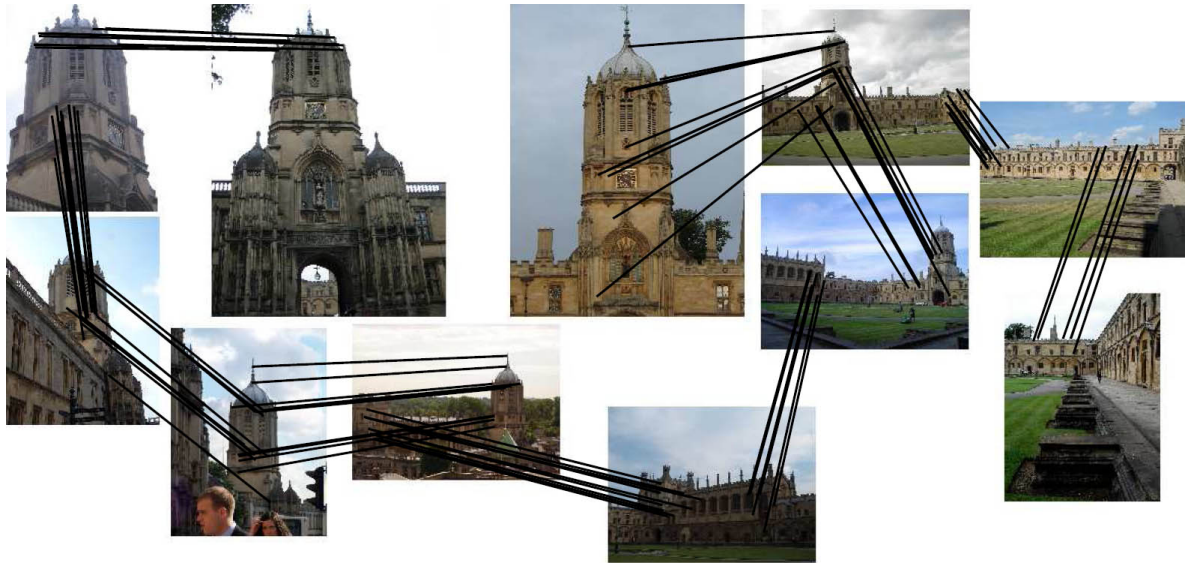


Figure 2.3: **Spatially related cluster.** Part of Chum and Matas [CM08] related cluster discovered from the 100K image database.

LSH improvement of the index access as a way to efficiently query databases containing a million of keypoints. Note that our approach using the text retrieval method works with more than four hundred millions of keypoints.

Chum *et al.* [CPIZ07] show a different way to measure the similarity of images. They represented an image as a set of visual words. This is a weaker representation than a bag-of-visual-words since they do not store the number of occurrences but only whether a word occurred or not. For the estimation of the similarity of two images, the multiple independent min-Hash functions are used. The fraction of the min-Hash functions that assigns an identical value to the two sets gives an unbiased estimate of the similarity of the two images.

2.2.3 Approaches using Tree Structures

Using tree structures is another option for sublinear time image retrieval. Shao *et al.* [SSF⁺03] presented Vantage Point Tree, which recursively organizes the feature vectors into a tree sorted according to their median distance of vantage point.

Re-rendering of image patches to train decision trees to index keypoints was presented by Lepetit *et al.* [LLF05]. This tree was used as the robust classification technique and it was fast enough for real-time application.

Obdrazalek *et al.* [OM02b] aimed at an efficient organisation of the object database, which allows fast recognition response. They proposed a method to learn the tree from

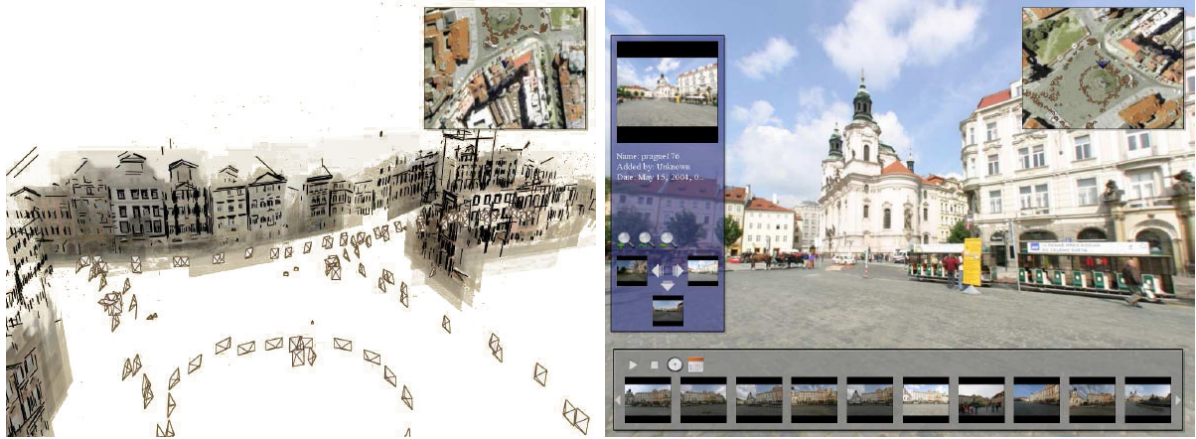


Figure 2.4: **Photo Tourism** [SS06]. From the collection of images (camera view triangles in left image) the scene is reconstructed. It allows to create 3D photo album (right image).

the LAF or SIFT features to significantly improve recognition performance.

2.3 Landmark Clustering & Photocollections Summarizing

James Philbin *et al.* [PCI+08] used bag-of-visual-words model to select image clusters from a image collection containing million of images. This approach is similar to our method as it is also based on the text search based image retrieval system. But they were interest more in the image clustering then in the recognition or retrieval. The paper presents experimets on a one million of imagaes.

Till Quack *et al.* [QLVG08] shows image gathering from internet, clustering retrieved photos to same object or events, classification of clusters into object/event, unsupervised linking with the text-information (Wikipedia) and a verification of those links. However complex the approach looks and the approach is, they use standard image-to-image matching resulting in extremely long computation times.

Previously cited Chum *et al.* [CPIZ07] work used min-Hash algorithm for fast detection of so-called cluster seeds. The seeds were than used as queries to obtain the cluster of images/objects. Figure 2.3 shows a part of a cluster of related images. The paper presents min-Hash for clustering as well as for automatic object labeling or detection of near duplicated images.

One of the most known 3D modeling system Photo Tourism [SS06] includes image retrieval and image clustering. Their approach is based on the Video Google framework [SZ03] pruned to contain 3D consistent matches with the twenty cameras closest to the initial location of a new photograph.

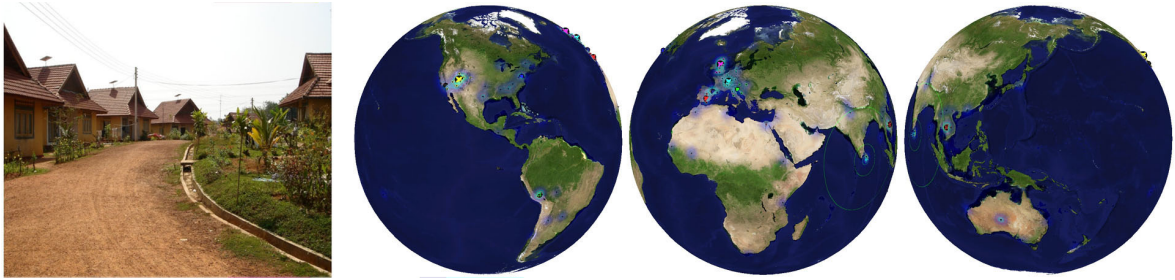


Figure 2.5: **GPS estimation from the single image.** Hays and Efros [HE08] approach. For the query image (left) the geo-location is estimated as the earth probability map (three spheres on the right side).

2.4 Image Based Localization

2.4.1 Localization using Reconstructed Scenes

Nister *et al.* [NS04] build dense 3D models (similar as Structure from Motion problem) out of incoming data based on multi-view linking, which is computationally and memory demanding and also not running well with planar objects.

Se *et al.* [SLL01] and Davison [Dav03] constructed the world only by the special features called *visual landmarks*. Despite of the better memory demanding, it is still not available for huge datasets. In addition, the metric error is accumulating.

Goedeme *et al.* [GNTvG07] presented a complete omni-images using autonomous mobile robot system for in-door as well as outdoor environment. They used world representation as the collection of connected nodes (topological map).

2.4.2 Large Scale Location Recognition

Finally, and the most importantly, most previous work on image based location recognition aimed on a small-scale settings [RR04, SSTvG03]. An exception is the work of Schindler *et al.* [SBS07] who proposed information theoretical criteria for choosing informative features to built vocabulary trees [NS06] for location recognition in a database of 30K images. They aimed at finding the specific (most informative) features occurring in all images of a specific location but rarely occur anywhere outside of this location. They defined for each location $l_i \in L$ and visual word $w_j \in W$ the information gain,

$$I(l_i|w_j) = H(l_i) - H(l_i|w_j). \quad (2.9)$$

They were interested in finding those visual words for each location that maximize information gain $I(l_i|w_j)$ which is equivalent to minimizing $H(l_i|w_j)$ entropy due to constant

$H(l_i)$. In the end, the visual vocabulary was then constructed only from these most informative features. Importantly, paper [SBS07] shows the significant overlap in image database as the key property to determine features which are most informative about each location.

Hays and Efros [HE08] used scene category matching to retrieve images of similar scene from geotagged databes of several million Flickr images. The significant contribution of the cited work lies in ability to work with extremely huge image databases. Hays and Efros downloaded about 20 millions images from which they excluded all photos containing text-labbel such as birthday,cameraphone” and so on. In the end they arrived at a database of almost 6,5M images. They were interested in a world-scale (continents or cities rather than city location) localization, see Figure 2.5 showing the world location probability grid for a given query image. Also, comparing to our work, they used several methods for image matching dominantly based on image color-histograms. Our work shows localization using the matching based on geometry verification.

2.5 Comparison of Our Work with the State of the Art

This section describes the differences between the state of the art (presented in the previous text) and our approach to image based localization. We summarize the main contributions of the thesis and our modifications of the previous work. In addition, we provide brief motivation for our approach.

2.5.1 Cascade for the Location Recognition

The most similar work by Schindler *et al.* [SBS07] uses the standard image retrieval approach [SZ03] to obtain the location of the query image. Our localization model, see Section 1.4, could be presented as a three stages cascade of filters, where each stage filters a defined number of images with the goal to keep at least one correct image. Our first localization stage is identical with Schindler *et al.* [SBS07] approach but we also append the verification stages as well as we used tf-idf weighting compared to using the vocabulary tree approach in the cited paper. Verification is an important part of our method as we will present its necessary for the localizaton of the challenging query images. Additional differences to their work are discussed bellow.

2.5.2 Visual Vocabulary & Detectors/Descriptors Investigation

Several works [MTS⁺05, BTVG06] present the comparison of performance of feature detectors/descriptors on general matching problem, 3D reconstruction and recognition. We show how different local detectors/descriptors types (presented at Section 2.1) affect the localization.

In [SZ03, CPS⁺07, PCI⁺08, NS06, SBS07] parameters of text based object retrieval method were experimentally evaluated on object retrieval approaches. We tested these parameters on the problem of the location recognition with the aim to select these parameters to maximize the localization performance. These results could be found through the whole thesis.

2.5.3 Visual Vocabulary Construction

Visual vocabulary is constructed as the set of k-means cluster centers on all image descriptors. When the number of dataset images increase significantly, it becomes impossible to cluster all data at once. Chum *et al.* [CPS⁺07] or Philbin *et al.* [PCI⁺08] selected a subset of images and then constructed visual vocabulary from this subset. On the other hand, Schindler *et al.* [SBS07] presented the solution for geo-located images based on the selecting features occurring in all images of specific location, but rarely occur anywhere outside of this location. They aimed at selecting the most informative visual words for each location based on informative gain defined in Equation 2.9.

Importantly, visual overlap of close database images is significant assumption for successful use of Equation 2.9. We generalize Schindler *et al.* [SBS07] work to geo-tagged image databases without the extreme visual overlap of images. The method is presented in Section 4.2.4 in detail.

2.5.4 Suppression of Confusing Regions

Locations in city-street image database contain significant amount of features on objects like trees, pavement, sky or water, which are not informative for recognizing a particular location as either (i) they appear frequently throughout the city, or (ii) they cannot be reliably matched by current local features based algorithms. We aim at detecting and suppressing these features. Note that our approach is complementary to Schindler *et al.* [SBS07] where they select the most informative features.

2.5.5 Location Query Expansion

In image/object retrieval the query expansion [CPS⁺07] (briefly described at Section 2.2.1) was shown to significantly improve retrieval performance when multiple database images of same place/object as the quereid one occur in the database. Query image is enhanced using spatially verified images in the database.

In the localization domain, city-street image databases contain only a small number (1-4) of particular location images. We turn to download images from public photo-sharing sites to obtain multiple images of the same places captured at different times or from different viewpoints.

2.5.6 Video Localization

As the related work (see Chapter 2) shows, the majority of image localization works are focused on small-scales problems. They used Bayes rule to process information from the whole video as these images from the video contain more location information.

We present a modification of our large-scale position recognition algorithm using the same approach as small-scale localization approaches which allows to estimate the position from videos.

2.5.7 Implementation

Here, we summarize which parts of the presented work were re-implemented ourselves and, on the other hand, which algorithms were used in the original implementation.

Firstly, we were very interested in understanding previous work. To achieve this goal we implemented the state of the art image retrieval and several localization methods (Schindler *et al.* [SBS07], Sivic *et al.* [SZ03] etc.). As the advantage, it allowed us to compare our method with the state of the art. In addition, our method is built on several elements of previous approaches.

We used in our localization approach the algorithm for approximate nearest neighborhood searching [ML09], local features detectors/descriptors [BTVG06] and the LO-RANSAC [CMK03] including the homography estimation [HZ00].

Chapter **3**

Image Datasets

In this chapter we describe three different datasets which were used for experiments throughout the whole thesis. Several datasets were downloaded during the evolution of our downloading script as the improvement of the script allowed to download better datasets. In the following chapters, we present that we are able to localize hard images in real world datasets.

The script for downloading city-street images is described in Section 3.1. Each dataset is then presented in the following sections, one section for one dataset.

3.1 Downloading Google Street-View Images

We firstly presents our method to obtain city-street images. To obtain the collection of geo-tagged images we used public google street-view API [vA] allowing to figure images specified by the GPS coordinates and the camera angles.

Full-automatic downloading script could by separated into several parts. (i) Goggle Maps was used to obtain GPS coordinates from a specified area using an automatic clicking algorithm; (ii) these coordinates generate queries for the Google Street View engine to get the corresponding street image in a flash web page; (iii) after that, image is saved based on a print-screen like method. As a result, each downloaded image contains the location information (GPS) as well as the camera view direction.

3.2 Prague Omni-Images Database (POI)

First image dataset was used from the reconstruction problem solved in CMP. This is the only dataset where the Google Street View downloading script wasn't used to obtain city images. Therefore, we manually obtained 13K Prague omni-images [SP02] during



Figure 3.1: **Examples of Prague dataset images.** Figure shows database omni images (top) and the slected Flickr query images (bottom).

the walk from Old Town Square to the Prague Castle. After that, every tenth image was selected to create the image database of about 1,300 images.

First 30 query images were selected from downloaded Flickr images using the “Prague” keyword to have visual overlap with some images in the training set. In addition, we photographed another 433 omnidirectional query images. These omni-images were also taken to have the visual overlap.

Only a portion of query images (11 omnidirectional and 14 perspective) were manually assigned to datasets images to create a ground truth.

An example of images from this image dataset is shown in Figure 3.1.

3.3 Paris Landmarks Image Database (PL)

Second image collection was crawled from the Google street-view image database. As it was automatically created by the downloading algorithm (see Section 3.1), it does not cover the area uniformly. Dataset consist of about 19,000 images, which cover several Paris streets with many landmarks as Moulin Rouge, Louvre, Arc de Triomphe, Panthéon, Bastille etc.

Separately, we downloaded a collection of Paris Flickr images and we then manually selected a group of 50 images with visual overlap with our database to create 50 positive queries. Another 50 negative queries were constructed by taking images without any visual overlap.

Huge advantage of this dataset is that we have manually labeled ground truth, which was step-by-step created for the 50 queries. Figure 3.2 shows image examples.

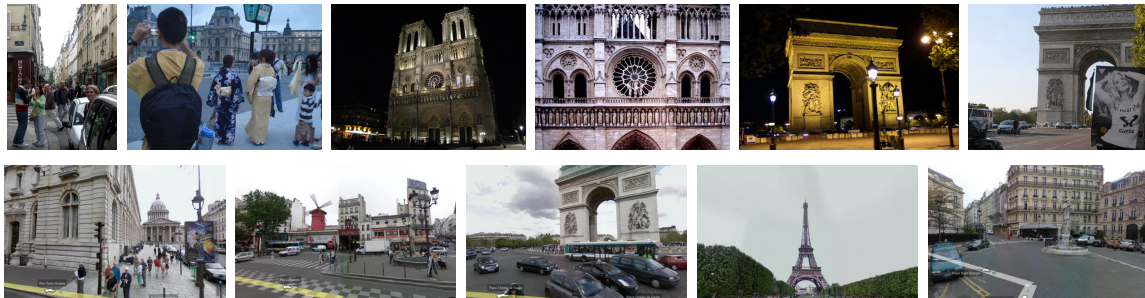


Figure 3.2: **Examples of Paris Landmarks dataset.** Figure shows the selected query images (top row) and database images (bottom row).

3.4 Paris Islands Image Database (PI)

This image collection is the last download dataset. It was created as the collection of all Google street-view images crawled from the selected Paris area covering about 1.7x0.5 kilometers. Dataset consists of about 17,000 images. In addition, we used the location and keyword search to Panoramio photo-sharing site to receive public non-geotagged photos downloaded from roughly same area as was covered by the geotagged images. Majority of important experiments was evaluated on this dataset.

200 non-geotagged images were randomly chosen as the queries to measure the performance of our location recognition algorithm. Query images could be divided into five groups: (i) challenging queries which in principle can be localized but may either have very small visual overlap or the overlap not sufficiently discriminate them; (ii) easy localizable queries; (iii) images where we are not able to decide if they were obtained in the database area; (iv) images that were out of database; (v) ambiguous position images which do not contain the relevant information about the location. Table 3.4 summarizes the query data.

First Figure 1.1 presents query images as well as the position of downloaded Google street-view images with their thumbnails.

	Image type	Short name	# of images
i.	Challenging queries	HR	81
ii.	Easy queries	ES	61
iii.	Unknow position queries	UP	23
iv.	Out of database queries	OD	21
v.	Ambiguous position queries	AM	14

Table 3.1: **Summary of query image set.**

Chapter 4

Representation of City Images

This chapter is focused on efficient image representation for fast image querying. The goal is to prepare the database for returning the most similar database image to the query image as fast as we can.

We used the approach based on popular text retrieval method which is used, for example, in Google web search engine. The chapter presents the used database image indexing step by step.

Section 4.1 describes the feature extraction method and its comparison with the state of the art of feature detectors/descriptors. Then, why and how the image is quantized into visual words is shown in Section 4.2. It includes the description of visual vocabulary construction and parameters setting. Section 4.3 is concerned with detection and suppression of confusing features.

4.1 Feature Extraction

Very often, images are represented by three matrices, one matrix for each color: Red, Green, Blue, but these matrices are not the applicable structure for image processing.

Images can be very efficiently represented by a suitable set of image features [MTS⁺05]. This section also compares different feature detectors and descriptors for the task of image retrieval previously described in Section 2.2.1, which is exactly the same as our initial location recognition method (describe later and introduced in Section 1.4). For each considered combination of a region detector and a descriptor (see Table 4.1) the localization performance was investigated on the geotagged omni-image POI dataset of about 1,300 images.

Similar to evaluation in object retrieval, detector/descriptor performance is measured using the precision-recall curve [PCI⁺07]. To report the single number performance

Short name	Detector	Descriptor	Implementation	# of desc.
MSER	MSER [MCUP02]	SIFT [Low04]	CMP (in-house)	1.53M
HessAffCMP	Hessian Affine [MTS ⁺ 05]	SIFT [Low04]	CMP (in-house)	2.34M
DoG	DoG [Low04]	SIFT [Low04]	CMP (in-house)	3.55M
SURF	SURF [BTVG06]	SURF [BTVG06]	ETH/Leuven [sur]	1.92M
HessLap	Hessian Laplace [MTS ⁺ 05]	SIFT [Low04]	Oxford [CbVC]	4.77M
HessAffOx1	Hessian Affine [MTS ⁺ 05]	SIFT [Low04]	Oxford [CbVC]	4.46M
HessAffOx2	Hessian Affine [MTS ⁺ 05]	SIFT [Low04]	Oxford (J. Philbin) [PCI ⁺ 07]	2.57M

Table 4.1: **Local invariant feature detectors and descriptors in our benchmark.** The last column shows the number of detected features on the training set of about 1,300 images.

	# of features	Detection [s]	Description [s]	Det.+Descr. [s]
MSER	1231	0.16	4.8	4.96
HessAffCMP	978	3.96	5.4	9.36
DoG	2605	3.1	3.1	6.2
SURF	2034	0.93	0.41	1.34
HessLap	1874	N/A	N/A	1.82
HessAffOx1	1773	N/A	N/A	3.44
HessAffOx2	2689	2.16	6.11	8.27

Table 4.2: **Comparison of running times on an example image for different feature detectors and descriptors.** With the exception of the SURF detector, which uses its own SURF descriptor, all extracted features are described using the SIFT descriptor. Different implementations of the SIFT descriptor have, however, different running times.

	1	2	3	4	5	6	8	9	10	11	Avg.
SURF	0.42	0.74	0.51	0.33	0.95	0.41	0.44	0.45	0.38	0.57	0.55
DoG	0.45	0.29	0.20	0.10	0.96	0.29	0.31	0.37	0.51	0.63	0.44
HessAffOx2	0.23	0.81	0.20	0.25	0.89	0.15	0.10	0.39	0.32	0.54	0.41
MSER	0.45	0.55	0.23	0.13	0.79	0.31	0.09	0.03	0.19	0.56	0.36
HessAffOx1	0.05	0.05	0.06	0.02	0.50	0.29	0.29	0.38	0.03	0.58	0.27
HessAffCMP	0.06	0.36	0.09	0.02	0.81	0.23	0.03	0.20	0.03	0.43	0.24
HessLap	0.05	0.08	0.18	0.08	0.45	0.08	0.10	0.03	0.03	0.10	0.14

Table 4.3: **Location recognition performance.** It was measured by average precision for 14 omnidirectional query test images and different local feature detectors/descriptors.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Avg.
SURF	0.03	0.21	0.04	0.66	0.14	0.29	0.55	0.68	0.53	0.14	0.07	0.18	0.18	0.76	0.32
DoG	0.02	0.40	0.03	0.75	0.06	0.39	0.60	0.44	0.57	0.03	0.05	0.05	0.03	0.79	0.30
HLap	0.03	0.04	0.03	0.28	0.16	0.39	0.11	0.39	0.46	0.17	0.02	0.73	0.04	0.46	0.24
HAffOx2	0.05	0.05	0.03	0.28	0.05	0.35	0.28	0.39	0.22	0.08	0.04	0.23	0.02	0.37	0.17
HAffCMP	0.02	0.03	0.01	0.19	0.02	0.06	0.40	0.02	0.34	0.04	0.12	0.02	0.01	0.02	0.09
HAffOx1	0.02	0.04	0.02	0.08	0.03	0.07	0.06	0.03	0.02	0.03	0.01	0.03	0.03	0.02	0.04

Table 4.4: **Location recognition performance.** It was measured by average precision for 14 perspective query test images and different local feature detectors/descriptors.

measure for entire set, we compute the mean average precision (mAP) as the arithmetic mean of AP across all queries. Note that a perfect performance (AP equivalent to 1) is obtained where the query is match to all database images that have a visual scene overlap (they are ranked as the first).

We decided to use SURF [BTVG06] features, because of their speed, robustness and good result compared to others [BTVG06, QLVG08, KSP09] and mainly the best localization performance results for SURF as is shown in Table 4.3 and Table 4.4. As a result, images are described by the collection of keypoints given by keypoint image position, scale and 64-dimension descriptor vector.

4.2 Image Indexing

Our database contains more than 20K images, where each image has about 2K SURF features. This involves about 400M keypoints in total. Searching this huge set is a not completely solved problem but some approximate methods with logarithmic time and acceptable results exist.

We decided to use visual vocabulary [SZ03] approach, inspired by the success of textual search, for example google web search engine.

4.2.1 Problem Formulation

In this paragraph, we present a more formal definition of the localization problem than in the previous discussion in the introduction Chapter 1.

We assume the image database I of geo-located images $i \in I$. In this work, the image database covers a part of a city. After that, we can formulate the location $l \in L$ as the group of images taken at the place l . The location l is also connected to x and y GPS coordinates. Each image i includes detected SURF features. Each SURF feature is described by the description vector \mathbf{d} . The set D is then the union of all feature descriptor vectors \mathbf{d} from the image database. A visual vocabulary is a collection of discriminative descriptor vectors $\mathbf{w} \in W$ and is often smaller than D . These discriminative descriptor vectors \mathbf{w} are called visual words.

After that, given the query image i_q described by features \mathbf{d}_q , we focused on selecting the most similar database image $i_s \in I$ described by \mathbf{d}_s vectors. If the most similar image i_s contains identical 3D object with the query image i_q , we assume that the query image i_q was obtained at the very close location to i_s database image. In the beginning, we suppose that searching for the most similar database image is equivalent to finding

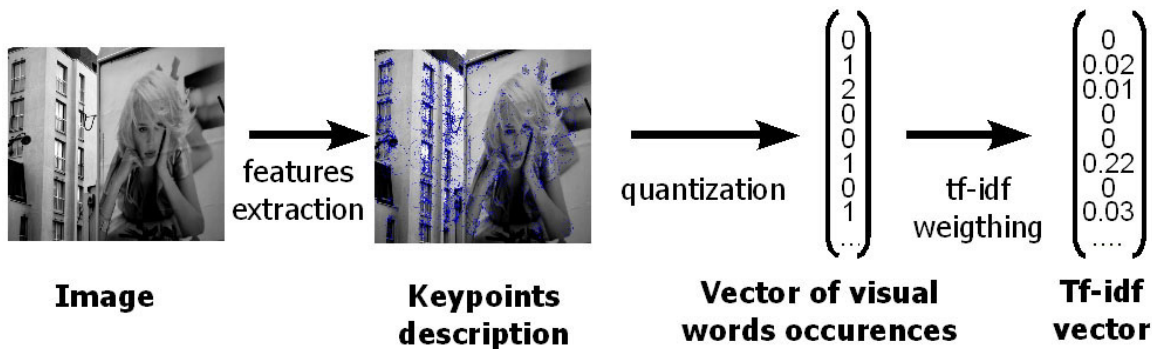


Figure 4.1: **Image representation using original Video Google [SZ03] approach.** We forget about images with descriptors and we will represent images by tf-idf weighted bag-of-visual-words vector.

the database image containing the set of the most similar descriptor vectors \mathbf{d}_s to query descriptor vectors \mathbf{d}_q .

4.2.2 Overview of the Text Search Inspired Image Retrieval

In this section, we present a brief and unified review of the work by Sivic *et al.* [SZ03]. Next sections then discuss it step by step as well as our modifications.

Given the localization problem, Section 4.2.1, we (i) compute the set of SURF features \mathbf{d}_a for each database image $i_a \in I$. Work [SZ03] computed visual vocabulary by k-means clustering. The cluster centers in the D constitute the visual vocabulary W . (ii) The query image descriptor vectors \mathbf{d}_q and all database images descriptor vectors $\mathbf{d} \in D$ are quantized by replacing them by the closest cluster center $\mathbf{w} \in W$. Note that each SURF 64-dimensional descriptor vector could be represented as a single number mapping it into the visual vocabulary: $\mathbf{d}_i \rightarrow \mathbf{w}_j$. It is sufficient to remember only the index of the assigned visual word instead of the vector \mathbf{d} . It also allows to forget about images and to remember only vectors of visual word occurrences for each image, called the bag-of-visual-words image representation. This image indexing stage is illustrated in Figure 4.1. (iii) Both, the query and the database images, are represented using tf-idf weighted visual word vectors, computed from the vector of visual words occurrences, see State of the art Section 2.2.1 where it is described in detail. This tf-idf vector favours discriminative visual words and downweights visual words which do not help in retrieval. Therefore, the similarity between the query and each database image is efficiently measured using the normalized dot product of two tf-idf vectors (query image tf-idf and database image tf-idf). This dot product is identical with the cosine of the

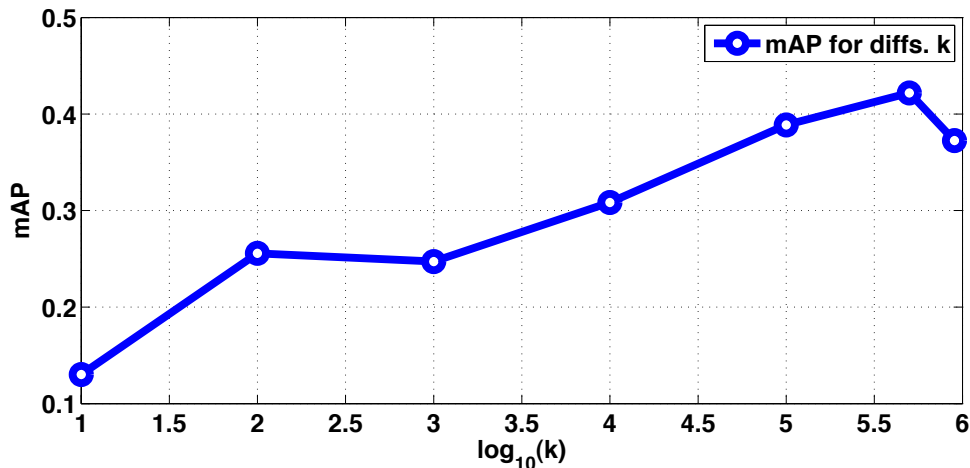


Figure 4.2: **Visual vocabulary size investigation.** Mean average precision recall as the function of number of visual words.

angle between these vectors, as was shown in Section 2.2.1. Database image ranking by the dot product of tf-idf vectors is our initial localization stage. It will be also discussed in the Chapter 5 focused on details of localization.

4.2.3 K-Means Clustering to Create a Visual Vocabulary

Visual vocabulary is constructed by k-means clustering of all image descriptors D . In this section we investigate location recognition performance as a function of k-means algorithm parameters. We show the localization performance on the same image dataset as detectors/descriptors experiment described in Section 4.1. On large-scale data, we measure real localization performance (the same as the Section 4.1) instead of the clusters distortion.

Vocabulary size. Figure 4.2 shows the location recognition performance, measured by mean average precision for different vocabulary sizes. The vocabulary is built from 1.92M SURF descriptors and the k-means algorithm was run for 20 iteration from a random initialization. Similar to [SZ03, SBS07] we observed a peak in performance at around 0.5M visual words. The intuition of the existence of the peak is that when the number of clusters is too small, the resulting visual words are non-discriminative (different features are assigned into the same visual word). On the other hand, large number of clusters allows to assign the feature of the same scene/object into the different visual words.

Although the performance peak is at around 0.5M, we decided to used visual vocab-

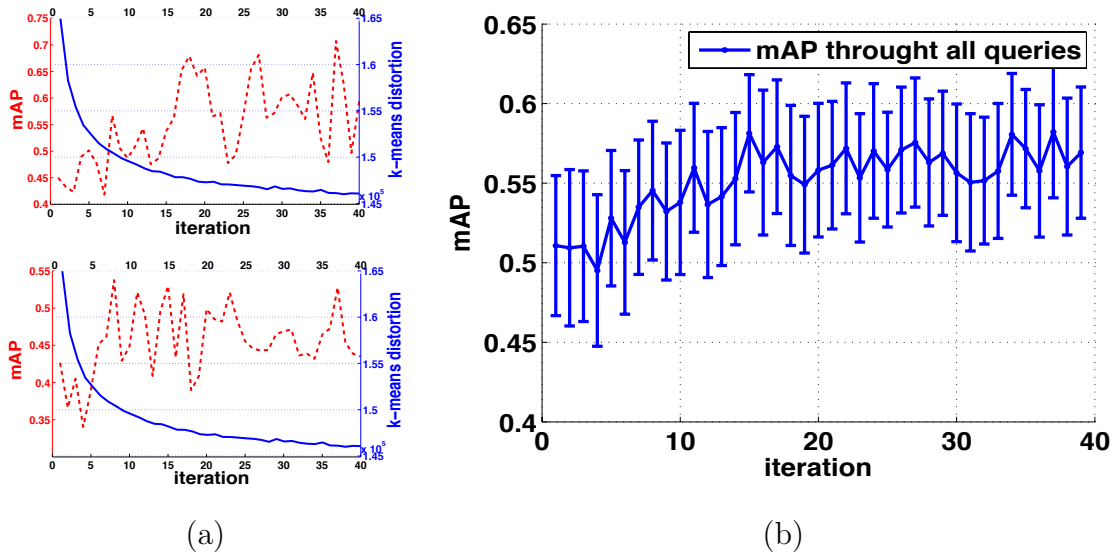


Figure 4.3: **Average precision as a function of the number of k-means iterations.** (a) Two examples of query images. Each plot shows the average precision (red) and the k-means distortion (blue). Note that the distortion steadily decreases with a small level of noise due to the approximate nearest neighbour search. The average precision is highly variable, but there seems to be slight increase with the number of iterations better visible in (b) where the mean AP through all queries is shown.

ulary of 130K visual words. It does not decrease the performance significantly (note that x-axis in Figure 4.2 has logarithmic scale) and we observed that image indexing using smaller vocabularies is faster than with larger vocabularies.

The number of k-means iterations. We investigate location recognition performance with respect to the number of k-means iterations. Figure 4.3(a) shows the average precision as a function of the number of k-means iterations for two location test queries and Figure 4.3(b) shows the mean average precision over all test queries. Note that the performance seems to level off at around 20 iterations.

4.2.4 Visual Vocabulary as the Set of Most Informative Words

When the number of dataset images increase significantly, it becomes impossible to cluster all data at once. The most popular method [CPS⁺07, PCI⁺08] is to select a uniform by distributed subset of images (training data). Unfortunately, we have to assume that our image subset is the representative one, it means that features have the same distribution through the whole database. Although it carries out in the cited papers, it does not in this work. If we work with city-street image database, each image subset would miss

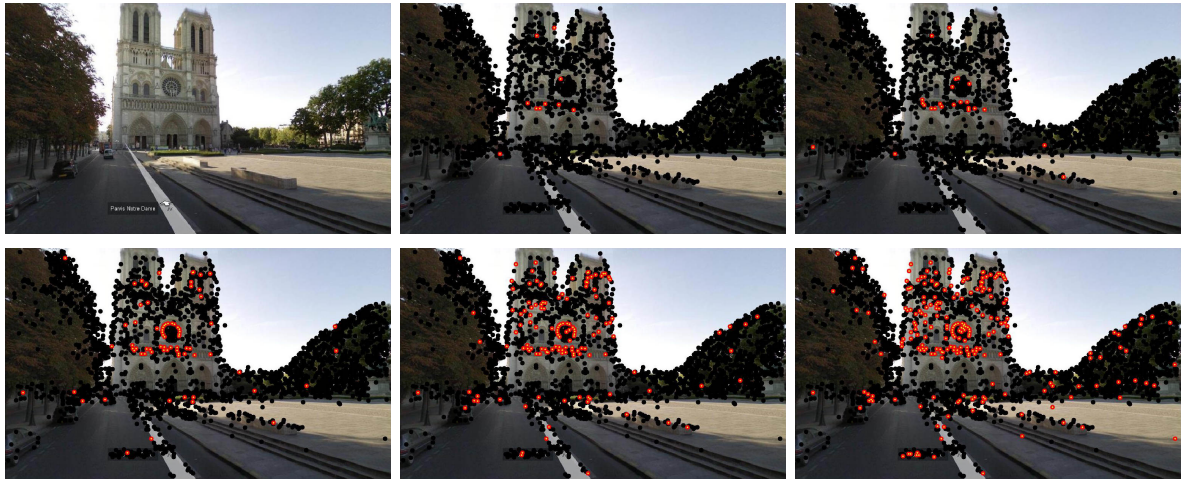


Figure 4.4: **Most informative features selection.** Left top image is an example of image and then continue in images with highlighted informative features for greater number of selected informative features sorted by information gain. Black color signalize detected features and red-yellow represent selected informative visual words for the specific threshold. We can see informative features on buildings (informative locations objects) and not on trees, payments etc.

method, vis. voc. size	# correct $n=1$	# correct $n=10$	# correct $n=50$	# correct $n=500$	# correct $n=2000$	# correct $n=10000$
a. original Schindler, 2M	9	11	15	23	37	50
b. original Schindler, 6M	16	17	22	30	40	50
c. expanded neighborhood, 6M	9	17	21	31	39	50

Table 4.5: **Summary of different parameters for the informative features selection.** Performance for informative features selection and the number of informative features to create visual vocabulary on initial retrieval (original Video Google) method as the number of correctly returned images in the first n retrieved entites. Ideal localization algorithm correctly sets 50 queries.

images of the missing locations. Therefore, every place contains characteristic features and we are not able to obtain these features without the images of the missing location.

Schindler *et al.* [SBS07] presented a solution without sampling subsets. The goal is to find specific (most informative) features occuring in all images of a specific location but rarely occuring anywhere outside of this location. Visual vocabulary is then constructed only from these most informative features. Process of finding the informative features is: Firstly, the image database is divided into small groups with the sufficient size for features clustering. Thus, all geo-tagged images I are divided into several groups of position close images I_a (L_a respectively, where L represents locations of images I), Figure 4.5 highlighted these groups by different colors. Importantly, the group size has to be sufficient for k-means clustering to create the visual vocabulary W_a . It results in

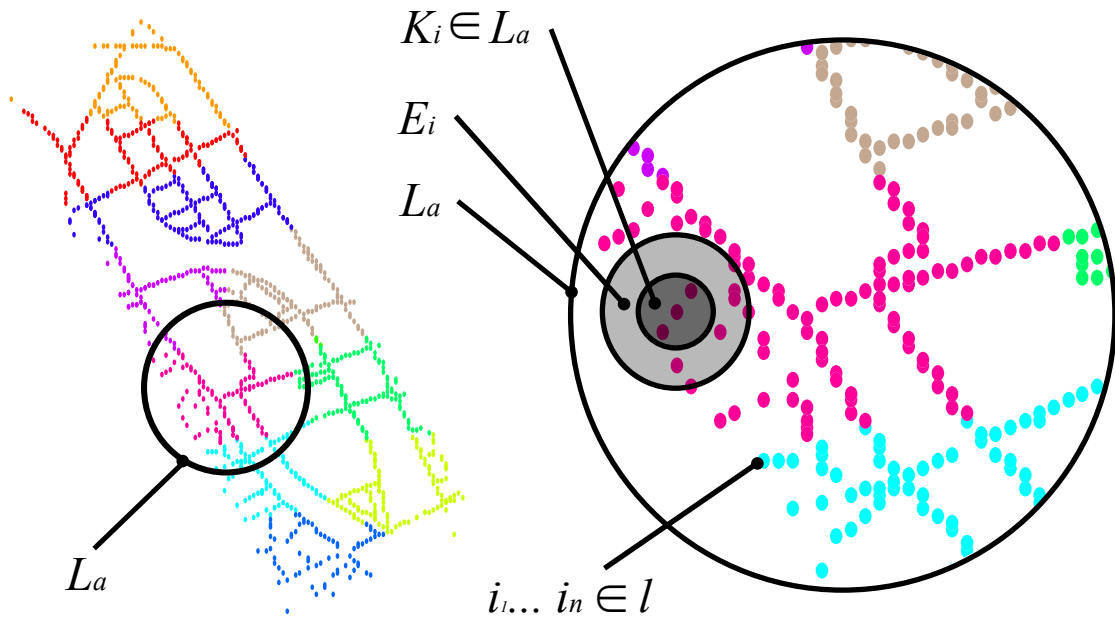


Figure 4.5: **Image location clustering.** Figure shows image position (dots) and clusters in which they were assigned, each cluster has a unique colour.

the specific visual vocabularies W_a for every group I_a (L_a). Secondly, each group I_a is again divided into small subgroups $K_i \subset L_a$ of (1-5) locations, it is also illustrated in Figure 4.5. Finally, we focused on the selection of most informative visual words $w_j \in W_a$ for the location K_i based on informative gain defined as,

$$I(K_i|w_j) = H(K_i) - H(K_i|w_j), \quad (4.1)$$

where the location entropy $H(K_i)$ is constant across all visual words at the group of locations L_a . Next, the probability of being at the location K_i when observing the visual word w_j will be defined,

$$P(K_i | w_j) = \frac{\text{occur}(K_i + E_i, w_j)}{\text{occur}(L_a, w_j)}, \quad (4.2)$$

where $\text{occur}(A, B)$ represents the number of occurrences of visual word B at location A . E_i is the set of images closer than 28m to the location K_i , illustrated in Figure 4.5.

In the end, we select features of most informative visual words per each location K_i . 6M of informative features are then taken to construct the visual vocabulary. 6M is an experimentally evaluated maximum number of features for running k-means clustering. Figure 4.4 shows an image with increasing number of informative visual words.

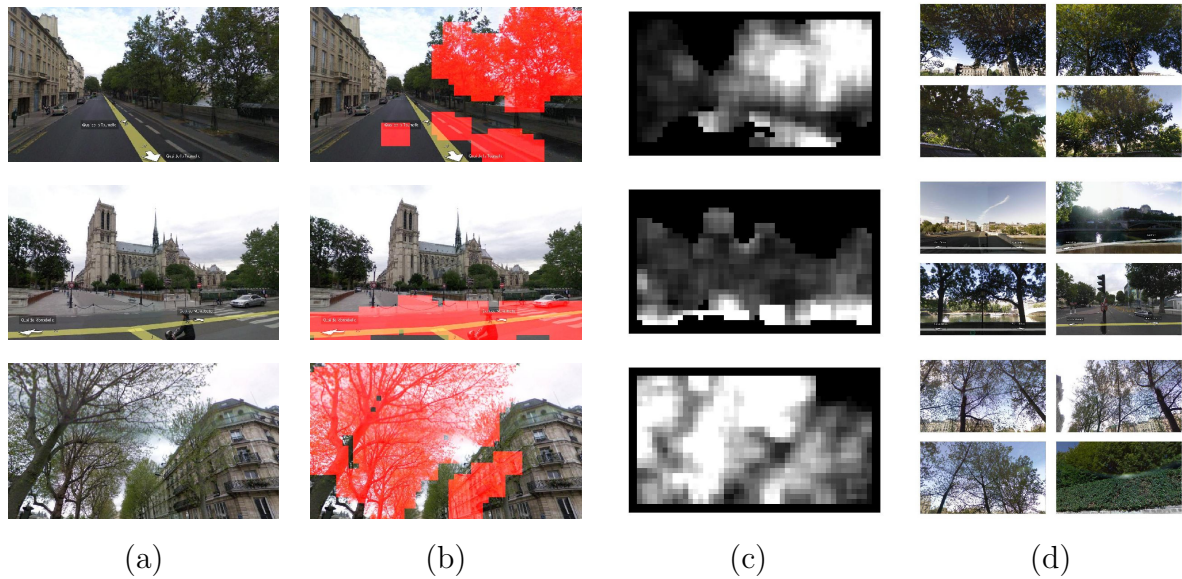


Figure 4.6: **Example of detected confusing regions.** (a) Original image. (b) Detected confusing image regions. Not how areas such as trees and roads are covered. (c) Local confusion score. Intensity indicates confusion score ρ . (d) Selected four mismatched images from different locations.

Table 4.5 shows location recognition performance. Unfortunately, the presented method doesn't give a significant improvement for the top-ranked image ($n = 1$) when increasing the number of selected features to create a visual vocabulary (from 2M to 6M). More testing and selecting better parameters could be one of the ways to make the method more useful.

4.3 Detecting and Suppressing Confusing Features

Images in city-street databases contain objects like trees, pavement, sky or water, which are not informative for recognizing a particular location as either (i) they appear frequently throughout the city, or (ii) they cannot be reliably matched by the current local feature based algorithms. To address this issue we aim at detecting and removing these features automatically. We used the fact that an image should not match too well to images from distant locations.

4.3.1 Local Confusion Score

Given the database image $i_k \in I$ we find a set $I_n \subset I$ of top confused images. This is achieved by retrieving the most similar images, which are further than 370 meters away

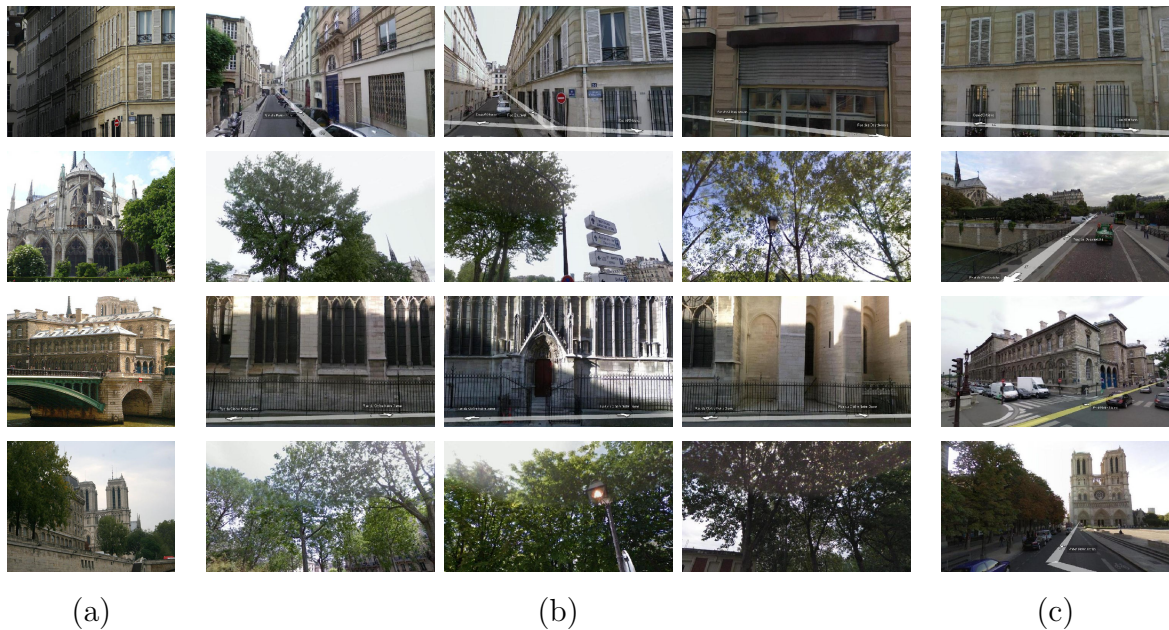


Figure 4.7: **Improvement in location recognition based on suppressing confusing features.** (a) The query image. (b) The top ranked image after initial retrieval and spatial verification. (c) The top ranked images after suppressing confusing image regions in the geo-tagged database. Note that the highly ranked false positive images shown in (b) are suppressed in (c).

from i_k location. It allows to retrieve similar images with no correct matches.

We defined a sliding windows w on a dense image grid. We evaluate local confusion score as,

$$\rho_w = \frac{1}{n} \sum_{k=1}^n \frac{M_w^k}{N_w}, \quad (4.3)$$

within a sliding window w by counting the number of matching visual words M_w^k weighted by the number of detected visual words N_w within a sliding window w . The score ρ is high when a large portion of visual words (within a sliding window) matches to “confused” images. We used sliding windows of size 75x75 pixels on a 5 pixels grid. Figure 4.6 illustrates the distribution of the local confusion score for several selected database images.

Note that our local confusion score is different from the tf-idf weighting, see Section 2.2.1, as the presented approach allows to remove features (tf-idf globally removing visual words) confusing for a particular image.

4.3.2 Suppressing Confusing Features

We off-line precomputed local confusion score for the whole city-street database and all features with the score greater than a threshold were removed. The remaining features

were then quantized using visual words. Although the initial retrieval stage runs only on non-suppressed features, verification, see Section 5.2, uses all image features as the matching object can be situated inside a confusing region.

Figure 4.7 presents how suppressing of confusing features improves final localization on selected images. We show several query images containing many confusers (trees, brics, etc.) which appear in the top-ranked confusers images. When confusers in city-street image database are suppressed, the correct images are found.

Chapter 5

Location Recognition

Previous chapter have been concerned with the indexing of image databases. In this chapter, we describe localization of the query image using this indexing approach. The chapter is structured as the whole localization pipeline, where each section represents each localization stage, except that the last two sections focus on image expansion and video localization.

One of the correct views at our approach is that each stage of the processing resorts database images to start with the most promising ones and rejects some portion of the least promising. Thus latter stages process gradually less and less images and can

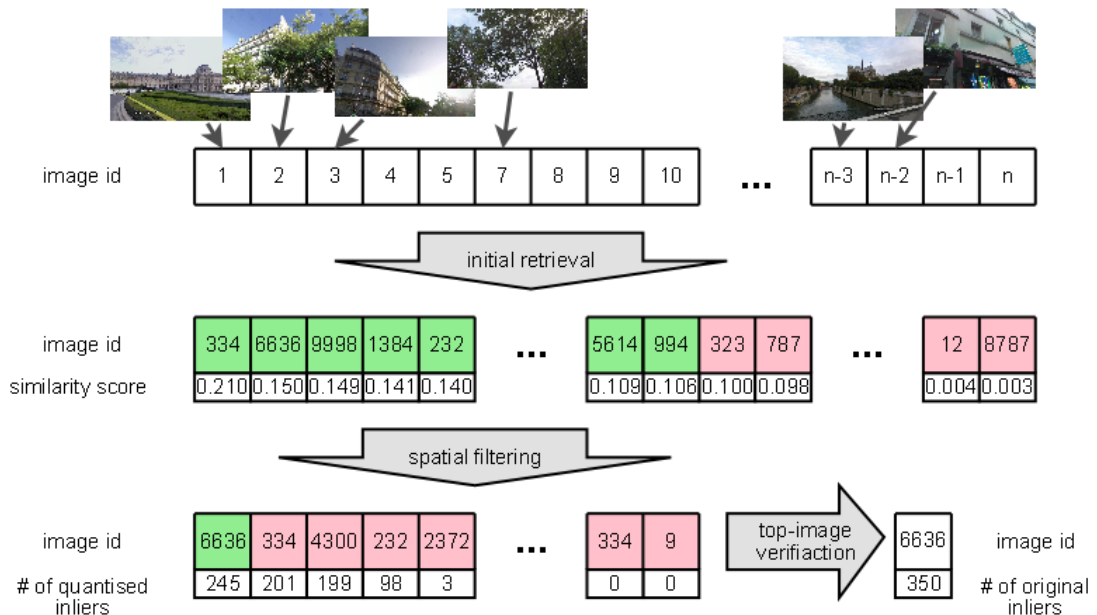


Figure 5.1: **Location recognition as the cascade filtering.** Each stage (represented as the arrow) resorts the image database. After that, the selected number of images is kept (highlighted by green colour) for the next stage and remainder images (red) are omitted once for all.

therefore spend more and more time on each image. This is a classical cascaded decision, which is our situation works as long as the right image does not get rejected. Figure 5.1 illustrates localization stages.

Firstly, Section 5.1 describes finding a small set of candidate images from the entire geotagged database. Then, top-ranked candidate images are re-ranked taking into account the spatial layout of local quantized image features, Section 5.2. Section 5.3 presents the final localization stage describing original features matching aiming at decision if the query image is obtained in our database at all. In addition, Section 5.4 shows significant localization improvement using query image enriching and finally, the method for localization of video is presented in Section 5.5.

5.1 Initial Retrieval of Candidate Locations

The first part of the localization cascade is a standard text search based image retrieval system [SZ03]. The goal is to find a small set (50) of images contains at least one correct database image to the query one as fast as we can. The stage has to work on extremely large number of images (millions).

The initial retrieval stage retrieved the top 50 images ranked by the similarity score computed as the dot product (see Equation 2.7) of tf-idf (described in Section 2.2.1) query and database images representation, discussed in Section 4.2.2. This dot product of two normalized vectors is identical with the cosine of the angle between these two vectors. Thus, a higher score represents higher image similarity.

Table 5.1 presents times for the first stage, which includes previously discussed image representation (see Section 4.2.2) and matching to database. The experiment was run on a 3Ghz Xeon and PI dataset. Note that features detection/description and the quantisation (feature matching to visual vocabulary) are slower in the order of magnitude, but these processes are independent from the size of the city image database. It also highly depends on the image resolution, i.e. we found that detection/description of SURF features takes

	200images [s]	mean for one image [s]
SURF detection/description	132	0.660
quntisation	148	0.740
tfi-idf computing	0.297	0.00148
tfi-idf based matching	7.95	0.00398
Σ	288.24	1.44

Table 5.1: **Time cost of initial location recognition method.**

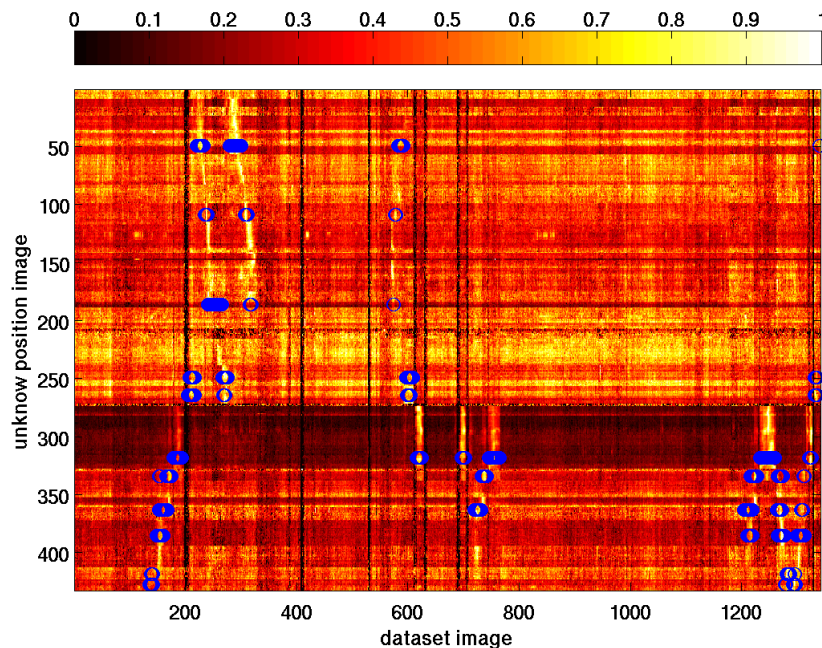


Figure 5.2: **Example of city-street image scores after the initial retrieval process on 1,3K Prague images dataset.** For each pair of and database images the value of the dot product is computed. Blue circles represent the ground-truth for selected query images. Therefore, the ideal location recognition algorithm should return high values in the blue circles.

about 170ms for the 500x400 resolution image (compared to 660ms for the 900x700 image).

Figure 5.2 illustrates the result of the initial retrieval. For every database image, the similarity score is computed by the dot product with the query image (dot product of the tf-idf vectors). Figure 5.2 shows dot products computed on POI dataset.

5.2 Filtering by Spatial Verification

Here, we focused on the verification of the 50 top-ranked images after the initial retrieval. To achieve this goal, we assume that the 3D structure visible between the query image and each candidate image can be approximated by a small number of planes (1-5). For the candidate correspondences obtained based on quantisation on visual words we estimated number of homography mappings \mathbf{H} to find inliers from the detected image features. First image features x_1 were projected into the second image,

$$\alpha x_2 = \mathbf{H}x_1. \quad (5.1)$$

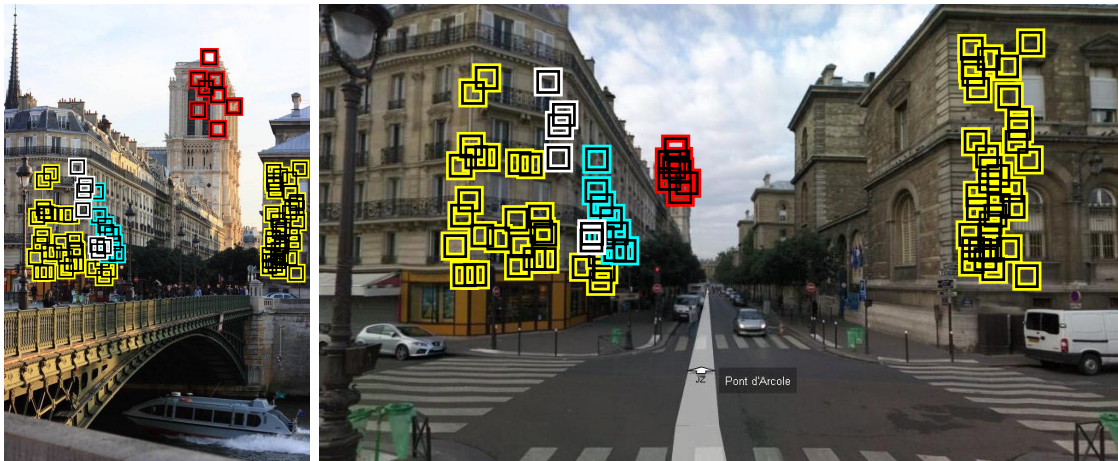


Figure 5.3: **Example of estimated multiple homographies.** Query and selected street-view image, inliers of each homography (verified SURF features) are highlighted by colored squares.

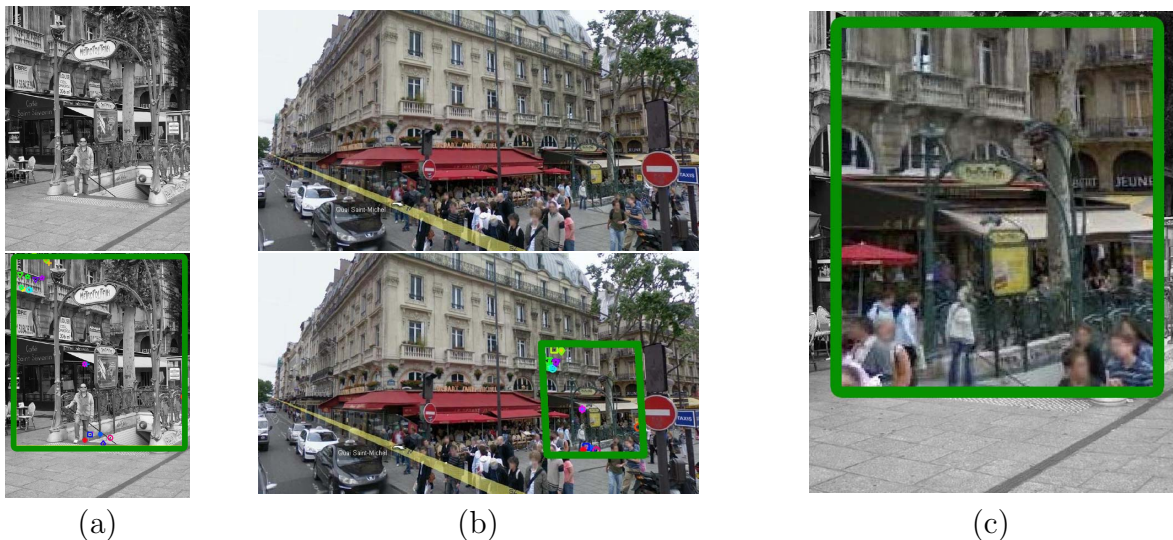


Figure 5.4: **Projection using the estimated homography.** (a,b) Query image and the database image. The first row shows the original images, the second one presents images with estimated inliers satisfying homography \mathbf{H} (inliers are inside the highlighted bounding box). (c) Query image with the projected bounding box by \mathbf{H} . This bounding box (green color) covers the area with inliers.

The homography \mathbf{H} was estimated from the LAF [OM02a] keypoint representation obtained from the feature point position and scale. This LAF representation allows faster computation time [OM02b] as the \mathbf{H} matrix could be estimated from a single pair of correspondences using LO-RANSAC [CMK03].

Then, the multiple homographies are fitted to 50 top-ranked images and candidates

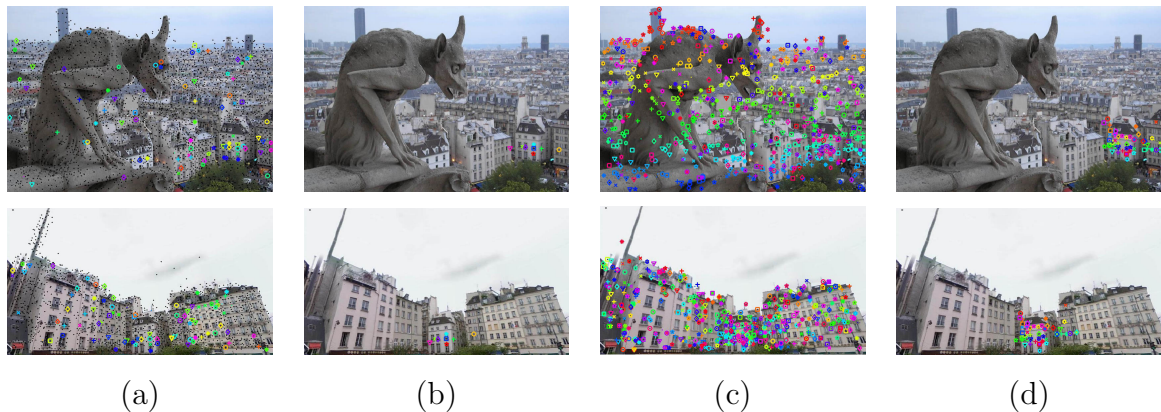


Figure 5.5: **Comparison of matching based on visual vocabularies with standart techniques.** Query image (top) with the top-ranked street-view image (bottom). We show several results, (a) matches based on visual vocabulary matching, (b) estimated inliers from visual vocabulary matches, (c) matches obtained after the second nearest neighborhood matching on feature descriptors and (d) inliers after the 2nd nearest neighborhood matching.

are re-ranked based on the number of inliers. We seek to select only the first top-ranked image as the input for the next stage.

Figure 5.3 shows an image pair with the detected multiple homographies. Figure 5.4 illustrates that the homography is the projection such that \mathbf{H} mapping points of the first image into the second image as is formulated in Equation 5.1. We show the projected first image region to the second image based on the estimated homography.

5.3 Verification of Top-Ranked Location

The goal of the final stage is to decide if the top-ranked image is the correct one or not. We verify the top-ranked image from the spatial filter by matching on the original (non-quantised) features.

Second nearest neighborhood matching was used as the matching technique to find tentative matches between images. A query image SURF feature descriptor vector \mathbf{x} is assigned as the match with the nearest database image SURF descriptor vector \mathbf{y}_x^{nn} if,

$$|\mathbf{x} - \mathbf{y}_x^{nn}| < \tau |\mathbf{x} - \mathbf{y}_x^{2nn}|, \quad (5.2)$$

where \mathbf{y}_x^{2nn} is the second nearest descriptor to \mathbf{x} and τ is the parameter affecting the number of matches, which was set to 0.91 in this work.

Note that the verification of the previous Section 5.2 could valuate the top-ranked image on the number of inliers as well. In contrast to Section 5.2, the verification using

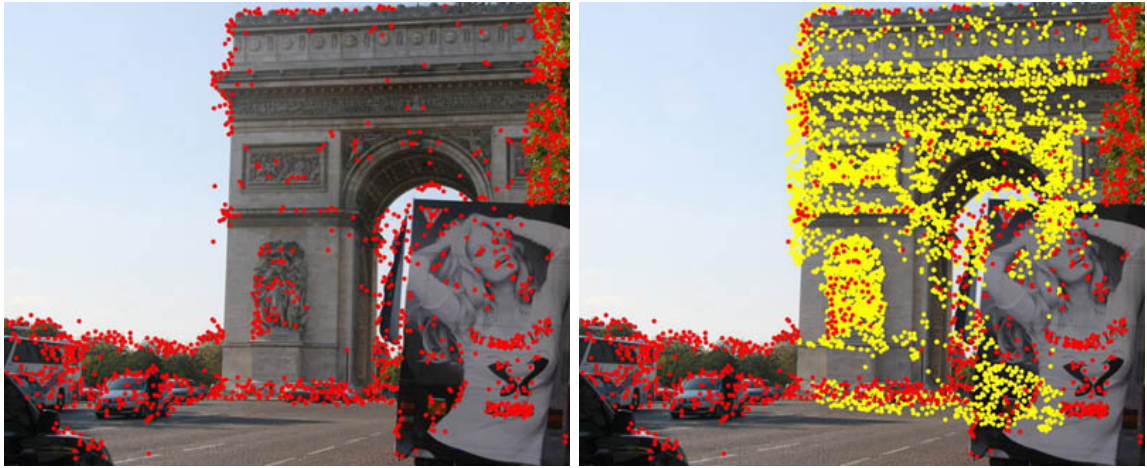


Figure 5.6: **Example of image with a standard set (red) and an enriched set (yellow) of features.** Note that the majority of features were added to the relevant structure - Arc de Triomphe.

the correspondences computed from the original SURF features has better performance than when using the quantized features, see Figure 5.5. As this verification is much more time expensive it is applied only to the top-ranked candidate images.

In the final localization stage, the number of these non-quantised inliers is estimated between the query image and the top-ranked street-view image. We consider experimentally defined threshold as the minimal number of inliers to decide whether is the top-ranked image depict the same location as the query image.

In Chapter 6 we show how this approach filters negative locations.

5.4 Location query expansion using non-geotagged images

Here, we describe the approach to improve localization using ungeotagged image database. Considering that standard matching techniques are useless for extreme view-point, lighting changes or partial occlusions by other objects, we turn to a collection of images downloaded from photo-sharing sites (Flickr, Panoramio...). These images are not geotagged but might contain multiple images of the same places captured by different photographers from different viewpoints or at different times.

We present the method to enrich the query image aimed at having the same information in the query image as in the close database images. The method is based on image retrieval method known as the Query Expansion [CPS⁺07]. The location expansion process is as follows: (i) firstly, we match the query image to the public non-geotagged image

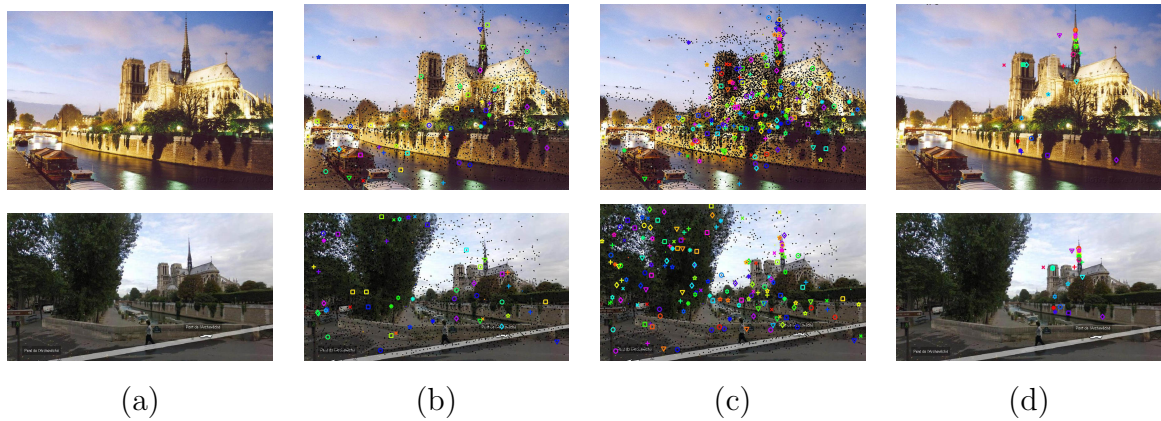


Figure 5.7: **Matching improvement using location expansion.** Query image (top) with the top-ranked street-view image (bottom). Figure shows several results: (a) original image, (b) tentative matches based on assignment to visual vocabulary, no inliers were found for this matches due to only two good matches in the scene, (c) expanded features were added to the query image, (d) estimated inliers using the expanded features. The number of tentative matches increased from 56 to 153.

database by tf-idf vectors similarity (the same as the initial retrieval); (ii) we use the 50 top-ranked non-geotagged images containing more than 18 inliers to enhance the query image. The inlier threshold was set to 18 to be conservative and not to expand with the non-matching images as they would pollute the query. (iii) All features within a rectangular region enclosing inliers are transferred using the computed homography. After that, the query image is expanded by the features, see Figure 5.6 presenting detected and expanded features. In addition, query image tf-idf vector is then updated as the mean of all verified tf-idf vectors as in Chum *et al.* [CPS⁺07] approach, described in Section 2.2.1.

The expanded query image contain original and expanded features. Image is represented as the new (mean) tf-idf vector as well. It is then used as the standard single image input to the whole location recognition algorithm.

Figure 5.7 illustrates the improvement of the spatial verification using the enriched image descriptors. Figure 5.8 presents the selected query images and the improvement of the location recognition after the location expansion.

5.5 Video Localization

There are several video sharing sites such as YouTube [You] on the internet. This section is focused on position recognition of videos. General videos should contain more location information thanks large number of images capturing each viewpoint and camera

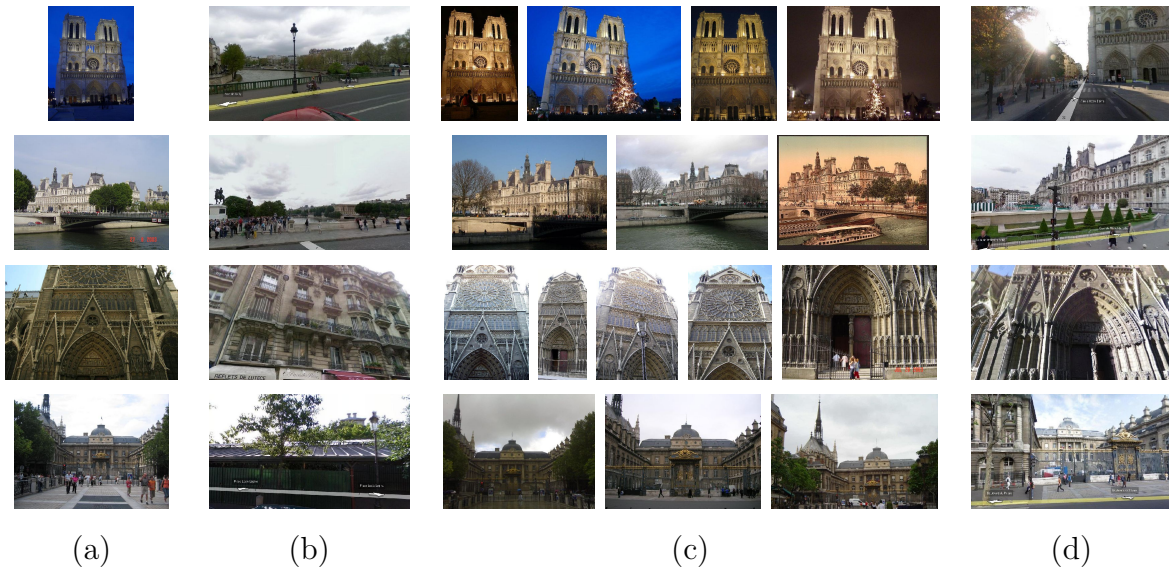


Figure 5.8: **Improvement based on enriching from non-geotagged database.** (a) The query image. (b) The top match (incorrect) obtained from the geotagged database using the original query. (c) Example matches found in the non-geotagged database. Note the variation in lighting and viewpoint. Features from these images are added to the original query. (d) The top found image (correct) found in the geotagged database after expanding the query with non-geotagged images in (c). Note the matches in (b) are corrected after the query expansion.

movement.

We used the formulation for the location recognition of video as the localization of sequence of images. This allows us to use a modification of the presented single image localization approach.

Firstly, this section present Bayes Filtering [GNTvG07] as the theory for position estimation from the upcoming images. Secondly, we show how we used it for video localization using previously presented methods.

5.5.1 Bayes Filtering

The *Belief function* that estimates for each world place $l \in L$ how we belief that we are occurring at l at time t , given by all previous observations $i_{i...t}$, can be defined as,

$$Bel(l, t) = \eta \boxed{P(i_t|l_t)} \sum_{l_{t-1} \in L} [\boxed{P(l_t|l_{t-1})} Bel(l, t-1)]. \quad (5.3)$$

Equation 5.3 depends on the probability that we see i at time t at position l (ensor model) and also on the probability that we are at l in t if we were at the same location l in $t-1$ (motion model) and recursively for observations till the start of data acquisition. The sensor and the motion models are part of the final solution of the video localization.



Figure 5.9: **Sample from the input sequence.** These images have extremely poor quality. Their resolution is 470x350 which makes the localization more challenging.

Sensor Model defines the probability of acquiring query image i_t at known location l_t . At a certain time t . Previous sections show how we match query images to street-view images I , where each image $i \in I$ is connected to a specific locations l . Therefore, the problem is to match the query image to the database of geo-tagged images, which is given by the number of inliers. Also, using the similarity score based on the dot product of two tf-idf vectors is another possibility when the inliers are not estimated. Thus, the sensor is modeled by a Gaussian,

$$P(i_t|l_t) = \frac{1}{\beta} e^{sim(i_t, i_t^d)/\sigma}, \quad (5.4)$$

where β and σ are parameters estimated experimentally. sim corresponds to the measurement of the image similarity, e.g. the number of inliers.

Motion Model models the distance between the geo-location l_t of the actual top-ranked image and the geo-location l_{t-1} of the last top-ranked image. The correct way to compute it is to use the distance between two points (l_t and l_{t-1}) and take into account all streets. It means to model the city streets. As this street modeling is very challenging task, we decided to use an approximate solution as the clearance,

$$|l_t, l_{t-1}| = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}, \quad (5.5)$$

where the location l is defined as the map position: $l = [x, y]^T$. The motion model is then realized with a Gaussian too,

$$P(l_t|l_{t-1}) = \frac{1}{\beta} e^{-|l_t, l_{t-1}|/\sigma}. \quad (5.6)$$

As the motion model depends only on the database of city images, the complete model was precomputed off-line. Computation time took about 55s.

Note that all functions model probabilities. So values have to be weighted and the maximal number has to be one.

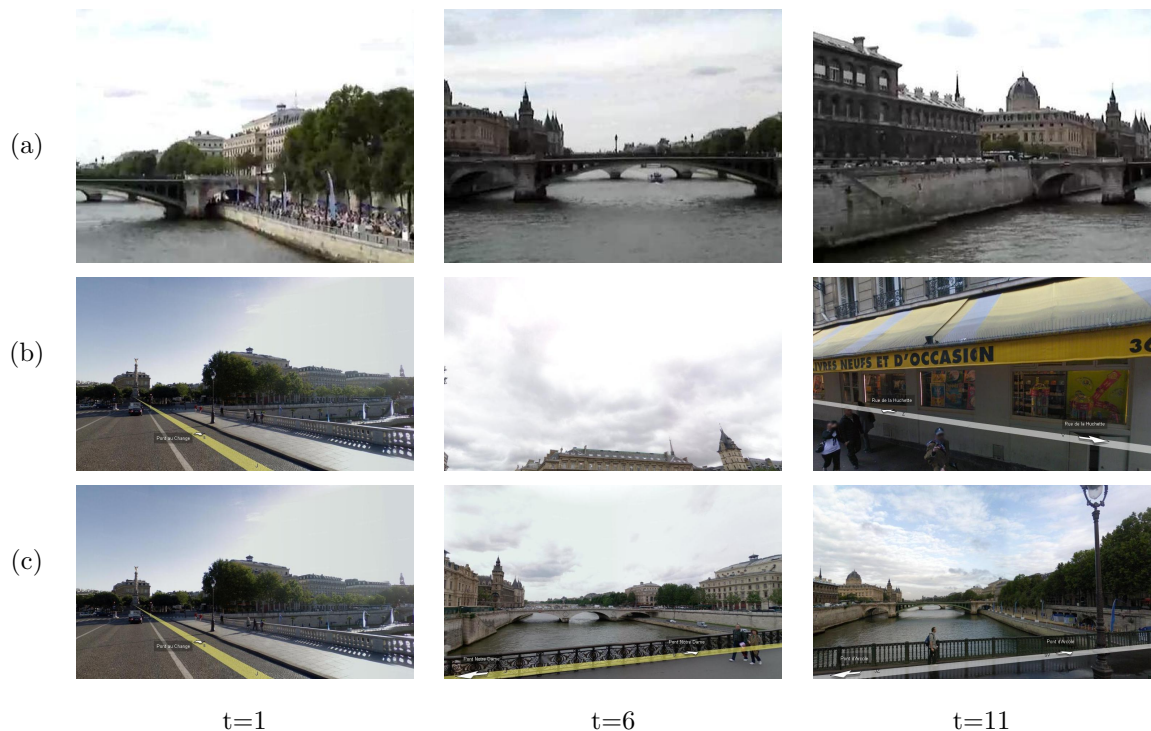


Figure 5.10: **Improvement based on using bayes filtering for localization of the image sequence.** Each column presents results for one specific time. (a) query image from the video. (b) top-ranked image without bayes filtering after the initial retrieval. (c) top-ranked image with bayes filtering after the initial retrieval.

5.5.2 Localization of Upcomming Images

At the beginnig, every location is equally probable before the firts input image comes. The probability will be updated based on images as they come. For each new input image the sensor model $P(i_t|l_t)$ is computed and the precomputed motion model with the last $Bel(l, t-1)$ values is used to re-estimate current $Bel(l, t)$ for the incoming images. Similar to previous localizations mehods, we are looking for the location with the maximum of the belief function value as our result.

Figure 5.9 presents an example of our video input from a collection of images crawled from the YouTube site. Results are illustrated in Figure 5.10. You will notice that the first image found (obtained at time $t = 1$) is the same as the image obtained by single image localization described in Section 5.1 and in Section 5.2, see Figure 5.10. This is because the first image is processed as a single image without using any information about the consecutive images in the sequence. Consequently, the belief function of the second input image is computed from its sensor model and the previous result. Benefits are shown in the Figure 5.10.

Chapter 6

Experimental Evaluation

In this chapter the performance of the proposed image based localization method is evaluated. We aim at testing the whole localization method.

Firstly, Section 6.1 discusses the way of showing limitations of the datasets. Secondly, Section 6.2 presents the evaluation of our approach on the location recognition problem. Last Section 6.3 suggests a way how to obtain the GPS coordinates and the street name of the query image.

6.1 Gold Standard Method

Here, we investigate the limitations of our database. Is it really possible to successfully localize all query images? To answer this question, we scored each database image by the number of original feature inliers with the query images. This is currently the best way to find the most similar database images, however it is an expensive method. It takes more

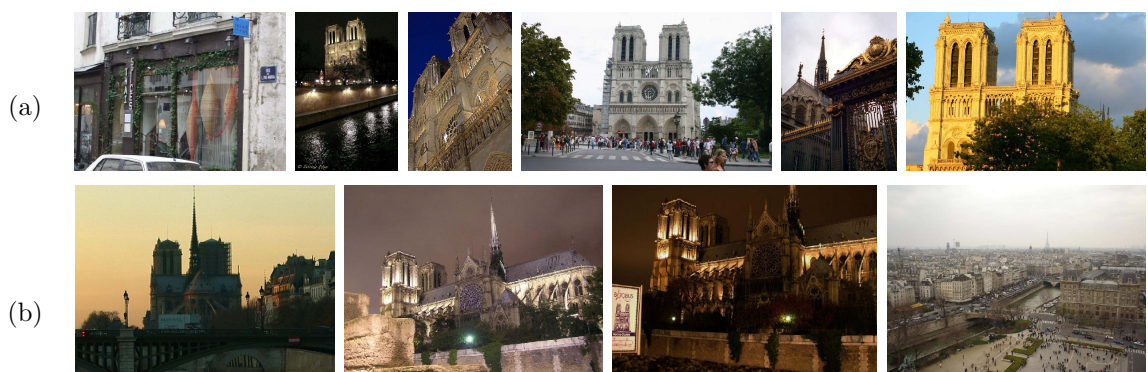


Figure 6.1: **Examples of not correctly localized query images after the Gold standard method.** (a,b) All these images have visual overlap with the image database, but they were not localized. The second row images (b) were correctly localized when our localization approach was used.

than 1 second for one image matching which results in about 6 hours of computing for one query. It used the verification procedure described in Section 5.3 in detail. The goal of all image/object retrieval approaches (f.e. Sivic *et al.* [SZ03], Chum *et al.* [CPIZ07], or our first and second localization stages) is to obtain the best approximation of this method.

In this experiment, we computed the number of inliers for each pair of query and database images on PI dataset. The dataset contains 200 query images and the subset of 142 images contains a visual overlap with the database of street-view images, described in Section 3.4 in detail. In the gold standard method, the most similar image corresponds to the one with the highest number of inliers. Then, after the manual inspection, we observed that 109 out of 200 query images have visual overlap with images in our database. Note that 109 is the maximal achievable number of correctly localized images based on image features and it is the number we are trying to achieve.

We found 91 not correctly localized query images. Although most of them were taken at a different location. Some of them have visual overlap with the database images but they represent extreme occlusions or view point changes, see Figure 6.1(a,b). In addition, we found 4 not correctly localized images by the Gold standard method, but our improvement of the localization algorithm (location expansion + suppression of confusing regions) results in the correct localization of these 4 images, see Figure 6.1(b). Although 4 looks like an extremely small number, we observed it as enough as this Gold Standard method correctly localized very challenging images with for example lighting changes.

6.2 Location Recognition

Note that we presented a sequence of results on the location recognition to demonstrate the improvements when using confusing regions suppression, Figure 4.7, and the location expansion, Figure 5.8. Both experiments were run on PI dataset

In this section, the main localization test is presented. We used PI dataset because it is large enough and street-view images were downloaded fully automatically. Two main experiments are shown here. Firstly, the results after the initial retrieval, see Section 5.1, and spatial verification, described in Section 5.2, are shown in Table 6.1. The performance of the location recognition after these methods was measured as the number of correctly retrieved top-ranked images. Note that it is the input to the final verification stage of our localization cascade, see Section 5.3. The evaluation of the top-ranked image verification is shown in the second experiment in Table 6.2. Considering the manual inspection, the

performance after the this final localization stage was measured as the number of verified street-view images containing a visual overlap with the query image (true positive-TP). On the other hand, we also computed the number of verified street-view images without a visual overlap with the query image (false positives-FP). Results are discussed below.

6.2.1 Initial Retrieval & Spatial Verification

Localization performance after each localization stage is presented in Table 6.1. The performance is measured by the number of correctly matched images. Remember that only 142 out of 200 query images have visual overlap with the database, see Section 3.4, and 109 query images were localized after the Gold standard method.

Spatial verification outperforms initial bag-of-visual-words matching in all stages, which means that the correct match is included within the fifty top-ranked initial retrieval entities and spatial verification finds it.

All presented new techniques: (b) query expansion and (c) confuser suppression, significantly improve the baseline localization, from 56 correct results to 78 and 65 respectively. Note that original Video Google approach correctly localized only 38 query images. We also find both methods complementary because together they (d) correctly localized 88 queries. Figure 6.2 illustrates correctly recognized locations.

Considering the previous Section 6.1, we localized 84 out of 109 images which were localized using the Gold Standard method (77% query images were localized). In addition, we correctly localized another 4 images although they were not localized using the Gold Standard method. Note that these 4 images plus 84 images make together 88 correctly localized query images in Table 6.1(d).

Also, we have found 50 images captured in geotagged database area which were not localized due to the large changes in viewpoint, scale, lighting condition and occlusion by another object. Examples are shown in Figure 6.3(b). These 50 not localized images

Method	# correct <i>initial retrieval</i>	# correct <i>spatial verification</i>
a. Baseline location recognition	38	56
b. Query expansion	49	78
c. Confuser suppression	52	65
d. Confuser suppr.+Query exp.	59	88
e. Gold standard	N/A	109

Table 6.1: **Location recognition performance.** The number of correctly localized test queries for different location recognition approaches.

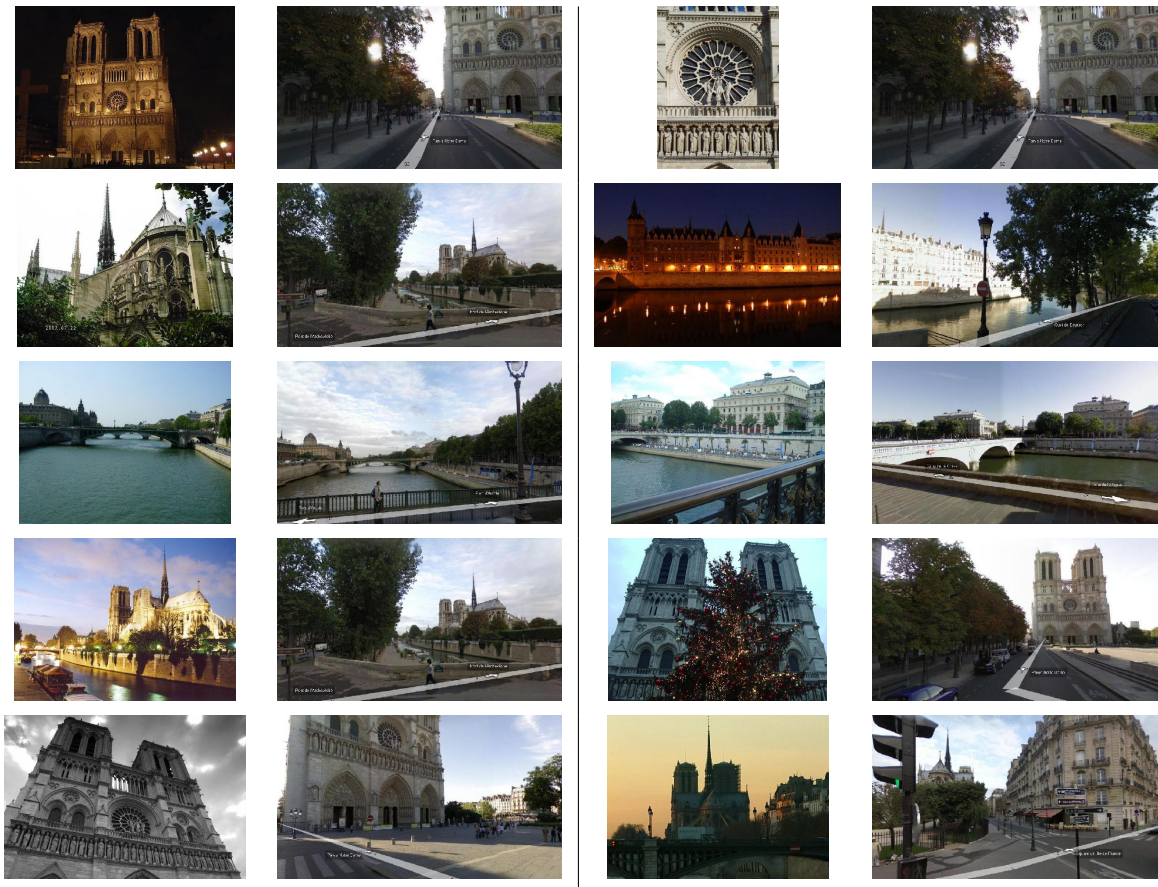


Figure 6.2: **Examples of correct location recognition results.** Each image pair shows the query image (left) and the best match from the geotagged database (right). Note that locations of query images are recognized despite significant changes in viewpoint and lighting conditions.



Figure 6.3: **Examples of not correctly localized query images,** (a) images obtained outside of our database area; (b) very challenging images; (c) ambiguous images.

are very challenging. Correct localization of these images is a difficult issue for people as well. Unfortunately, we have also found 12 out of 61 images which are easy to localize, but our algorithm did not find the correct match. These 12 query images are very similar (almost identical) to other correctly localized images. We did not observe any correct matches yet after the initial retrieval and these images do not get ranked among the 50 images after the first stage. Thus, the improvement of the image indexing looks like a way to correctly localize these 12 images. More precise verification is a complementary way for the future work as we also found several query images with a few correct tentative matches and without correctly estimated homography.

6.2.2 Verification of the Top-Ranked Image

Table 6.2 presents differences between the second stage (spatial verification which uses matches based on assignment into the visual vocabulary) and the third stage (verification of top-ranked image where the matches were computed using original feature descriptors). Both methods compute the number of inliers for the top-ranked database image and the query image. Therefore, we measure the performance in these stages as follows: (i) we compute the number of inliers between the top-ranked image and the query image; (ii) threshold τ was experimentally estimated as the number of inliers to minimize the number of the false positives while not significantly decreasing the number of true positives. We estimated τ for each method. (iii) Finally, all top-ranked images with a visual overlap to query image containing more inliers than τ were set as true positives (TP) and otherwise, top-ranked images without the visual overlap and with also the higher number of inliers than τ were set as false positive (FP).

localization method	TP	FP
a. Baseline location recognition	41	32
a'. Baseline location recognition + final. ver.	47	3
b. Query expansion	72	6
b'. Query expansion + final. ver.	75	2
c. Confuser suppression	54	6
c'. Confuser suppression + final. ver.	59	2
d. Confuser suppr.+Query exp.	86	9
d'. Confuser suppr.+Query exp. + final. ver.	85	0

Table 6.2: **Evaluation after the top-ranked image verification.** Number of correctly/incorrectly retrieved test queries decided by the comparison of number of inliers with the experimentally found threshold.



Figure 6.4: **Examples of estimation the map position for a query image.** (a) Query image, (b) the map including computed position highlighted by the callout with the geo location, street name and some google informations. Google street-view preview is shown below the map. This presents that we can obtain street name and panormatic view for the query image.

As a result, top-ranked image verification using the third localization stage (a', b', c', d') improves results significantly. We observed the important decrease of the number of FP after the third localization stage with the insignificant decrease of the number of TP, unlike when using the number of inliers based on feature quantization into a visual vocabulary (a, b, c, d in Table 6.2). Overall, (d') confuser suppr.+query expansion method with the top-ranked image verification correctly localized 85 images without any false match assigned as the positive one. This is a significant improvement compared to (d) method without final verification where the number inliers was computed only on quantized features.

6.3 Geo-location Estimation

All results were presented as the image retrieval, which has to find the most similar image obtained at the similar location. When the location of city-street images is known, the problem defined as finding the map position becomes an elementary issue. Figure 6.4 shows the query image with the top-ranked database image and its estimated position. It also allows to easily find the name of the street (asking the Google street view). In addition, position re-estimating using the epipolar geometry [HZ00] could improve final localization. Figure 6.4 illustrates the example where the correct match is found, but the position is spatially far (e.g. other side of the river).

Chapter 7

Conclusion & Future Work

The goal of this thesis is to localize the query image of a particular street or building facade using the effective representation of the images.

We have implemented and tested current state of the art bag-of-visual-word model with large vocabularies, spatial verification and modified them to solve the location recognition problem efficiently on several city-street databases containing about tens of thousands images crawled from the popular Google street-view engine and Panoramio/Flickr photo sharing site.

Several methods were experimentally evaluated with the goal to obtain the best localization performance. We have found that SURF features outperform other detectors/descriptors (MSER, Hessian, Harris, SIFT) for the task of location recognition. Higher number of clusters (visual vocabulary size) improves location recognition performance until the peak around 0.5M visual words but then continually increasing visual vocabulary size decreases the performance since descriptors from the same scene/object can be assigned into the different cluster.

In particular, we have found that suppressing confusing regions and query image expansion from collection of images significantly improved the localization. Both methods complementarity was presented as well, which addresses improving localization of the different query image types.

We plan to extend the query expansion and the suppression of confusing regions for all images. The first idea is to learn the classifier for automatic confusing regions suppression from the query image. Another future work lies in using the query expansion on all city-street images.

In spite of the being able to localize many images correctly, unlocalized query images were found as well. As the several images with the huge visual overlap were not correctly localized, we plan to improve the initial retrieval and the verification parts in future work.

Bibliography

- [BTVG06] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proc. ECCV*, 2006.
- [CbVC] Collaboration between: VGG, KUL, INRIA Rhone-Alpes and CMP. <http://www.robots.ox.ac.uk/~vgg/research/affine/>. Affine Covariant Features.
- [CM08] O. Chum and J. Matas. Web scale image clustering: Large scale discovery of spatially related images. Technical Report CTU-CMP-2008-15, Czech Technical University in Prague, 2008.
- [CMK03] O Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *DAGM*, pages 236–243, 2003.
- [CPIZ07] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *Proc. CIVR*, 2007.
- [CPS⁺07] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007.
- [Dav03] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. ICCV*, 2003.
- [Fac] Facebook. <http://www.facebook.com/>.
- [Fli] Yahoo!: Flickr. <http://www.flickr.com/>.
- [GNTvG07] T. T. Goedeemé, M. Nuttin, T. Tuytelaars, and L. van Gool. Omnidirectional vision based topological navigation. *IJCV*, 2007.
- [HE08] J. Hays and A. Efros. im2gps: estimating geographic information from a single image. In *Proc. CVPR*, 2008.

-
- [HS88] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Alvey Vision Conf.*, pages 147–151, 1988.
- [HTKP09] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *Proc. CVPR*, 2009.
- [HZ00] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
- [IM98] P. Indyk and R. Motwani. Approximate nearest neighbor-towards removing the curse of dimensionality. In *Proceedings of Symposium on Theory of Computing*, 1998.
- [KH04] R. Ke, Y. ad Sukthankar and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, 2004.
- [KS04] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. CVPR*, Jun 2004.
- [KSP09] J. Knopp, J. Sivic, and T. Pajdla. Location recognition using large vocabularies and fast spatial matching. Technical Report CTU-CMP-2009-01, Czech Technical University in Prague, 2009.
- [Lin98] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 1998.
- [LLF05] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for realtime keypoint recognition. In *Proc. CVPR*, 2005.
- [Low99] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.
- [Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [map] Google: Google maps. <http://maps.google.com/>.
- [MCUP02] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC.*, pages 384–393, 2002.

-
- [ML09] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAAP*, 2009.
- [Mor83] H. Moravec. The stanford cart and the cmu rover. In *IEEE*, pages 872–884, 1983.
- [MS01] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, 2001.
- [MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*. Springer-Verlag, 2002.
- [MS04] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 2004. submitted to PAMI.
- [MTS⁺05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005.
- [NS04] D. Nistér and F. Schaffalitzky. What do four points in two calibrated images tell us about the epipoles? In *Proc. ECCV*, LNCS. Springer-Verlag, 2004.
- [NS06] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006.
- [OM02a] S. Obdržálek and J. Matas. Local affine frames for image retrieval. In *CIVR'02: Proceedings of International Conference The Challenge of Image and Video Retrieval*, volume 1, pages 318–327, 2002.
- [OM02b] Š. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. BMVC.*, pages 113–122, 2002.
- [Pan] Google: Panoramio. <http://www.panoramio.com/>.
- [PCI⁺07] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.
- [PCI⁺08] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, 2008.

-
- [QLVG08] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *Proc. CIVR*, 2008.
- [RR04] D. Robertson and Cipolla R. An image-based system for urban navigation. In *bmvc*, 2004.
- [SBS07] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proc. CVPR*, 2007.
- [SLL01] S. Se, D. Lowe, and J. Little. Local and global localization for mobile robots using visual landmarks. In *IROS*, 2001.
- [SP02] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *IJCV*, 2002.
- [SSF⁺03] H. Shao, T. Svoboda, V. Ferrari, T Tuytelaars, and L. van Gool. Fast indexing for image retrieval based on local appearance with re-ranking. In *Proceedings of the IEEE International Conference on Image Processing*, 2003.
- [SSS06] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *Proc. ACM SIGGRAPH*, pages 835–846, 2006.
- [SSTvG03] H. Shao, T. Svoboda, T Tuytelaars, and L. van Gool. Hpat indexing for fast object/scene recognition based on local appearance. In *civr*, 2003.
- [sur] <http://www.vision.ee.ethz.ch/~surf/index.html>.
- [SZ03] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.
- [vA] Google: Street view API.
<http://code.google.com/apis/maps/documentation/examples/>.
- [vie] Google: Street view. <http://maps.google.com/help/maps/streetview/>.
- [You] Google: YouTube. <http://www.youtube.com/>.