Object Duplicate Detection

THÈSE N^{O} 5166 (2011)

PRÉSENTÉE LE 14 OCTOBRE 2011

À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

GROUPE EBRAHIMI

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Péter VAJDA

acceptée sur proposition du jury:

Prof. P. Vandergheynst, président du jury
Prof. T. Ebrahimi, directeur de thèse
Prof. M. Ansorge, rapporteur
Dr F. Dufaux, rapporteur
Prof. S. Süsstrunk, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2011

I have told you, Man: strive on, and trust!

Mondottam EMBER, küzdj és bízva bizzál!

Imre Madách (1823 — 1864)

Abstract

With the technological evolution of digital acquisition and storage technologies, millions of images and video sequences are captured every day and shared in online services. One way of exploring this huge volume of images and videos is through searching a particular object depicted in images or videos by making use of object duplicate detection. Therefore, need of research on object duplicate detection is validated by several image and video retrieval applications, such as tag propagation, augmented reality, surveillance, mobile visual search, and television statistic measurement. Object duplicate detection is detecting visually same or very similar object to a query. Input is not restricted to an image, it can be several images from an object or even it can be a video.

This dissertation describes the author's contribution to solve problems on object duplicate detection in computer vision. A novel graph-based approach is introduced for 2D and 3D object duplicate detection in still images. Graph model is used to represent the 3D spatial information of the object based on the local features extracted from training images so that an explicit and complex 3D object modeling is avoided. Therefore, improved performance can be achieved in comparison to existing methods in terms of both robustness and computational complexity. Our method is shown to be robust in detecting the same objects even when images containing the objects are taken from very different viewpoints or distances. Furthermore, we apply our object duplicate detection method to video, where the training images are added iteratively to the video sequence in order to compensate for 3D view variations, illumination changes and partial occlusions.

Finally, we show several mobile applications for object duplicate detection, such as object recognition based museum guide, money recognition or flower recognition. General object duplicate detection may fail to detection chess figures, however considering context, like chess board position and height of the chess figure, detection can be more accurate. We show that user interaction further improves image retrieval compared to pure content-based methods through a game, called Epitome.

Keywords: object duplicate detection, image analysis, mobile visual search, graph matching

Abstrakt

Die rasante Entwicklung von digitalen Aufnahme- und Speichertechnologien hat dazu geführt, dass täglich riesige Mengen von Bildern und Videos aufgenommen und durch Internetdienste verteilt werden. Um diese gewaltige Menge an visuellen Daten effizient nach interessanten Objekten durchsuchen zu können, bieten sich automatische Verfahren für die Erkennung von Objektduplikaten an. Deren Ziel ist es ein Objekt in anderen Bildern oder Videos trotz abweichender Grösse, Position, Beleuchtung oder Ansicht wiederzufinden. Neben der klassischen Bildsuche kann die Objekterkennung auch in vielen anderen Anwendungen wie Tag Propagation, Augmented Reality, Videoüberwachung eingesetzt werden.

Diese Dissertation beschreibt den wissenschaftlichen Beitrag des Autors im Bereich der Objekterkennung, der sich aus der Entwicklung eines robusten Verfahrens und dessen Nutzung in verschiedenen Anwendungen zusammensetzt. Um die Robustheit gegenüber Änderungen in Größe, Position, Ansicht, Beleuchtung und Verdeckungen zu erhöhen, wurde eine neuartiger graphenbasierter Ansatz für die Erkennung von zwei- und dreidimensionalen Objekten entwickelt. Anstelle eines expliziten dreidimensionalen Modells, wird die räumliche Anordnung lokaler Merkmale durch einen zweidimensionalen Graphen beschrieben, der anhand eines Beispielbildes gelernt wird. Im Vergleich mit anderen aktuellen Verfahren wird dadurch eine geringere Komplexität und eine erhöhte Robustheit gegenüber typischen Variationen erreicht. Für Videos wurde der Ansatz dahingehend erweitert, dass zusätzliche Beispielbilder iterativ zum Modell hinzugefügt werden, was die Robustheit gegenüber Variationen zusätzlich erhöht. Um die Vielseitigkeit des entwickelten Verfahrens zu unterstreichen, wurde es für verschiedene Anwendungen adaptiert. Dazu gehören unter anderem eine JPSearch konforme Plattform zur Verwaltung und Annotation von Bildern, eine Plattform zur semiautomatischen Geo-Tagging von Bildern und ein interaktiver Museumsführer für Mobiltelefone mit integrierter Kamera. Schlussendlich wird in Epitome die inhaltsbasierte Bildsuche durch ein interaktives Spielkonzept ergänzt was die Genauigkeit der Suche weiter verbessert.

Schlüsselworte: Objekterkennung, Bildanalyse, visuelle Suche, Graph-Matching

Acknowledgements

Finishing this thesis would have been impossible without the help, support, and encouragement of many people whom I would like to thank in the following.

First, I would like to express my gratitude to Prof. Touradj Ebrahimi for giving me the possibility to do a PhD at the MMSPG and for supporting me when it was necessary. Also I would like to thank Prof. Pierre Vandergheynst, Prof. Michael Ansorge, Prof. Sabine Süsstrunk, and Dr. Frederic Dufaux for accepting to be part of the thesis committee, for reading my thesis, and for valuable comments that helped to improve my work.

For the exceptionally friendly and relaxed atmosphere at MMSPG, I would like to thank all the people working there and having worked there.

The research behind this thesis was done in collaboration with Ivan Ivanov, Lutz Goldmann, Jong-Seok Lee, Martin Proença, Katerina Aretaki, Vuarnoz Vincent and Damien Matti, therefore I would like to thank for their contributions.

This work was supported by the Swiss National Science Foundation Grant "Multimedia Security" (number 200020-113709 and 200020-126786), and partially supported by the European Network of Excellence PetaMedia (FP7/2007-2011).

Last but not least I would like to thank my family for supporting and encouraging me during the last four years.

Contents

List of Figures

List of Tables

What does not kill me, makes me stronger.

Friedrich Nietzshe (1844 — 1900)

1 Introduction

In this Chapter, object duplicate detection problem is defined. Need of research on object duplicate detection is validated by several applications on tag propagation, augmented reality, surveillance, mobile visual search, and television statistic measurement. Finally, main contribution of this thesis is discussed briefly and the organization of this thesis is presented.

1.1 Definition of Object duplicate detection

In general, content-based image retrieval can utilize different representations for describing the image content, including global descriptors, feature points, or regions. Recently, interest has turned towards higher-level representations such as objects. Given a query image containing an object, an image retrieval system can perform two tasks: object recognition or object duplicate detection. Object recognition aims at finding all the instances of a certain object class (such as cars, or shoes), while object duplicate detection represents a more specific task of finding only a particular sample of that object class (such as "red Citroen C3 car" or "white Converse sneakers"). Figure 1.1 illustrates the relationship between object duplicate detection and object recognition problems. Therefore, within a complete system object recognition is usually applied first to detect a relevant class of objects (e.g. faces, cars) and then object duplicate detection is used to find a specific instance of that object class. The proposed object duplicate detection system, presented in this thesis, is able to fulfill both tasks together, depending on training data.

In this thesis, we are focusing on object duplicate detection task. The general goal is to detect the presence of a target object in a set of images based on an object model created from a small set of training images. Duplicate objects may vary in their perspective, have different sizes, or be modified versions of the original object after minor manipulations, which do not change their identity. Therefore, object duplicate detection should be robust to changes in position, size, view,

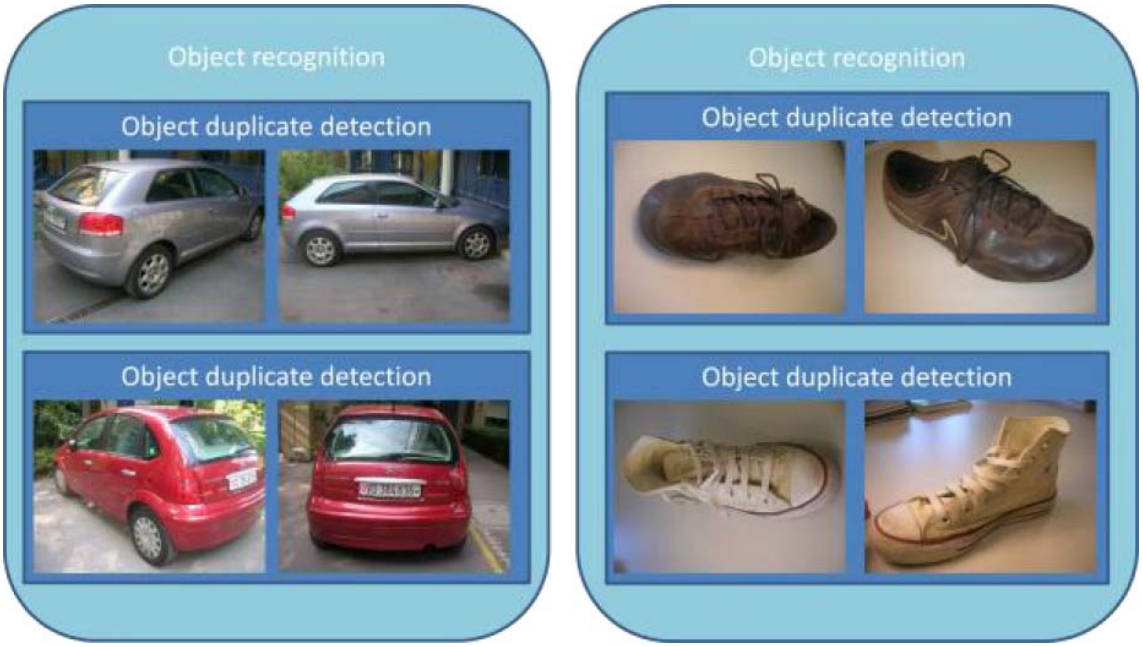


Figure 1.1 — Illustration of relationship between object recognition and object duplicate detection. While the former groups objects into different classes such as cars and shoes, the latter distinguishes between specific shoes or cars.

illumination, and partial occlusions.

1.2 Motivations

The past few years have witnessed an increasing popularity of social networks, digital photography and web-based personal image collections. A social network service typically focuses on building online communities of people who share interests and activities, or who are interested in exploring the interests and activities of others. Most social network services are web-based and provide a variety of ways for users to interact. They have become also a popular way to share and disseminate information, e.g. users upload their personal photos and share them through online communities asking other people to comment or rate their content. This has resulted in a continuously growing volume of publicly available photos, e.g. Flickr* hosts more than 5 billion photos since September 2010, and every month more than 3 billion photos are uploaded to Facebook†. Every minute, 35 hours of video are uploaded to YouTube, and 20 million videos are uploaded to Facebook every month [Pingdom, 2011].

The importance of object duplicate detection can be shown from the great interest of large companies, such as Google, Nokia, Microsoft. Moving Picture Experts Group (MPEG‡) recognized a need for standardization of visual search methods (CDVS).

*http://www.flickr.com
†http://www.facebook.com
‡http://mpeg.chiariglione.org

A large number of applications can benefit from a precise object duplicate detection. For example, when a user takes a picture of an object with his/her mobile phone, additional information about the object can be retrieved from the web, such as the price of a product, or the name and location of a monument, once the object in the picture is accurately detected and recognized.

1.2.1 Tag propagation and recommendation

As the popularity of social networking is on a constant rise, new uses for the technology are constantly being observed. To manage a large number of photos, tagging is one of the popular methods, which enables us to search our photo collections with keywords. However, tagging a lot of photos by hand is a time-consuming task. Users typically tag only a small number of the shared photos, leaving most of the other photos with incomplete metadata. This lack of metadata seriously impairs search, as photos without proper annotations are typically much harder to retrieve than correctly annotated photos. Therefore, robust and efficient algorithms for automatic tagging (or tag propagation) are desirable to help people organize and browse large collections of personal photos in a more efficient way. Automatic photo tag propagation and recommendation is capable by object duplicate detection, where ones tag an image and the system propagates or recommends tags through object duplicate detection algorithm.

In particular, object duplicate detection may be used for automatic geotagging in images. Considering the most popular tags from photo sharing sites, such as Flickr, tags are mostly related to either persons, objects, events or locations. In a large scale analysis of users' tagging behavior and the information provided by tags, Sigurbjörnsson and van Zwol [Sigurbjörnsson and van Zwol, 2008] found that 28% of the tags in a random set of 52 million photos from Flickr corresponded to the location type of WordNet [Fellbaum, 1998] categories. For a large portion of images, the association to their geographical locations provides a powerful cue for grouping and indexing. This is especially true for the large number of images depicting famous places from all over the world. Usually, the most salient region in the image corresponds to a specific landmark or object. When users annotate such images, they link a geotag to the object depicted in the image. Therefore, use of object duplicate detection for the propagation of geotags, is robust in detecting the same object and discarding similar objects. Untagged images are automatically annotated based on the detection of the same object from a small set of training images with associated tags.

1.2.2 Surveillance

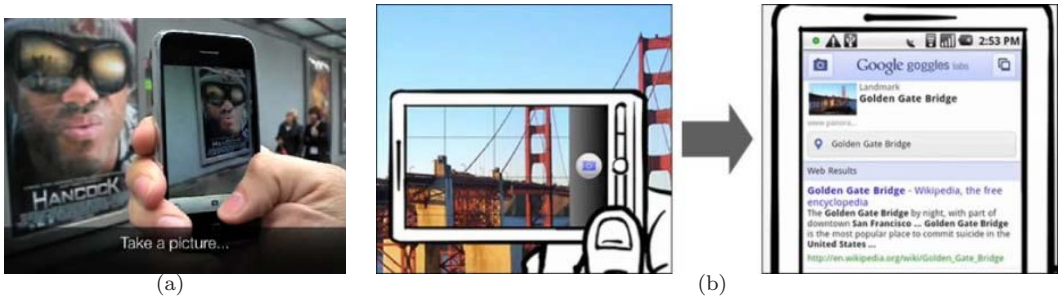
The idea of surveillance system is to assist human observers in monitoring places and events. Since human capabilities are limited, surveillance system can be extended by visual analysis. Object duplicate detection may be used to search a specific object in a large collection of image or video database, such as a suspect car in a video surveillance database. In this case, objects should be detected from any view point and at any size with certain efficiency, because the objects in training and test images may appear in different viewpoints and/or sizes. Therefore, it is important to understand the limits of multi-view object duplicate detection.

1.2.3 Television channel statistical measurement

Television ratings or statistical measurement is a system where companies measure the number of people, who are watching television shows or movies and make these statistics available for advertisers and cable networks. Statistical data collected by statistical sampling from a percentage of the customers. Aside of television channel detection, teletext, electronic program guide are recognized for further analysis. Object duplicate detection and visual search is a potential way to recognize channel from logo or teletext by similar image search comparing to a pre-determined database.

1.2.4 Mobile visual search

Mobile image search and retrieval has become increasingly popular and several commercial applications and services have been developed, including Kooaba, Google Goggles and Snaptell. *Kooaba*[*] detects specific objects, such as posters, CDs, DVDs, books, and game covers as shown in Figure 1.2 (a). *Goggles*[†] is the most recent commercial application from Google, shown in Figure 1.2 (b). It can detect logos, book covers, artworks, places and wines using visual and GPS information.



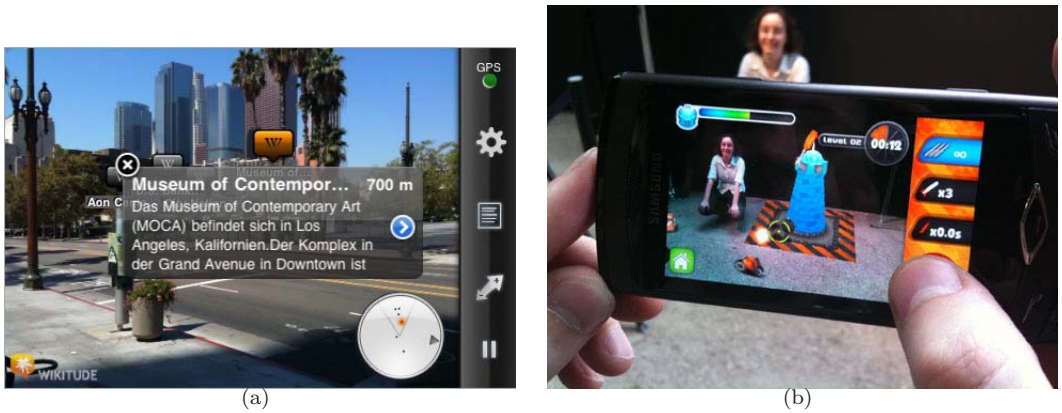
(a) (b)

Figure 1.2 — Mobile visual search by Kooaba (a) and Google Goggles (b).

1.2.5 Augmented Reality

Augmented reality is a view of a physical, real-world environment which is augmented by real-time computer-generated animation, as shown in Figure 1.3. It can enhance one's perception of reality. By the popularity of smart phones, the number of applications is increasing. *Wikitude*[‡] is a mobile application which brings Wikipedia information to view of the scene by recognizing it using GPS and other contextual information. *Augmented Reality Tower Defense*[§] for Nokia N95 is recognizing preprinted symbols to play tower defense game on mobile phone as it is shown in Figure 1.3 (b).

[*] http://www.kooaba.com/
[†] http://www.google.com/mobile/goggles/
[‡] http://www.wikitude.org
[§] http://www.ardefender.com/



(a) (b)

Figure 1.3 — Augmented reality applications: (a) Wikitude and (b) Tower defense.

1.3 Main Contributions

1.3.1 3D Object duplicate detection

A novel graph-based approach is proposed for 3D object duplicate detection in still images [Vajda *et al.*, 2009b]. This approach combines the efficiency with the accuracy, by making an attempt towards 3D modeling, while keeping the efficiency of 2D processing. A graph model is used to represent the 3D spatial information of the object based on the features extracted from the training images so that we can avoid explicitly making a complex 3D object model. Therefore, improved performance can be achieved in comparison to existing methods in terms of robustness and computational complexity. Another advantage of our method is that it requires only a small number of training images in order to build a robust model for the target object. Usually, several images from different views of an object are needed to create its 3D model. However in our approach, only a few common features are necessary to link spatial graphs from different views; therefore fewer training images are needed for the model creation. The method is evaluated through a comprehensive set of experiments, in which an in-depth analysis of its advantages and limitations is performed and optimal algorithm parameters are derived from the analysis. A comparison with the state of the art best-performing methods shows its significant performance improvement, because unlike our method, they consider a 3D object as 2D.

Further improvement achieved by generating more training images. Synthetic training images is created through automatic affine transformations as simulating the transformation of planar objects. An other way to generate more training images is to use already detected objects as new training images. Therefore objects are detected in video content iteratively in order to compensate for 3D view variations, illumination changes and partial occlusions. Given a query image with the object of interest, the proposed system retrieves key frames with duplicates of that object. Due to invariance of the object duplicate detection approach to minor appearance changes, the retrieved frames usually contain also variations from the object of interest. Therefore, the retrieved objects are considered as iterative queries to retrieve object duplicates with larger variations. For

example, given the frontal view of a car as the initial query, the iterative query mechanism can retrieve the back side of the car if intermediate views of the car are available in the video clip. The third way to increase the number of training images is to use stereoscopic images. Many cameras are provide stereoscopic image which can be considered as two separate training images. All these methods demonstrate to improve the accuracy of the original graph based object duplicate detection algorithm.

1.3.2 Swiss Cheese

As demonstration tool, Cheese*, an advanced image management platform for online and mobile use is developed for large scale database search, as shown in Figure 1.4. Beside standard features such as image upload, tagging and keyword based search, it offers the user visual similarity based search, object based tagging and semi-automatic tag propagation. For improved interoperability between different image repositories and applications, the platform supports the export and import of image files with embedded metadata in JPSearch - Part 4 compliant format.

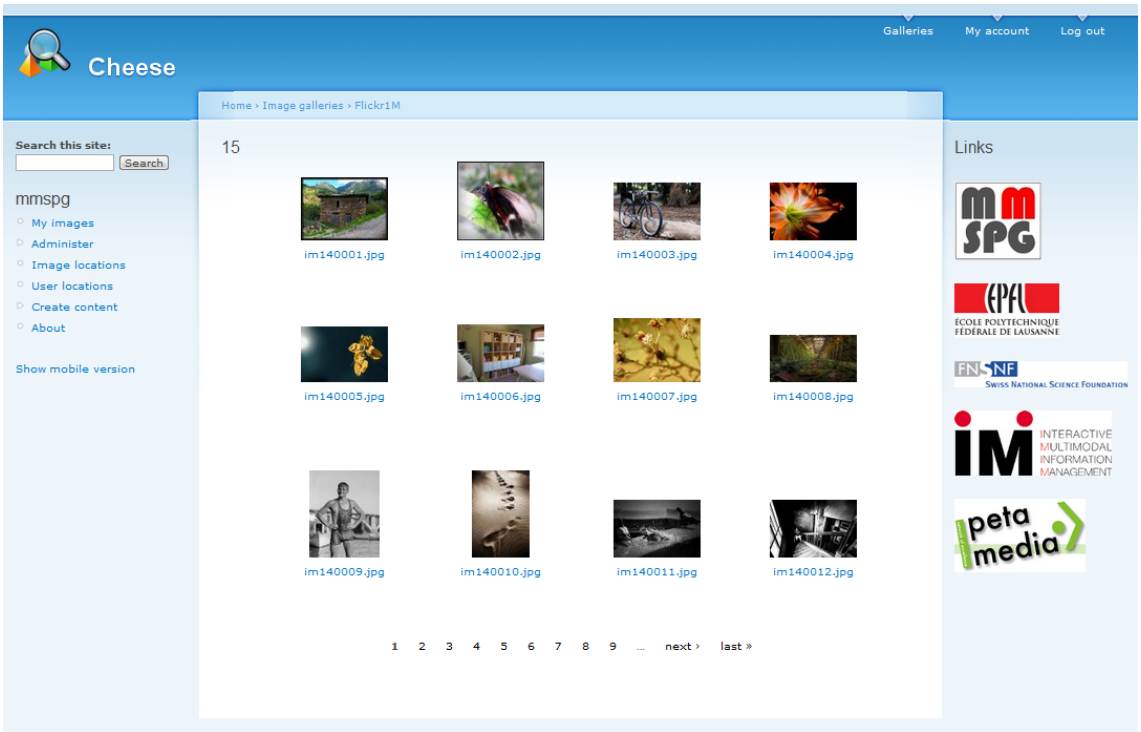


Figure 1.4 — Cheese, advanced image management platform.

1.3.3 Specific mobile applications

We developed and analyzed mobile applications for object duplicate detection to demonstrate that user interaction, context, improve the detection accuracy compare the pure content based

*http://cheese.epfl.ch/

algorithm.

Firstly, we have analyzed and described a content based applications, such as web navigation application for mobile phones or object duplicate detection based museum guide. Instead of a traditional text-based query which is quite inconvenient given the constraints of mobile phones, the query is simply formulated by capturing a photo of the object of interest. The search application will then use that photo to find similar instances of that object in a database, and provides users with associated information, such as tags, descriptions, links to web pages, audio or video material. These applications have a great capacity for gaming, education and personal usage.

Secondly, considering the environment and prior knowledge on the task, the specific object duplicate detection algorithms can be successfully applied where the general approaches are failed. Coin and chess board, figure recognition algorithms are demonstrate this principle. Priori knowledge on the content of the targets is considered in detection and recognition by taking into account the rules of chess. The chess figures recognition is based on heights comparisons through a perspective transformation. Mobile-intended application for currency exchange and counting is developed based on coin and banknote recognition using content and contextual information by taking into account the relative size information of the coins. This application has shown to be 100% effective for banknotes recognition and has proved to be almost perfect in recognizing coins by geometric considerations. Additionally, from the point of view of the user, this application has shown to be very reliable since 95% of the tested images have been recognized without any error. Above all, since all errors happen with small value coins, the average error is only of 0.02 CHF per image, with a variance of 0.008 CHF.

Thirdly, user can solve complex problems, therefore user interaction is very important to improve the accuracy of our algorithms. A novel social game "Epitome"* is proposed for photo album summarization as an Android and Facebook application, which is shown in Figure 1.5. "Epitome" is a social application, which provides many pleasant hours while playing it and enjoying photos. At the same time, it summarizes photo albums and provides useful research data. Users play with photos of their Facebook friends through two games. In these games, the user has to select either better of two photos or pair of photos that is more different. Results of these two games are integrated to produce a summarization for a Facebook photo album. These photos can be used to create a collage of an album, a cover for an album, or to be included in a photo book.

1.4 Organisation of the document

The thesis is organized as follows. In this chapter motivation of the topic and brief contribution of this thesis is presented. The next chapter is discussing the necessary background to understand the rest of the thesis. It introduces object detection in biological point of view and state-of-the-art techniques for object duplicate detection in computer vision. Then, in Chapter 3, our novel object duplicated detection algorithm is presented based on graph matching method, which capable for training from multiple images. In Chapter 4, we extend our approach for robust multi-view object recognition and for detection of objects in video. For large scale object duplicate detection,

*http://apps.facebook.com/epitome/

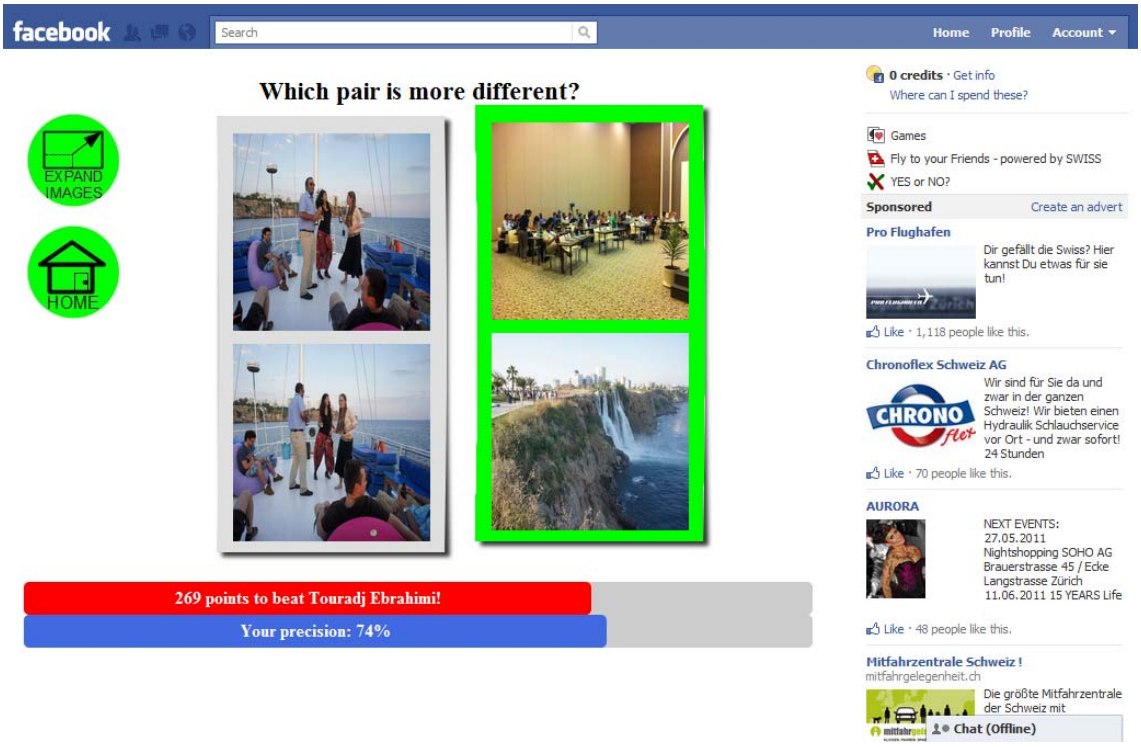


Figure 1.5 — EPITOME game for photo album summarization.

dedicated database search is necessary, which will be presented in Chapter 5. Specific applications for mobile phone are discussed in Chapter 6. Finally, we conclude the thesis with a summary in Chapter 7.

Study the past if you would define the future.

Attributed to the Greek philosopher Confucius (*cerca* 551 B.C. — 479 B.C.)

2 Background

In this Chapter we discuss the biological background of object duplicate detection algorithms, and then stat-of-the-art algorithms are presented for visual search methods. Human brains have very accurate and fast object recognition system, however computer simulation of brain is not feasible now and therefore it is necessary to approximate through efficient visual search algorithms. Visual search in image retrieval is separated by four different steps; feature extraction, similarity measurement, search in large database structures and validation. Each of these tasks is described in this Chapter.

2.1 Introduction

In this Chapter, biological and computational background for object duplicate detection is described. Human brain contains 100 billions of neurons and 100 trillions of connections between them. CPUs of computer contains 2 billions of transistor in 2011. If the integrated circuit complexity follows the Moore's Law, then the computational capacity of computers could reach the capacity of the human brain in 2022. Similarly, we will be able to store this amount of connections in our hard disk in 2022. In the near future, very interesting improvements will be done in this field.

After showing a computer approximation of the human visual system for object recognition in this Chapter, the state-of-the-art algorithms are presented, which is separated into four different steps as shown in Figure 2.1.

Vectorized description of an image or object, represented by features, is practical in computer object duplicate detection to be searchable among a large number of images, videos and objects. Features for object duplicate detection in images can be grouped into global and local features. Global features usually describe an object as a whole, while local features are extracted from

particular parts of the object, such as salient regions. The extraction of these local features usually consists of two steps. Firstly, the salient regions are detected in a way that is robust to geometric transformations, which may sensitive to corners of blob regions. Secondly, a region description for each of the detected regions is generated to make them distinguishable. We categorized them into histogram of gradient based, pixel pairwise comparison based and convolution based approaches. Due to efficient search, special database architecture is used, which could make difficult to assess spatial information or photometric information of the target object. Therefore further validation is necessary to improve the accuracy of object duplicate detection. Most common database structures are based on tree or hash algorithms.

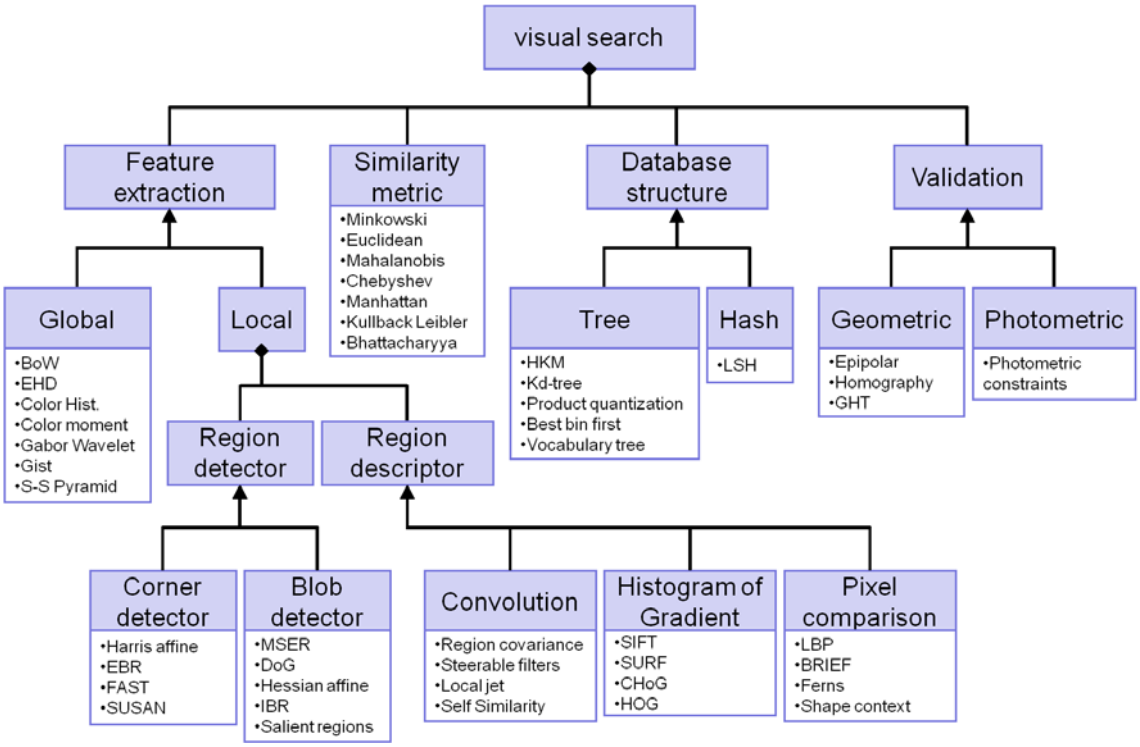


Figure 2.1 — System architecture of visual search algorithms.

2.2 Human visual object recognition

Sensory system of human is transmitting the information from outside or from our body to our brain. The visual system is the dominating sensory system, which is the most specialized among all human sensory systems.

Evolution of the eyes

The visual processing start in the eyes, therefore it is interesting to see the reason of its development. The main purpose of the development of the eye was to reach higher accuracy movement. This is

the reason why plants, which can not move, do not have eyes. The evolution of the eyes can be seen in Figure 2.2.

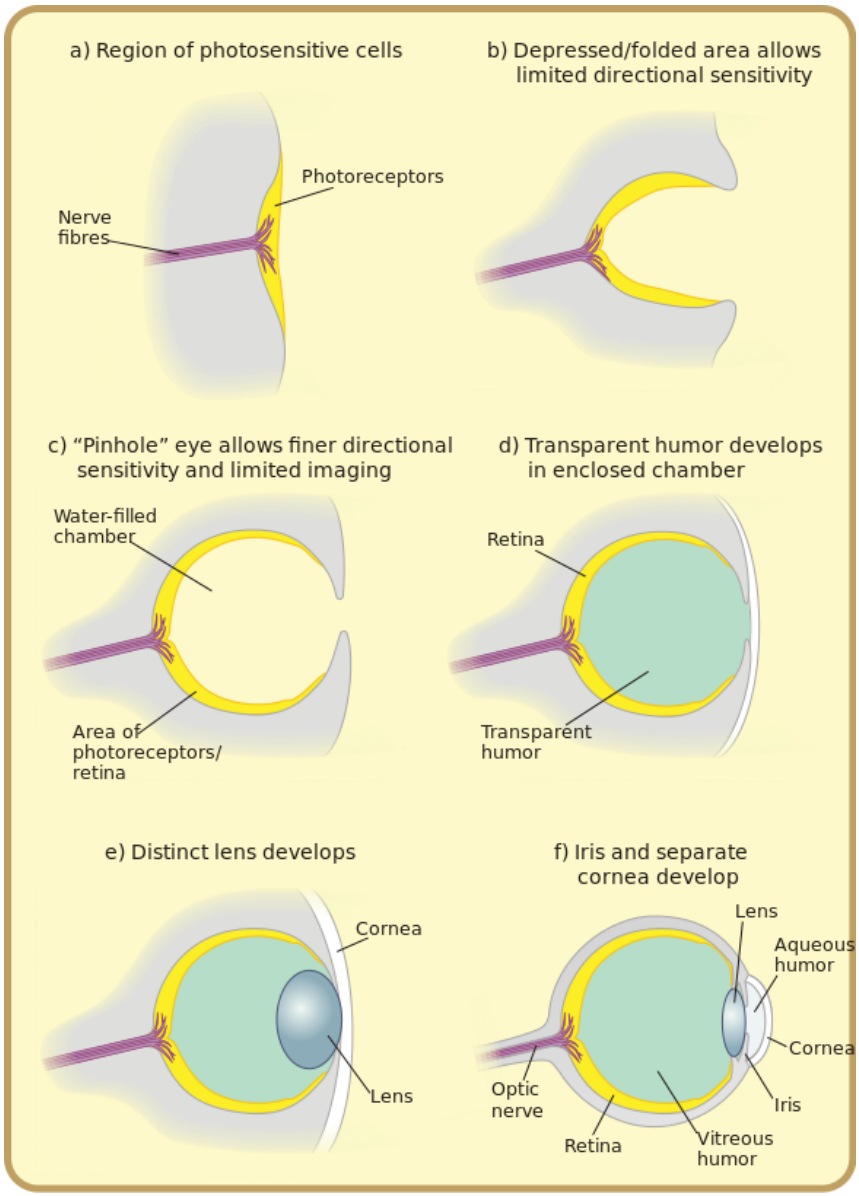


Figure 2.2 — Evolution of the eye.

Nautilus has eyes as a pinhole eye, shown in Figure 2.2 (c) and Image 2.3 (a). Its eye is not closed and it is filled with water. This eye enables the fine discrimination of light directions, but it does not allow for high contrast imaging. The Nautilus uses its eyes to navigate in coral fields. The problem appears when the water moves compare to the coral field, but the Nautilus try avoid to drift away from is. With this simple eye, the Nautilus can coordinate its movement to stay in the coral field.

Scallops has between 50 and 200 simple eyes, as shown in Image 2.3 (b). It cannot detect

shapes, however it uses his eyes to detect the velocity of animals, which allows him to predict that an animal is a prey or a predator.

(a) (b)

Figure 2.3 — Nautilus (a) has pinhole eyes, which allows him to move together with corals in the water. Scallops (b) can distinguish between preys and predators by recognizing its velocity.

Eyes of prey and predator differs in mammals, as shown in Image 2.4. Interesting to see that prey, like rabbit, has two eyes aside of their face to see the whole view of field, however predators like eagle, has his eyes next to each other, pointing to front. Therefore predators are capable for binocular, 3D vision, which can be more robust for detection, localization and segmentation of preys. Similarly, in computer vision, 3D surface can be reconstructed from stereoscope images and "fisheye" lens, similar to eyes of fish, can be used for wide-angle, panoramic and hemispherical image capturing. In Section 4.4, stereoscopic images are used for object duplicate detection.

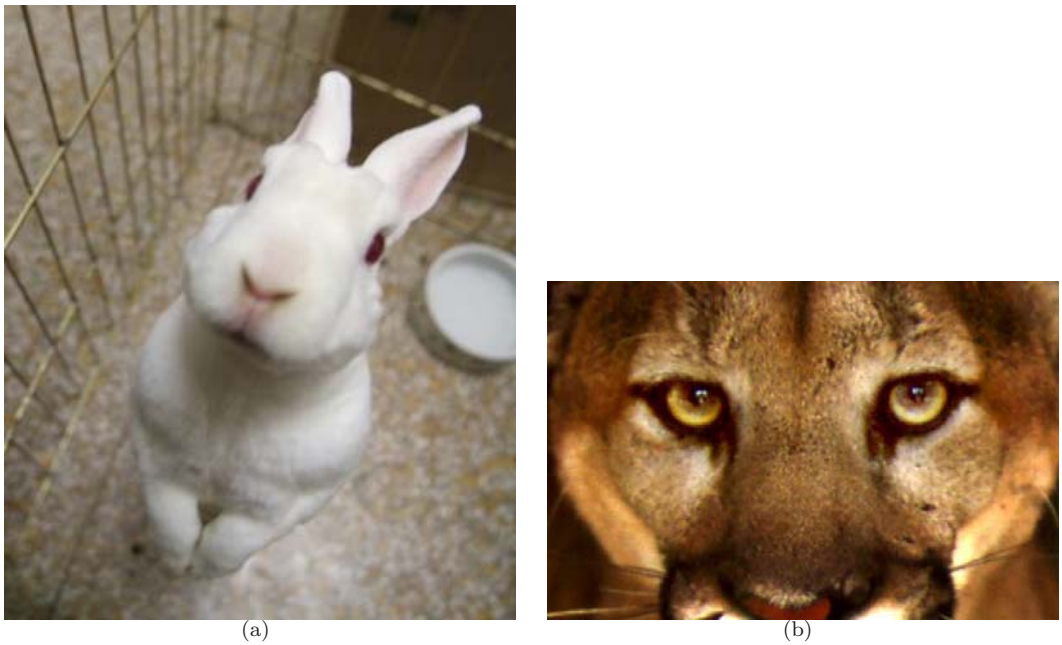
There are different specializations of eyes, for example eagle has more than one fovea to keep the focus on the prey, faster than eyes can move. To generate more contrast from the real world, some of the animal eyes can see ultraviolet light, to distinguish among specific flowers.

2.2.1 Eye

Eyes transmit very small amount of information to the brain, compare to the amount of perceived information in the eye. In computer vision, small features are extracted from image, which will be processed later. Similar to human eyes, feature can have much smaller size than the image has.

The goal of the eyeball is to focus the light ray to specific part of *neural retina*, called *Macula*, as shown in Figure 2.5. The center of the Macula is the *Fovea*. This neuron system has connection to the brain through axons of neurons, in optic nerve.

- *Cornea* is a transparent external part of the eye, where the light is transmitted through, with certain refraction.
- Next structure of the eye is the *pupil*. It changes the aperture of eyes, depending on the light condition. It is very important, because the neurons in retina can die, above a certain amount of light, and they will be not replaced, which cause permanent injury. This can happen when someone look to the sun or look into a laser beam.



(a) (b)

Figure 2.4 — Rabbit (a), as a prey, has eyes aside in his head. Lions (b) has eyes pointing to ahead, which will enable 3D perception.

- Followed by the *lens*, which allows to bring objects at different distances into focus by changeable refraction. It works as optical lens, therefore the real image, which is rendered on the retina, is upside-down. The brain changes back the orientation of the object. There are some illness attacked these parts of the eyes, for example *cataract*. It is a clouding in the lens of the eye, which results in difficulties in seeing, because light cannot go through the lens. It is a biochemical changes in the structure of the lens. Another disorder is *presbyopia*, which appear in aging. The lens loss of its flexibility, therefore it will be difficult for human to focus to close or far objects. Elder people need to wear glasses, because of this disorder.

- The *retina* is composed of multilayered neurons, located at the back of the eyeball. These neurons are part of the central nervous tissue. *Macula* (macula lutea means yellow spot) stays at the center of the retina. The name originates from the yellow pigments characterizing this area. It absorbs short wavelength light (ex. blue), because this area is responsible for high visual precision and large amount of blue light degrades visual images.

- *Fovea* is the center of macula which is responsible for the highest visual acuity and color vision.

- *Optic disc* and *optic nerve* are the axons of the retinal ganglion cells, which are the only neurons which leave the eyeball. In this area there is a blind spot, where humans cannot see, however people does not notice it, because the brain correct and fill this information automatically.

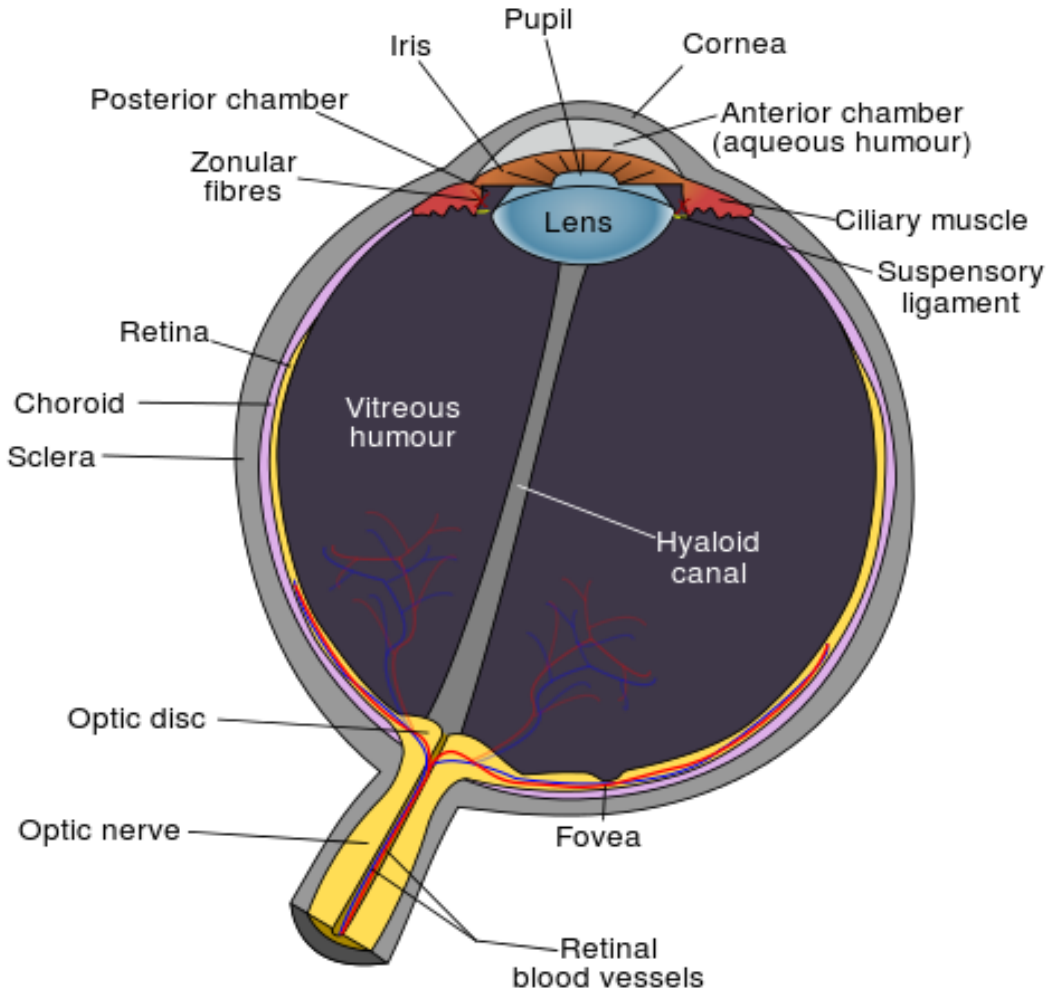


Figure 2.5 — Structure of the eyeball.

Retina

Retina contains five types of neurons as shown in Figure 2.6:

- Photoreceptor: rods and cones. Which is able to absorb light.
- Interneurons: bipolar, horizontal, amacrine cells.
- Projection neurons: retinal ganglion cells. Only neurons, whose axon leaves the eyeball.

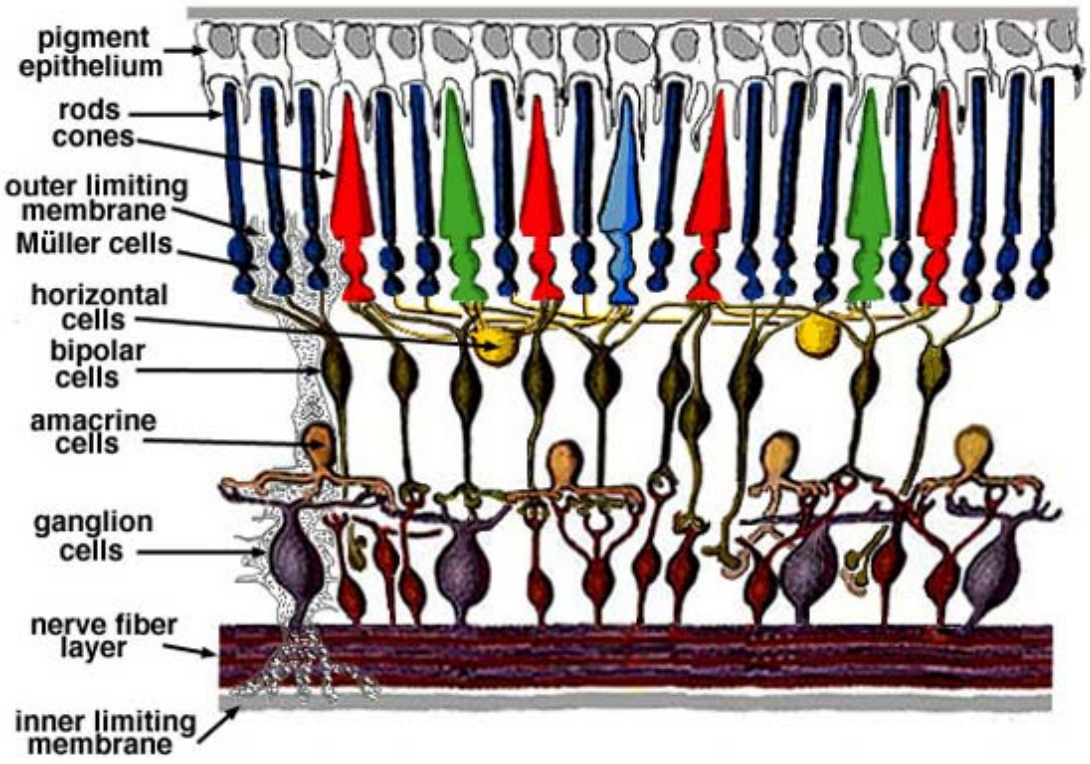


Figure 2.6 — Structure of the eyeball.

- 120 million *rods* are in the retina. They are specialized for low light conditions and they can even fire on only one photon. Most of them are in the peripheral area of the retina. A consequences is that a star looks darker in the night, if you focus on a star than if you focus next to it. Most of the features extracted form an image in computer vision, considers only the luminance of the light similar to the rods.
- There are 6 million *cones* in the retina. They are specialized for high acuity color vision. Most of them are in the fovea part of the retina. This is the reason why humans have to turn their head towards the focus point when they would like to see something clearly.

- *Bipolar neurons* are connected to photoreceptors and retinal ganglions. Cones, which are responsible for the high acuity, has one-to-one connection to bipolar neurons, however huge amount of rods, which are very sensitive to light, are connected to a bipolar neuron.
- *Amacrine and horizontal* cells enable achieving a higher contrast and they emphasize edges, thanks to the cross-linked neurons.
- *Retinal ganglion* cells are the only retinal neurons whose axons leave the eye to the brain, to the *Lateral Geniculate Nucleus* in the Thalamus and it projects the view to the *Visual Cortex*. It reconstructs the actual world and create subjective perception from them.

Retinal ganglion cells are projecting just small part of the information to the brain. There are only 1 million of ganglion cells and there are 130 millions of photoreceptor. The reason of the 100 times compression is that the brain only care about the changes in vision. In the case of Fovea, ganglions projects mainly the contrast, edges and color information, however the peripheral ganglions are sensitive only to movements. If an object moving in our visual area, first peripheral ganglions are detect the movement, and then human turns their head towards to the object to move the focus to the Fovea. Then, ganglions, which are connected to Fovea, will transmit color and contrast information of the object. Finally, the brain will identify the object considering this information. Ganglion cells project 1 billion bit information per second to the brain, which could be interpreted as $1000Hz$ frame rate data. The receptive fields of the retinal ganglion cells comprise a central approximately circular area, where light has ON effect on the firing of the cell, and an annular surround, where light has the opposite effect (OFF) on the firing of the cell, similar as Difference of Gaussian method works in computer vision as shown in Section 2.4.2.

2.2.2 Visual processing in cortex

Retinal ganglion cells project information to the cortex, where humans perceive or see. Humans do not see with their eyes, they see with their brains! Ganglions project the information by multiple parallel pathway through Lateral geniculate nucleus (LGN) to the primary visual cortex at the back of the brain.

The field, what is seen by the human, is called visual field. Only one eye can see monocular part of the visual field and both eyes can see the binocular part of visual field. Image, which was presented to the left side of eyes, is projected to the right side of the brain and right side of the visual field is projected to the left hemisphere, as you can see in Figure 2.7.

Therefore, LGN receives information from both eyes, but the projection is kept separate in LGN in different layers. Also the specific information, such as color, motion, is processed in different layers. This projection is point to point map, however more tissue is related to the Fovea and color, shape information than to the peripheral retina.

In primary visual cortex the eyes, color, motion information is still kept separately. There are two basic streams information. Dorsal stream (top part of the brain), also called "Where pathway", is associated with motion, representation of object locations, and control of the eyes and arms, especially when visual information is used to eye movement or reaching [Goodale and Milner, 1992], as it is shown in Figure 2.8. Ventral stream (bottom part of the brain) also, called

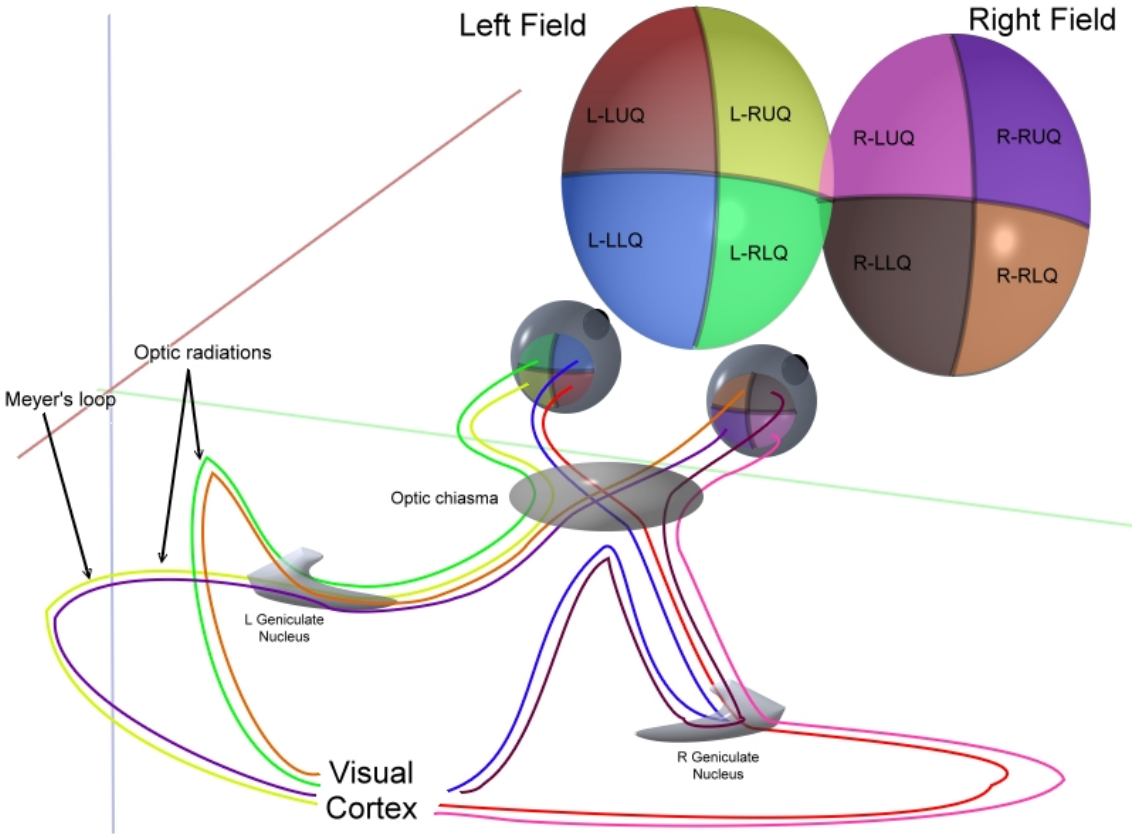


Figure 2.7 — Illustration of the optic tract.

the "What pathway", is associated with object recognition, representation and with storage of long-term memory. These areas are specialized, higher order areas. For example, part of this area process only the velocity of an object or another part does only face recognition; moreover there are areas of the brain which does frontal face recognition and other areas which does facial recognition from side view. 30 of these areas in the brain are processing only visual information. By inspiration of this mechanism, we also did research on specific object recognition as described in Chapter 6. The result showed significant improvement compare to general object duplicate detection algorithms.

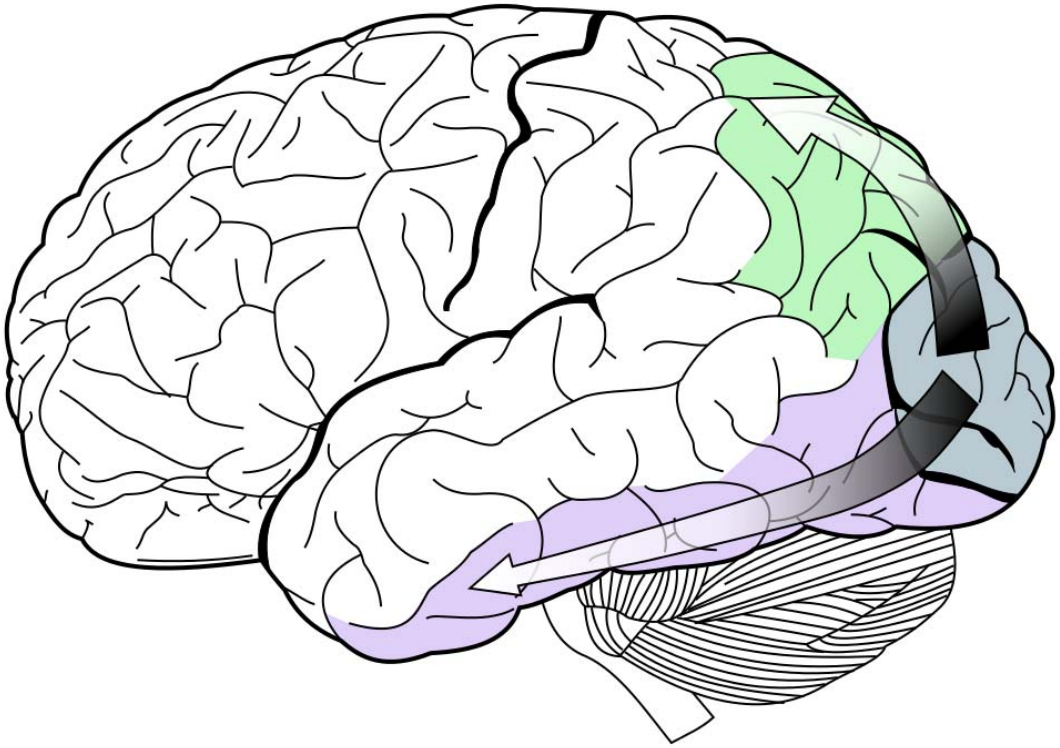


Figure 2.8 — Dorsal (green) and Ventral (purple) streams.

V4 is next to the primary visual cortex, which process the color visual information. In human color vision, there are three types of cones: red, green and blue cones. Distributions of these sensors are not the same, there are specific filter in the eyes for low wavelength of light, which absorb part of the blue light. Also there are no blue cones in the Fovea; however brain can predict the blue color respect to the other color information in this area.

Why humans see color? Brain processes the changes and color enhances contrasts in natural images. 500 gray levels and 6 million hues of color can be distinguished, which lead better segmentation of the view. It is shown in computer vision, that color based features are perform the best compare to other types of features. However, usually researchers discard the color information in object duplicate detection due to the light sensitivity of these algorithms.

Visual illusion

An visual illusion is characterized by visually perceived images that differ from objective reality. Analysis of visual illusions is one of the way to discover how our visual perception works, which can be modeled by computer vision to improve the detection.

Visual illusions are constructed in the brain. Most of the illusions can be explained with Gestalt law. There are five principle in Gestalt law: emergence, reification, multi-stability, invariance and grouping [Sternberg, 2002; Woodward and Cohen, 1991].

- *Energebce* is the process, when complex pattern formation from simpler rules. An examples can be seen in Figure 2.9 (a). The object is not recognized by identifying its parts (ears, nose, etc.), instead, it is perceived as a whole, all at once.

- *Reification* is a constructive illusion. In Kanizsa triangle, as shown in Figure 2.11 (a), a white triangle is perceived, but in fact none is drawn. The illusionary contours can be explained with Gestalt theory. Gestalt effect represents the form generating capability of our senses. Visual system, recognize figures and whole forms, instead of just a collection of simple lines and curves.

- *Multi-stability* or figure-ground organization illusion can be seen in Figure 2.9 (b). The image as a whole switches back and forth from being the women then being the lady. In this image the retinal information is same, however one can interpret this image as an old women or a beautiful young lady.

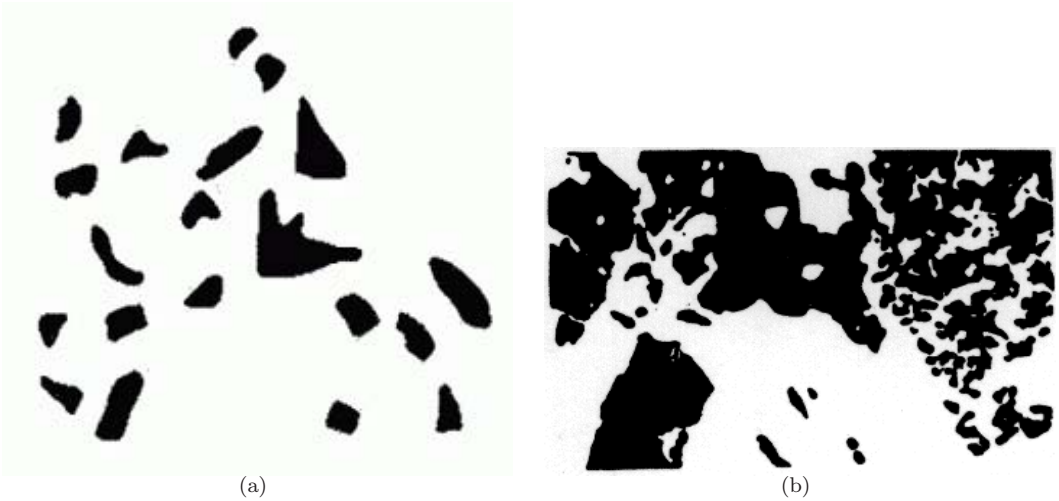
- *Invariance* is the property where objects can be recognized independent from rotation, translation, and scale, as well as several other variations such as elastic deformations, different lighting, etc.

- *Grouping* is the process, where humans tend to order our experience in a manner that is regular, orderly, symmetric, and simple. Further categorization made in grouping law as follows:

  - *Closure* is the process to complete a regular image.
  - *Similarity* is the process, when similar elements are grouped into collective entities depend on relationship of form, color, size or brightness.
  - *Proximity* is the law, where spatial or temporal proximity of elements are grouped into collective entities.
  - *Symmetry* is the process, where symmetrical images are perceived collectively.
  - *Continuity* is the process, where the mind continues visual, auditory, and kinetic patterns.
  - *Common Fate* is the law, where elements which are moving in the same direction, are perceived collectively.

(a) (b)

Figure 2.9 — Example of visual illusion.

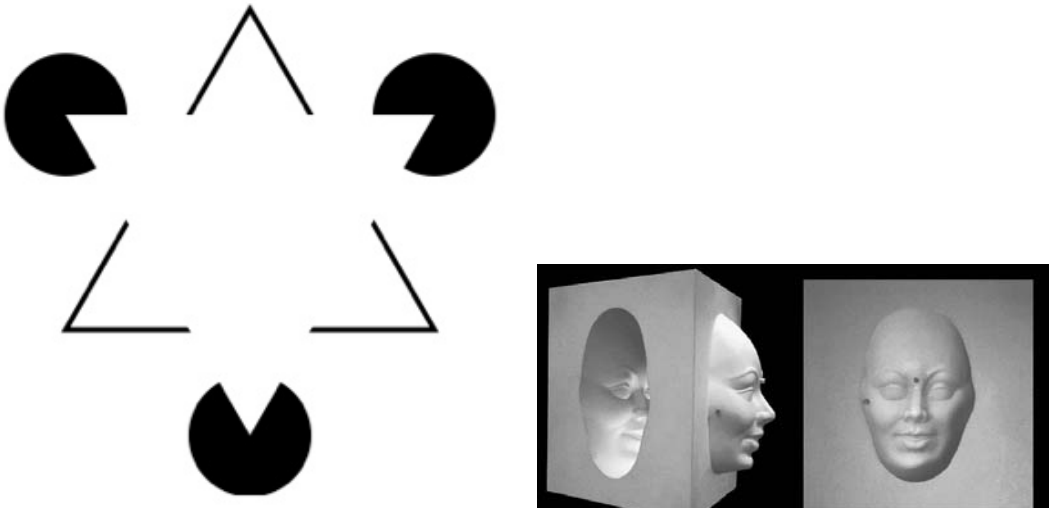


(a) (b)

Figure 2.10 — Example of visual feedback from memory.

Another example for visual illusion is the blind spot in the view, caused by the hole in the retina,where the sight nerves come out of the eyeball. Humans do not recognize it in their perception, because the brain fills this hole with predicted information.

[Gregory, 1997] claims that knowledge representation combined with low level brain processing tools can lead to a more accurate object recognition. In Chapter 6, we developed chess and money recognition algorithm using prior knowledge information. The results are significantly better than for general object duplicate detection. The Hollow-Face illusion is an visual illusion in which the perception of a concave mask of a face appears as a normal convex face. Our perceptual system reconstructs the face this way, relying on the a priori knowledge of the face as shown in Figure 2.11 (b). This shows that the ventral stream of visual cortex is two way stream. From the memory, information is sent back to the lower level of visual system. Two examples can be seen in Figure 2.10. People who first see these images, cannot recognize them, however if they know that the Figure (a) shows a horse rider and Figure (b) shows portrait of Jesus, then they will recognize these figures for the rest of their life. These feedback is not well developed for children under 6 years, so they cannot recognize these photos.



(a) (b)

Figure 2.11 — Example of visual illusion.

2.2.3 3D perception

There are several cues for 3D perception as shown in Figure 2.12. While humans eyes scan the scene, called saccadic movement, 3D information can be retrieved from eye movements. For example accommodation, which represents the focus of the eye or convergence of the eyes, by rotating the eyeballs to gaze the object as shown in Figure 2.13. Object has to be close, maximum $5 - 10m$ to perceive 3D information in this way.

Human eyes are directed to ahead and their visual fields are overlapping by an approximate

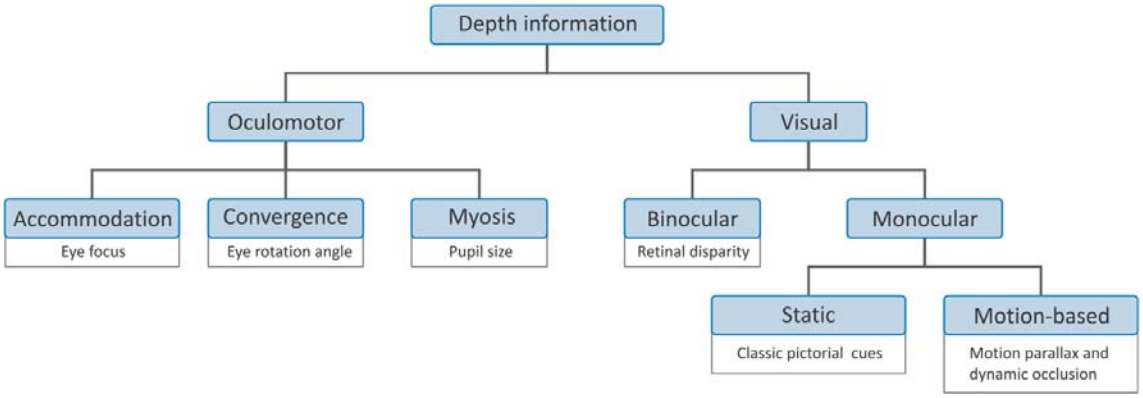


Figure 2.12 — Classification of 3D perception cues.

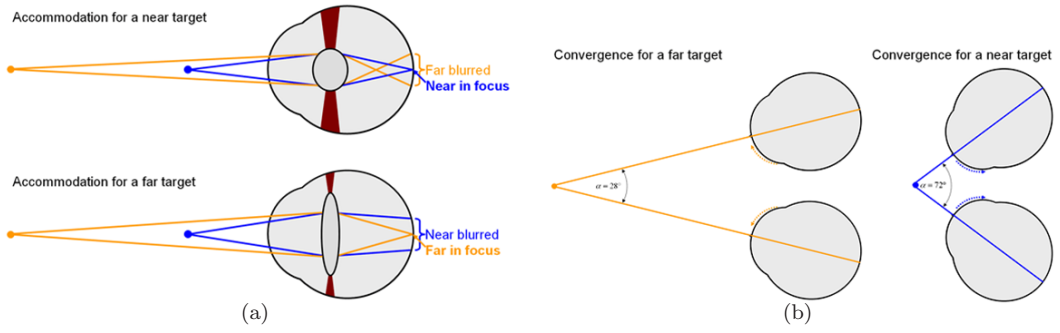


Figure 2.13 — Accommodation eye movement can be seen in (a) and convergence of the eyes in (b)

angle of 120°. This leads to binocular disparity, which provides cues about the relative depth of objects. It is efficient for close objects, and it is possible to perceive depth with this method closer than $50 - 100m$. An illustration can be found in Figure 2.14 (a). 12% of people has problem to see 3D by binocular disparity.

3D perception from monocular view is the most important in human vision and it can retrieve depth information from far objects too. Motion parallax based on apparent displacement or difference in the apparent position of an object viewed from different views during the human movement. Pictorial cues, however mostly based on object recognition as explained in Section 2.2.2. Prior knowledge from the object can fill or enhance information from the object, such as depth or distance, computed from the size as shown in Figure 2.14 (b).

(a) (b)

Figure 2.14 — Stereoscopic view can be seen in (a) and street drawing in (b)

2.3 Bio-inspired object recognition

Human visual processing is very accurate and efficient. Therefore it is obvious to copy some idea from human visual processing and use it in computer vision algorithms. In this Section two object recognition algorithms are briefly described, which were highly adapted from human vision processing. Firstly, convolutional network method is presented inspired on brain analysis, then HMAX algorithm is described based on this approach for object recognition in computer vision.

2.3.1 Convolutional network

Convolutional Neural Networks are multi-layer neural networks. They are trained by using a special structured back-propagation algorithm. Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal preprocessing, and they operate similarly to the retina and to the visual cortex. They recognize patterns with extreme variability, and with

robustness to distortions and simple geometric transformations. It is applied for face [Lawrence *et al.*, 1997], object and letter recognition [Lecun *et al.*, 1998; Simard *et al.*, 2003] with success.

The Convolutional Neural Network consists of several layers of neurons, each of which could contain more neural planes. These layers are sequentially connected when considering their spatial position on each plane. So, the input of each neuron is coming from the neighbors of its plane from previous layer, as shown in Figure 2.15. The input of the whole network is usually low level images, without preprocessing. Local filters are the examples of this connectivity between the layers, such as Gaussian of Laplacian or a Gaussian filter. Moreover any translation invariant linear operator is a convolution.

The idea of connecting layer to local neighbors in different layers comes from discovery of locally sensitive, orientation-selective neurons in the cat's visual system, presented in [David H. Hubel, 1965]

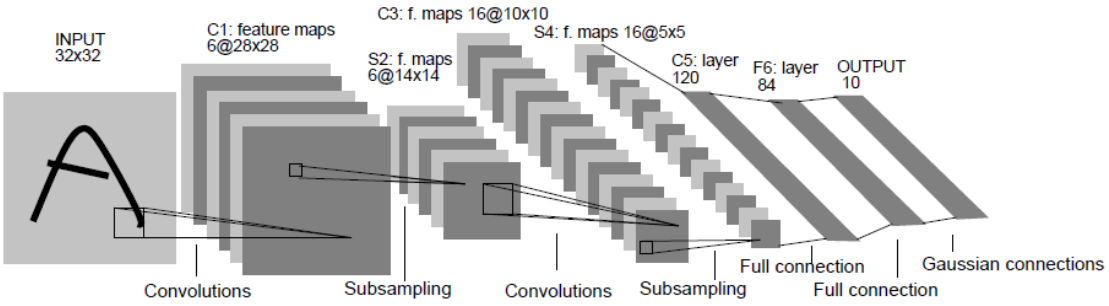


Figure 2.15 — Convolution network for letter recognition [Lecun *et al.*, 1998].

2.3.2 HMAX

In this Section a bio-inspired object recognition algorithm is presented based on convolutional network. Object recognition in cortex is processed by ventral visual pathway, which starts with primary visual cortex, V1, over V2, V4 to inferotemporal cortex (IT). Object recognition is postulated to play central role in object recognition. A basic quantized model is set for computer simulations. This model is called the standard model for visual recognition [Riesenhuber and Poggio, 1999]. This model is hierarchical, which combines simple filters into more complex ones as shown in Figure 2.15. Article [David H. Hubel, 1965] describe, that primary visual cortex consist two cells, simple and complex cells. Max operation is applied between simple and complex cells, and weighted sum is calculated to more complex features in simple layer, meanwhile the different orientation and positions are kept in separate nodes. Brief architecture follows [Serre and Poggio, 2010; Serre and Riesenhuber, 2004; Serre *et al.*, 2005a] and illustration is shown in Figure 2.16:

S1 Layer : Input images, $128x128$ grey-scale pixel images are densely sampled by two-dimensional oriented Gaussian filters into four direction. At each pixel of the input image, filters of several size and orientations are applied.

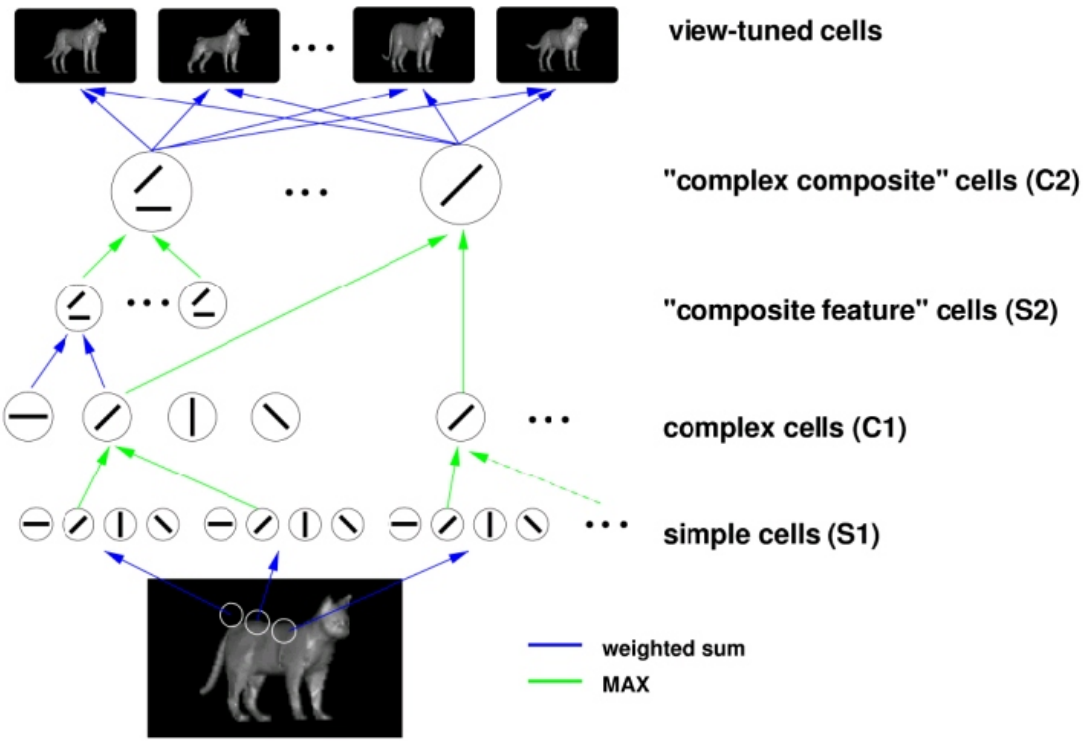


Figure 2.16 — Illustration of HMAX, bio-inspired object recognition algorithm in different stages [Riesenhuber and Poggio, 1999].

C1 Layer : Only S1 filters with the same preferred orientation feed into a given C1 unit. Max operator is then applied on neighbors, which is determined by the strongest input it receives.

S2 Layer : Square of four continues, non-overlapping C1 units are grouped and normalized. Therefore, there are $4^4 = 256$ different S2 units.

C2 Layer : Shift and rotation invariance are achieved by applying max operation in S2 layer [Wiskott, 2001]. There are 256 C2 units, each of which receives input from all S2 units of one type at all positions and scales.

VTU Layer : Viewtuned units (VTUs) provide the result for view specific object recognition of a 3D object. It is also shown that similar behavior of neurons is existing in monkey brain[Logothetis *et al.*, 1995]. The training process applied in this layer, where the target view is set for input in the S1 level and each neuron in this layer tuned by selecting the activities of the 256 C2 units in response to that stimulus as the center of a 256-dimensional Gaussian response function.

Learning phase from multiple examples, such as different view-tuned neurons, leads to view-invariant units. Inputs are views and the outputs are the label of the object or its position.

2.4 Feature extraction

Vectorized description of an image or object, represented by features, is practical in computer object duplicate detection to be searchable among a large number of images, videos and objects. Features for object duplicate detection in images can be grouped into global and local features. Global features usually describe an object as a whole, while local features are extracted from particular parts of the object, such as salient regions. The extraction of these local features usually consists of two steps. Firstly, the salient regions are detected with methods which are robust to geometric transformations but remain sensitive to corners of blob regions. Secondly, a region description for each of the detected regions is generated to make them distinguishable. We categorized them into histogram of gradient based, pairwise pixel comparison based and convolution based approaches, as shown in Figure 2.1.

Most important properties of features are the followings Tuytelaars and Mikolajczyk [2007]:

- *Robust to occlusion* and few local features are enough to recognize the whole object.
- *Invariant* to scale, orientation and translation for robust detection.
- *Robust* to noise, blur, discrimination, compression, light condition.
- *Distinctive* enough to accurately compare to a large dataset of features.
- *Large quantity* is also necessary to detect even small object too.
- *Accurate* and precise localization.
- *Efficient* in time to use in real-time application.

Brief descriptions of popular features in the state-of-the-art are presented in the following.

2.4.1 Global features

Four types of global features are distinguished in this Section: local based features (BoW, Spatial and Scale Pyramid), edge features (EHD), color features (Color histogram, Color Moment) and texture features (Gabor wavelet, Gist) as described in follows:

Bag of Words (BoW) [Fei-Fei and Perona, 2005] model in computer vision was derived from BoW model in natural language processing (NLP) . The BoW in NLP is a popular method for representing documents, which ignores the order of words. Each document is considered as a "bag", which contains some words from a dictionary without considering the order of them. Documents in computer vision represent images or objects, and visual clusters of local features are considered as words. Thus, BoW is a vector which represents the histogram of visual features. Brief description can be found in Section 2.4.7.

Spatial and Scale Pyramid [Lazebnik *et al.*, 2006] is an approximate global geometric correspondence method. This technique partitions the image into increasingly fine sub-regions and compute histograms of local features found inside each sub-region.

Edge histograms (EHD) [Park *et al.*, 2000] describes edge distribution with a histogram based on orientation of local edge distribution in an image.

Color histogram descriptors in the MPEG-7 standard include a histogram descriptor that is coded using the Haar transform, a color structure histogram, a dominant color descriptor, and a color layout descriptor [Manjunath *et al.*, 2001]. Color descriptors often fail in image retrieval in different light conditions.

Color Moment [Stricker and Orengo, 1995] denotes the color distribution by using mean, standard deviation, and the third root of the skewness of each color channel.

Gabor wavelet transform [Zhu *et al.*, 2008] is applied to a rescaled version of the image down to 64×64. Different levels and orientations are used, respectively 5 and 8, yielding in 40 sub images. Then mean, variance and skewness moments are computed for each subimage, finally producing a 120-dimensional vector.

Gist descriptor [Douze *et al.*, 2009] is computed on an image by dividing it into a 4-by-4 grid for which orientation histograms are extracted.

2.4.2 Region detector

Two types of region detectors are described in this Section. (Harris affine - SUSAN detector) and Blob detector which recognize stable regions in images.

Edge and corner detector, which recognize discriminative points on edges of an image:

Harris affine key point detector [Mikolajczyk and Schmid, 2002] is an iterative algorithm. Firstly, combination of Gaussian scale-space images and the Harris corner detector Harris and Stephens [1988] is applied on the image, called Harris-Laplace detector. Then, iterative key point refinement is applied for each scale and location.

Edge-Based Region detector (EBR) is presented in [Tuytelaars and Van Gool, 1999, 2004]. Edges are stable features, which can be detected over different viewpoints, scales and/or illumination changes. Harris corner points are detected and nearby edges are extracted by the Canny edge detector from multiple scaled images. A corner with two points on the edge defines a parallelogram as its three corners. These two points move away from the corner in both directions along the edge. These two points therefore were chosen to be affine invariant and based on the maxima or minima of the intensity of this parallelogram, the stable region is detected, as shown in Figure 2.17.

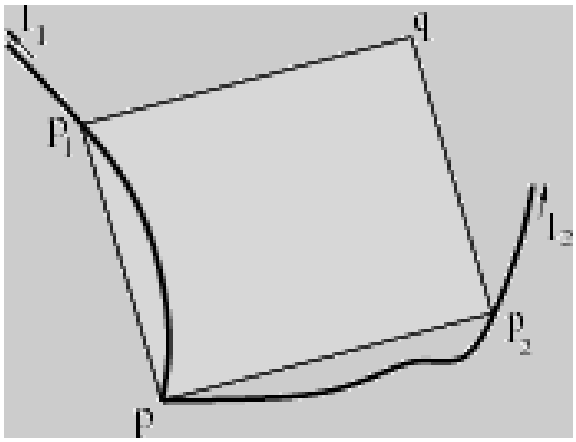


Figure 2.17 — Edge-Based Region detector.

Feature from Accelerated Segment Test (FAST) [Rosten and Drummond, 2006] detects 16 pixels in a circle around each point and determines if n consecutive points in this circle are brighter, darker then or similar to that of the center point. The algorithm selects the pixel, which yields the most information about whether the candidate pixel is a corner, by calculating its entropy. A recursive algorithm generates a decision tree which can correctly classify all corners in real-time.

Small Univalue Segment Assimilating Nucleus (SUSAN) [Smith and Brady, 1997] computes self similarity by looking at the part of the pixels inside a disc whose intensity is similar to the center, called nucleus value. Pixels, closer in value to the nucleus, receive higher weighting. A pixel having a low weighting value is considered as a corner. Local maxima are chosen as regions.

Blob detector recognize stable regions in images.

Maximally stable extremal regions (MSER) [Matas *et al.*, 2002] is a blob detection algorithm. All the pixels below a given threshold become an extremal region. These regions are maximally stable if they are local minima or maxima in a certain distance on the threshold.

Difference of Gaussians (DOG) [Enroth-Cugell and Robson, 1966] algorithm is similar how neural processing in the retina extracts features from images. Local minima or maxima of

DOG represent the descriptor point as applied in Scale-invariant feature transform (SIFT) features [Lowe, 1999]. Illustration is shown in Figure 2.18.

Hessian affine detector [Mikolajczyk and Schmid, 2002] relies on interest points detected at multiple scales using the Harris corner measure on the second-moment matrix of Gaussian images.

Intensity extrema-based region detector (IBR) [Tuytelaars and Gool, 2000; Tuytelaars and Van Gool, 2004] detects intensity extrema in multiple scales of an image and examines the image from this point in a radial way. Intensity extrema is detected in these radial lines and selected as a border of the region.

Salient Regions detector is described in [Kadir *et al.*, 2004]. From each pixel, all parameterized ellipsis is calculated. From the intensity of this area, a probability density function (PDF) and its entropy are determined. Key-points are selected by calculating the extremal values of these entropies across different parameters of the ellipsis. The saliency value is computed by the magnitude of the derivative of the PDF with respect to scale.

2.4.3 Region descriptor

Here we list some representative techniques for region descriptor.

Convolution based feature evaluate image patches using specific filters do describe the region of interest.

Region covariance feature [Tuzel *et al.*, 2006] is the covariance of several features, e.g., the three-dimensional color vector, the norm of the first and second derivatives of intensity along the horizontal and vertical directions, etc. These features characterize regions of interests.

Steerable filters [Freeman and Adelson, 1991] allow to synthesize filters as region descriptor, of arbitrary orientation from linear combinations of basis filters, allowing to adaptively "steer" a filter to any orientation, and to determine analytically the filter output as a function of orientation.

Local jet feature [Manzanera, 2010] is based on Taylor expansion of image patches to create compact representation of image patches.

Self similarity [Shechtman and Irani, 2007] correlates the image patch centered at each point in the image, with a larger surrounding image region having a radius of 40 pixels, resulting in a local internal correlation surface. The correlation surface is then transformed into a binned log-polar representation as a feature vector.

Histogram of Gradient algorithms are based on gradient information of the image, which are robust for illumination change.

Scale-invariant feature transform (SIFT), proposed in [Lowe, 1999], is invariant to scaling, rotation, translation. It shares similar properties with neurons in inferior temporal cortex that are used for object recognition in primate vision [Serre *et al.*, 2005b]. Histogram of

gradient is extracted on detected DOG regions. After normalization, 128 byte feature vector is generated and further feature compression technique (PCA) may be applied as described in Section 2.4.4.

Speeded-Up Robust Features (SURF) is a scale- and rotation invariant detector and descriptor [Bay *et al.*, 2006b]. It is a fast approximation of the SIFT algorithm using Haar features with integral images. The detection of interest points is performed using an approximation of the determinant of the Hessian matrix by 2D Haar-like features. These approximations and use of integral images are the reasons why this algorithm is much faster than its predecessors. Integral images drastically speed up the computation of the sum of elements inside any rectangle. If the sum of elements in the upper-left part of an image is pre-computed for each pixel, the sum of elements inside any rectangle at any location in the image can be computed with just one addition and two subtractions. Brief description can be found in Section 2.4.5

Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005] is similar to SIFT and SURF region descriptors, calculates the histogram of gradients in the region around the one keypoint. It is evaluated on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. Using gradient information for feature description is very robust to different illumination conditions. HOG description extraction explained briefly in Section 2.4.6.

Compressed Histogram Of Gradients (CHOG) [Chandrasekhar *et al.*, 2009] is a low bit-rate feature descriptors. It represents a gradient histogram in tree structures which can be efficiently compressed. This algorithm is applied for mobile visual search to reduce the bandwidth of communication between a mobile and server.

Pixel comparison algorithms are extremely fast and simple methods for region descriptor.

Local Binary Patterns (LBP) are proposed in [Ojala *et al.*, 1994]. The examined region is divided into 16×16 non overlapping pixels cells and each pixel in a cell is compared to each of its 8 neighbors in a clockwise order, which gives an 8-digit binary number. The histogram over the cell based on these numbers is computed. The feature will be the concatenation of these histograms. Recently, it is shown that this algorithm, combined with HOG method, is robust for human recognition [Wang *et al.*, 2009].

Binary Robust Independent Elementary Features (BRIEF) method [Calonder *et al.*, 2010] performs a relatively small number of pairwise binary intensity comparison of pixels and composes a binary feature vector from the results.

Ferns [Özuysal *et al.*, 2009] feature extraction is a trained naive Bayesian classification on pairwise binary intensity comparison of pixels. Image patches evaluated on these classifiers is generating the feature itself.

Shape context feature [Belongie *et al.*, 2002] at a reference point on and edge captures the distribution of the remaining edge points relative to this reference point in a log-polar histogram.

2.4.4 Scale-invariant feature transform - SIFT

Among the most commonly used region descriptors is the Scale Invariant Feature Transform (SIFT), proposed in [Lowe, 1999], which is invariant to scaling, rotation, translation. It shares similar properties with neurons in inferior temporal cortex that are used for object recognition in primate vision [Serre *et al.*, 2005b]. Algorithm can be divided into four parts [Lowe, 2004]:

- *Scale-space extrema detection*: Keypoints are detected by calculating the local maxima or minima of Difference of Gaussians (DOG as seen in Figure 2.18) occur in consecutive scale-space image:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.1}$$

  where k is scaling factor and scale-space image, $L(x, y, \sigma)$, is defined as convolution $(*)$ of Gaussian, $G(x, y, \sigma)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.2}$$

  where I is the original image.

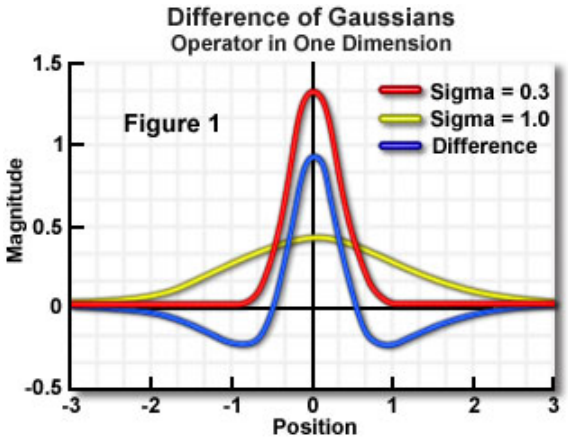


Figure 2.18 — Difference Gaussians.

- *Keypoint localization*: In this step, poorly localized keypoints along edges and low contrast keypoints are eliminated. Precise location is calculated and low contrast keypoints are discarded by Taylor expansion, calculated till the second derivative on the discreet DOG. Poorly localized keypoint, along the edges are discarded by using the eigenvalues ratio of the DOGs Hessian matrix [Harris and Stephens, 1988].

- *Orientation assignment*: Gradient magnitude and orientation for each pixel in the keypoint region is calculated from discrete approximation of the scale-space image. Then histogram is generated, using 36 orientation bins for every 10°. Local peaks of orientations are assigned to the keypoint.

- *Keypoint descriptor* Gradient magnitude and orientation are calculated at each keypoint region, as shown on the left on Figure 2.19. These gradients are weighted by a Gaussian

window. $4x4$ subregions of histogram is created using 8 bins of orientation, as shown on the right of the Figure.

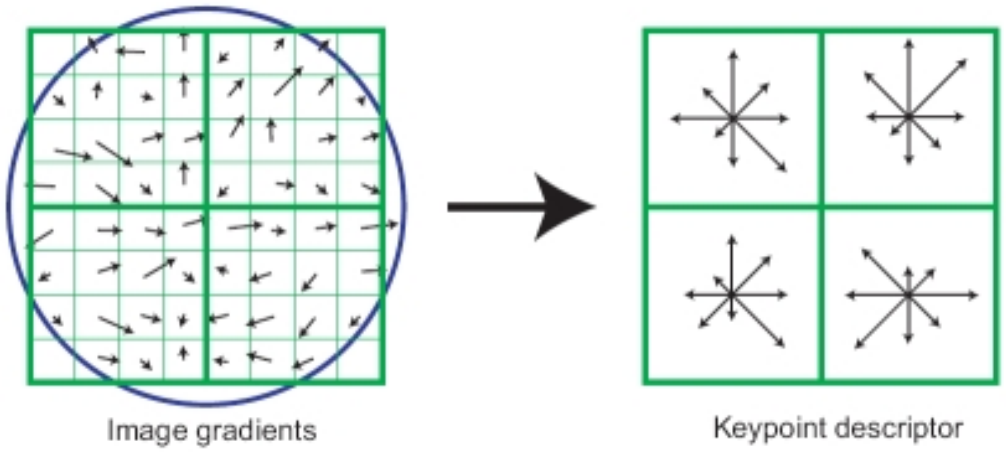


Figure 2.19 — Scale-invariant feature transform. This figure shows a $2x2$ descriptor array computed from an $8x8$ set of samples, whereas the algorithm use $4x4$ descriptors computed from a $16x16$ sample array.

2.4.5 Speeded-Up Robust Features - SURF

Speeded-Up Robust Features (SURF) is a scale- and rotation invariant detector and descriptor often used for object recognition tasks [Bay *et al.*, 2006b]. It is a fast approximation of SIFT.

The detection of interest points is performed using an approximation of determinant of Hessian by 2D Haar-like features, as it is illustrated in Figure 2.21. These approximations and use of integral images are the reasons why this algorithm is much faster than its predecessors. Integral images drastically speed up the computation of the sum of elements inside a rectangle. If the sum of elements in the upper-left part of an image is pre-computed for each pixel, the sum of elements inside any rectangle at any location in the image can be computed with just one addition and two subtractions.

$$I_{integral} = \sum_{i \leq x, j \leq y} I(i,j) \tag{2.3}$$

$$D = (A+B+C+D) - (A+B) - (A+C) + A \tag{2.4}$$

$$D = I_{integral}(4) - I_{integral}(2) - I_{integral}(3) + I_{integral}(1) \tag{2.5}$$

where I is the original image, $I_{integral}$ is the integral image and we are looking for sum of the D area as shown in Figure 2.20.

Haar-like features are approximated second order partial derivatives of Gaussian, which are used at several scales for the detection of blobs of various sizes. Their localization is performed

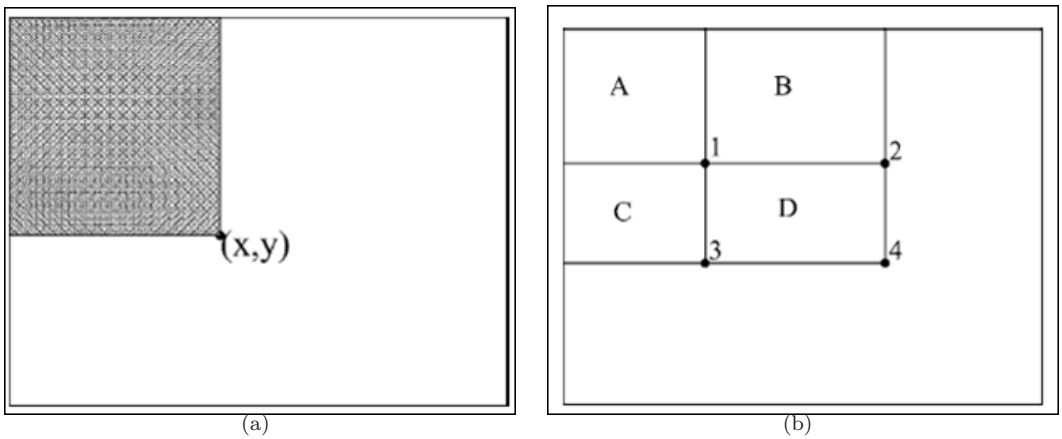


Figure 2.20 — Sum of values in a rectangle is calculated in constant time. Integral image calculation is shown on (a), where each pixel contains the value of the sum of all pixel values in the shaded rectangle of the original image. The sum of pixel values in rectangle D is calculated using integral image as shown on (b).

by detection of maximums of determinant of the Hessian, as expressed hereunder, across scale and space.

$$\det(H(\mathbf{x},\sigma)) = L_{xx}(\mathbf{x},\sigma)L_{yy}(\mathbf{x},\sigma) - L_{xy}(\mathbf{x},\sigma)L_{yx}(\mathbf{x},\sigma) \tag{2.6}$$

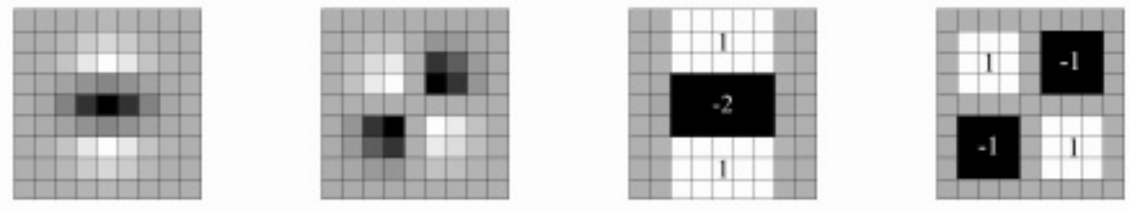


Figure 2.21 — Partial derivatives L_{yy} and L_{xy} and their Haar-like approximations.

In order to describe those interest points, SURF uses the sum of Haar wavelet responses in the x and y directions. After determining the dominant orientation of those responses in order to be invariant to image rotation, the interest region is split up into smaller 4×4 square sub-regions. For each of those squares, Haar wavelet responses d_x and d_y are computed again and summed up (generating sub-region descriptors $\mathbf{v}_i$), then concatenated in one region descriptor $\mathbf{V}$ of size $16 \times 4 = 64$ (number of sub-regions $\times$ dimension of the sub-region descriptor), as illustrated in Figure 2.22. Again, this process is speeded up by the use of integral images. Those descriptors can then be matched with those of the database to perform recognition.

$$\mathbf{v}_i = \left(\sum d_{xi}, \sum d_{y_i}, \sum |d_x|_i, \sum |d_y|_i\right) \quad (2.7)$$

$$\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{15}, \mathbf{v}_{16}) \quad (2.8)$$

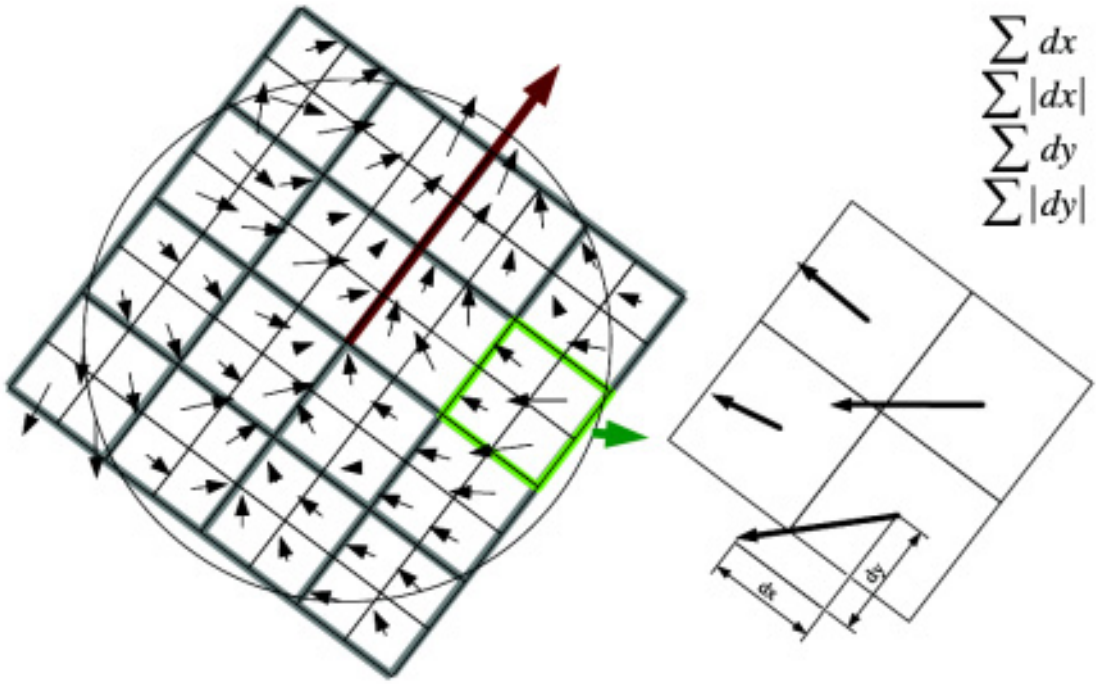


Figure 2.22 — Division of the interest region into sub-regions.

2.4.6 Histogram of Oriented Gradients - HOG

Similar to SIFT and SURF region descriptor, Histogram of Oriented Gradients (HOG) calculate the histogram of gradient in the region around the keypoint [Dalal and Triggs, 2005]. However it is evaluated on dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. HOG description extraction contains the following steps:

- *Gradient computation*: Simply 1-D centered discrete derivative mask is applied in one or both of the horizontal and vertical directions: $[1, 0, -1] and [1, 0, -1]^T$.

- *Orientation*: Unsigned gradients used in conjunction with 9 histogram channels performed best in person detection experiments.

- *Descriptor blocks*: Gradients are normalized locally, in order to account for changes in illumination and contrast. The HOG descriptor is the vector of the normalized cell histograms

from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once. There are two blocks exists: Rectangular R-HOG blocks and circular C-HOG blocks. For example, R-HOG blocks are shown in Figure 2.23. 3x3 cell blocks of 6x6 pixel cells with 9 histogram bins were chosen. Blocks are computed in dense grids at some single scale without orientation alignment, which make this feature more descriptive, however it is not invariant for orientation.

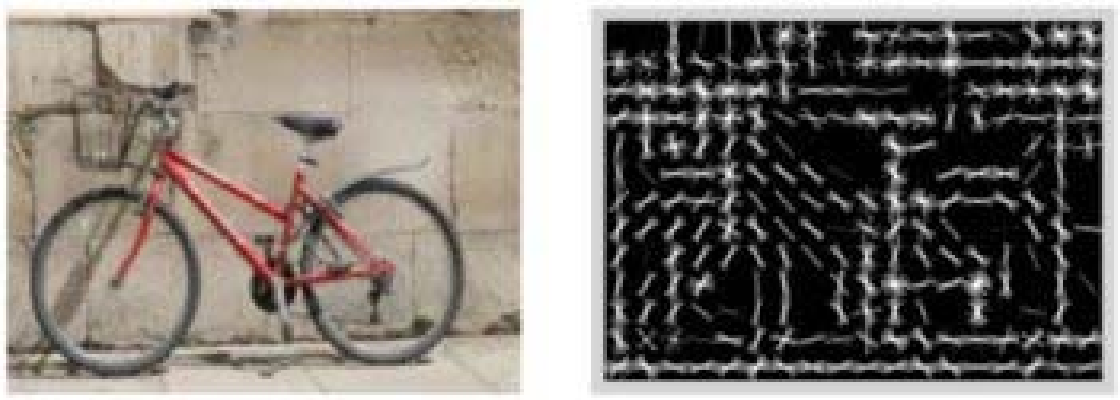


Figure 2.23 — Example of Rectangular Histogram of Oriented Gradients.

- *Block normalization*: Different block normalization, such as L2 is applied to improve the recognition task.

2.4.7 Bag of Words - BoW

Bag of Words (BoW) model in computer vision was derived from BoW model in natural language processing (NLP) [Fei-Fei and Perona, 2005]. The BoW in NLP is a popular method for representing documents, which ignores the order of words. Each document looks like a "bag", which contains some words from a dictionary without considering the order of them; illustration is shown in Figure 2.24. Similar method in computer vision documents represents images or objects, and visual clusters of local features are considered as a word. Thus, BoW is a vector which represents the histogram of visual features.

2.5 Similarity

Measureing similarity or distance is important part of visual search methods, which is related to feature extraction part. Measurement between histograms or numbers may different, however we can define a common distance function, as follows: Distance function or metric is a function which defines a distance between elements of a set (X).

$$d : X \times X \to \mathbf{R} \tag{2.9}$$

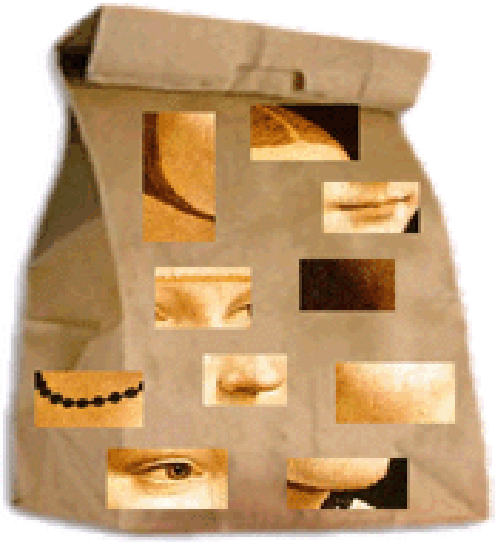


Figure 2.24 — Illustration of BoW method.

d required to satisfy the following conditions:

$$d(p,q) \geq 0 \tag{2.10}$$
$$d(p,q) = 0 \iff x = y \tag{2.11}$$
$$d(p,q) = d(q,p) \tag{2.12}$$
$$d(p,r) \leq d(p,q) + d(q,r) \tag{2.13}$$
$$\forall p,q,r \in X \tag{2.14}$$

The following known distance measures are recalled, by noticing that Kullback Leibler divergence is not true metric, because it is not necessarily fulfill the symmetric formal condition stated in Expressions 2.10 to 2.14.

Minkowski distance is a generalization that unifies Euclidean distance ($p = 2$), Manhattan distance ($p = 1$), and Chebyshev distance ($p \to \infty$).

$$d(p,q) = \Big(\sum_{i=1}^{n} (q_i - p_i)^p \Big)^{1/p} \tag{2.15}$$

Euclidean distance is the most common used distance.

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2.16}$$

Mahalanobis distance is normalizes based on a covariance matrix to make the distance metric

scale-invariant.

$$d(p,q) = \sqrt{(q-p)^T S^{-1}(q-p)} \tag{2.17}$$

where S is the covariance matrix.

Chebyshev distance measures distance assuming only the most significant dimension is relevant.

$$d(p,q) = \max_{i=1}^{n} |q_i - p_i| \tag{2.18}$$

Manhattan distance measures distance following only axis-aligned directions.

$$d(p,q) = \sum_{i=1}^{n} |q_i - p_i| \tag{2.19}$$

Kullback Leibler divergence is non-symmetric measure of the difference between two probability distributions. Commonly used for histogram comparison.

$$d(p,q) = \sum_{i=1}^{n} p_i \frac{p_i}{q_i} \tag{2.20}$$

Bhattacharyya distance measures the amount of overlap between probability density functions.

$$d(p,q) = -ln\Big(\sum_{i=1}^{n} \sqrt{q_i \cdot p_i}\Big) \tag{2.21}$$

2.6 Data structures for search

Data structure is a particular way of storing and organizing data for efficient search.

Hierarchical K-means (HKM) defines hierarchical quantization of the feature space [Ivanov *et al.*, 2010b]. K-means algorithm is used to split the training data into a certain number of groups. Then, this clustering process is recursively applied to the groups from the previous level until a maximum depth is reached as shown in Figure 2.25. It allows for logarithmic search in large database; however k-means clustering are slow and often not balanced.

Local Sensitive Hash (LSH) [Indyk and Motwani, 1998] a randomized hashing technique using hash functions that map similar points to the same bin, with high probability as illustrated in Figure 2.26. The main advantages are fast training and fast search, however it is difficult to find accurate hash functions for this approach.

Product quantization search [Jégou *et al.*, 2011] decomposes the space into a Cartesian product of low dimensional subspaces and quantize each subspace separately.

Kd-tree [Bentley, 1975] is a special case of binary space partitioning trees (BSP) [Fuchs *et al.*, 1980]. The kd-tree is a binary tree in which every node is a k-dimensional point. In case of Figure 2.27 k is 2. Every non-leaf node can be thought of as implicitly generating a

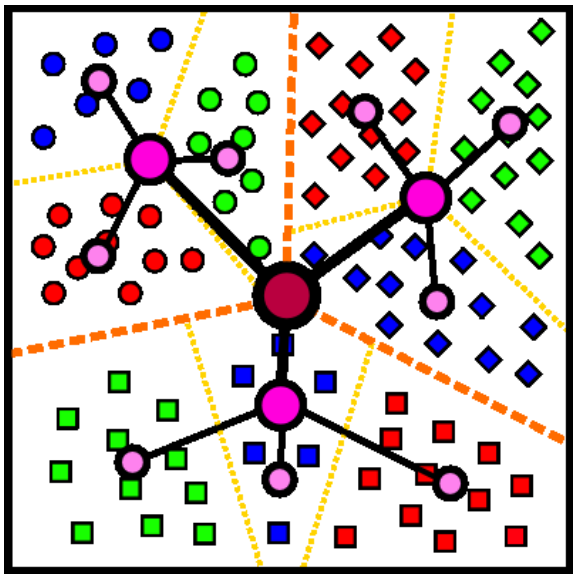


Figure 2.25 — Hierarchical K-means.

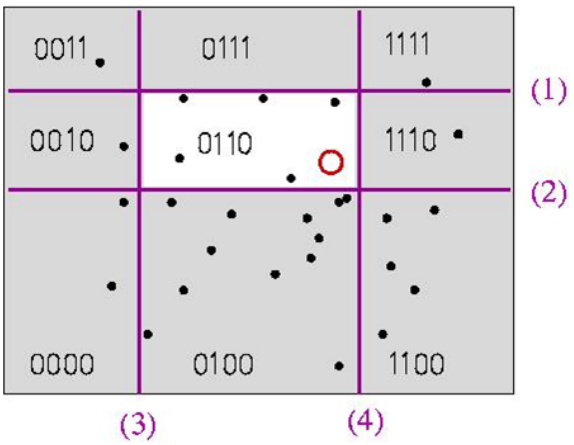


Figure 2.26 — Local Sensitive Hash illustrated with hamming hash function.

splitting hyperplane along a dimension that divides the space into two subparts as shown in Figure 2.27 with horizontal and vertical separation lines. This method creates a binary tree.

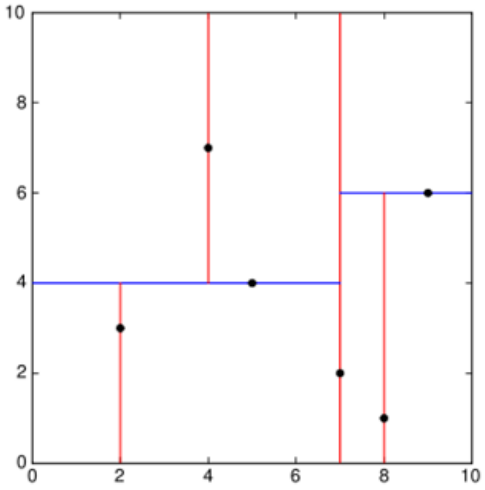


Figure 2.27 — Kd-tree database search structure is illustrated.

Best bin first [Beis and Lowe, 1997] a variant of kd-trees that uses priority queue to examine most promising branches first and uses backtracking algorithm for further search. The computation complexity for search is logarithmic to the number of elements of the database.

Vocabulary tree [Nister and Stewenius, 2006] is derived by applying hierarchical k-means clustering to group the features according to their similarity. Then, a fast approximation of the nearest neighbor search is used in the feature matching step. Within the tree, a parent node corresponds to the cluster centers derived from the features of all its children node. The clustering leads to a balanced tree with a similar depth for all the leaves. Each leaf node corresponds to several images in the database which have similar features. Therefore if a query feature is searched through the vocabulary tree, several images refereed to it, similar as the inverted files. Finally each of the features in the query image are voting for images in the database which results in similarity value as illustrated in Figure 2.28.

2.7 Geometric and photometric validation

Due to efficient search, special database architecture and special features are applied, which could make difficult to assess spatial information of the target object. Therefore further validation is necessary to improve the accuracy of object duplicate detection.

Epipolar geometry [Hartley and Zisserman, 2004] is the geometry of stereo vision with assumption that the cameras can be approximated by the pinhole camera model, where camera

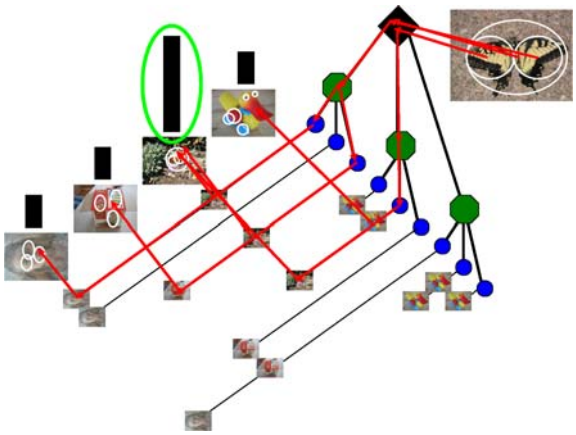


Figure 2.28 — Vocabulary tree and matching method is represented.

aperture is described as a point and no lenses are used to focus light. The Figure2.29 depicts two cameras looking at point X. O_L and O_R represent the focal points of the two cameras, points X_L and X_R are the projections of point X onto the image planes. Here we can define epipolar points, line and plane as represent constraints of the point's position in both views. Epipolar points are e_l and e_r which are the projected points from the other focal point to the image plane. Epipolar plane is the plane defined by the two focal points and the target point X. Epipolar plane projected to the image plane construct a line, which is the epipolar line. Each point which projected into one point X_l in the left view is an epipolar line in the other view. This constraint can be use for validation of object duplicate detection using Ransac method as described in Section 2.7.1.

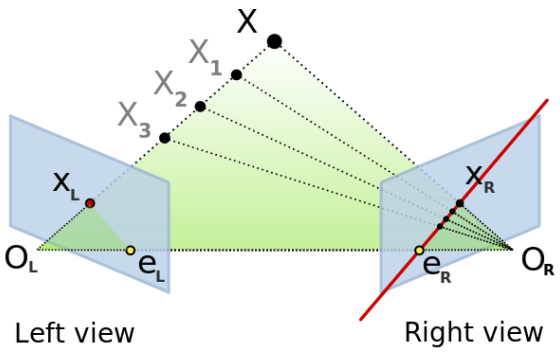


Figure 2.29 — Epipolar geometry represented with epipolar line, plane and points.

Homography [Schmid and Zisserman, 1998][Hartley and Zisserman, 2004] is perspective transformation between real 3D view plane to image plane and vice-versa, with assumption that the cameras can be approximated by the pinhole camera model. More details can be found in Section 2.7.3. This constraint can be use for validation of object duplicate detection using Ransac method as described in Section 2.7.1.

General Hough Transformation (GHT) [Ballard, 1981] is shape detection and localization method. In general objects, edges are detected end each of the points on the edges are voting for the parameters of object, such as center of the object, size and orientation, considering each point on the training object. Example can be found in Section 2.7.2.

Photometric constraints [Tuytelaars and Gool, 2000] calculates moments of the intensities of image regions, and determines the linear transformations in both images regions. The constraints allow only an overall scale factor difference.

2.7.1 RANdom SAmple Consensus - RANSAC

RANdom SAmple Consensus (RANSAC) [Fischler and Bolles, 1981] is an optimization algorithm used to find iteratively the best model fitting some data, as well as the best parameters fitting that model. Given a set of observed data, some of them are considered as *inliers* (that is data that fit the model) while others are considered as *outliers*, which do not fit to the model, as illustrated in Figure 2.30.

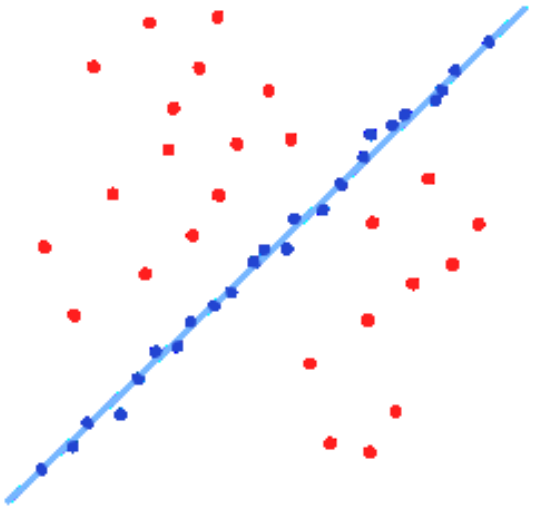


Figure 2.30 — RNASAC method to find the best model fitting some data. The line describes the best model. Inliers are in blue, outliers in red.

If N is the minimal number of data needed to fit the model, the main idea of this algorithm is to take randomly N data among the entire set and to build a model based on those N data. Then RANSAC checks how many of the remaining data fit that specific model. If the number of data fitting to the current model is large enough, i.e. larger than some threshold, the model becomes a candidate as the best model. Then, it is tested for the entire set of data and global error is computed, which describes how the model fits the entire set. This process is then repeated with N other randomly-chosen data. The best model is the one that minimizes the global error.

2.7.2 Hough transform

The Hough transform is a method used to find parameterized curves or shapes in an image, such as lines, circles, ellipses, or general point sets Ballard [1981]. In this description we consider the Hough line transform for simplicity; however it can be generalized using different parameters.

The main concept of this method is to use a new parameterization to express the equation of a line, and to use those parameters for a voting process. In the case of a line, the usual parameterization is done with the Cartesian coordinates x and y. But using ρ as the minimal distance to the origin and θ as the angle formed by the normal of the line and the x-axis as shown in Figure 2.31, it is also possible to parameterize a line as follows:

$$\rho = x \cos\theta + y \sin\theta \tag{2.22}$$

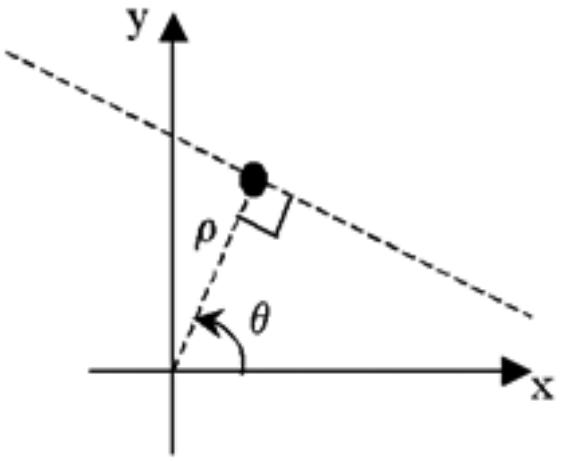


Figure 2.31 — Hough line parameters.

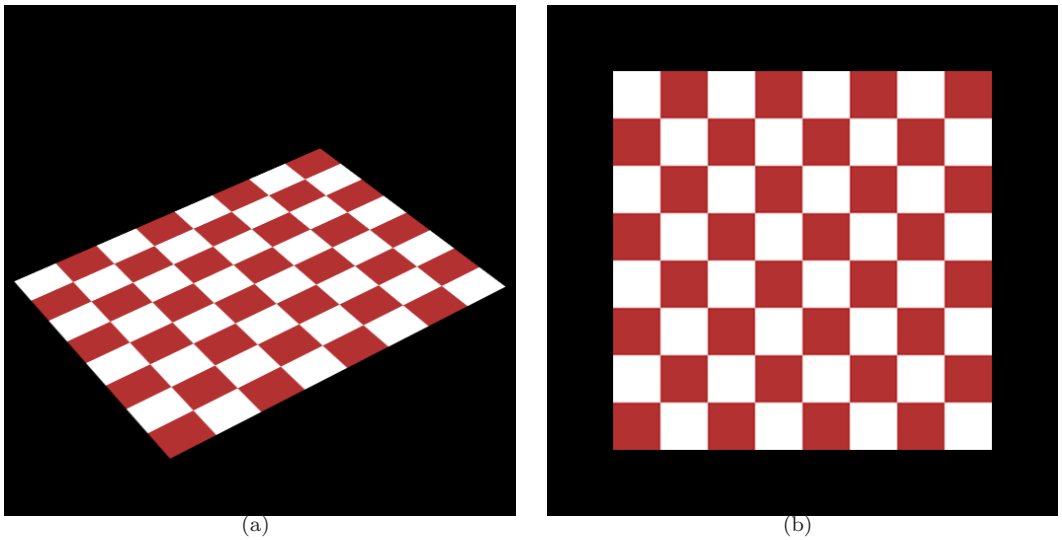
Using this parameterization, for each point located on an edge, one considers a set of possible lines going through that point, each of them having different θ (i.e. different directions). For each of those lines, their minimal distance to the origin (i.e. the value of ρ) is computed using the normal line. Hence, with N lines going through one single point, one obtains N pairs (ρ, θ), which correspond to a point in the *accumulator plane*. Consequently, each possible line "votes" for one point in the accumulator plane.

Repeating this same process for all edge points, one can then look for local maxima in the accumulator, which will give the (ρ, θ) parameters pairs for the lines found in the image. The higher the accumulator value, the stronger the line.

2.7.3 Homography

Perspective transformation, also known as homography, can be used to wrap a 3D real-world object on the 2D frontal view [Hartley and Zisserman, 2004], as in Figure 2.32.

In order to explain how those perspective transformations work, firstly, it is necessary to introduce a model of camera known as the *pinhole camera model*, in which each point M of a real-world 3D object goes through the pinhole of a camera and is projected in a point m on the image plane, which is the image as seen by the camera. In Figure 2.33, the image plane is placed between the object and the camera.



(a) (b)

Figure 2.32 — Perspective transformation.

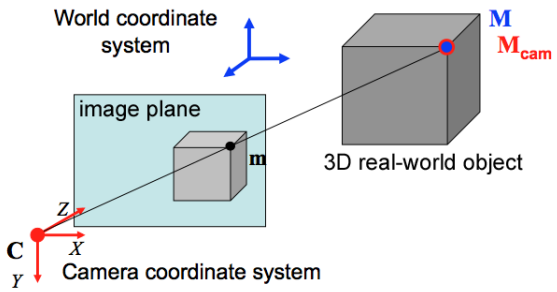


Figure 2.33 — Pinhole camera model.

In order to compute the relation between the 3D real-world point M_{cam}, which is M expressed in the camera coordinate system, and its corresponding projected 2D-point $m_{cam} = (m_x, m_y)$, also expressed in the camera coordinate system, one can use Thales' theorem and the *focal length* f of the camera, as shown in Figure 2.34 (a).

$$m_x = f\left(\frac{X_{cam}}{Z_{cam}}\right) \tag{2.23}$$

$$m_y = f\left(\frac{Y_{cam}}{Z_{cam}}\right) \tag{2.24}$$

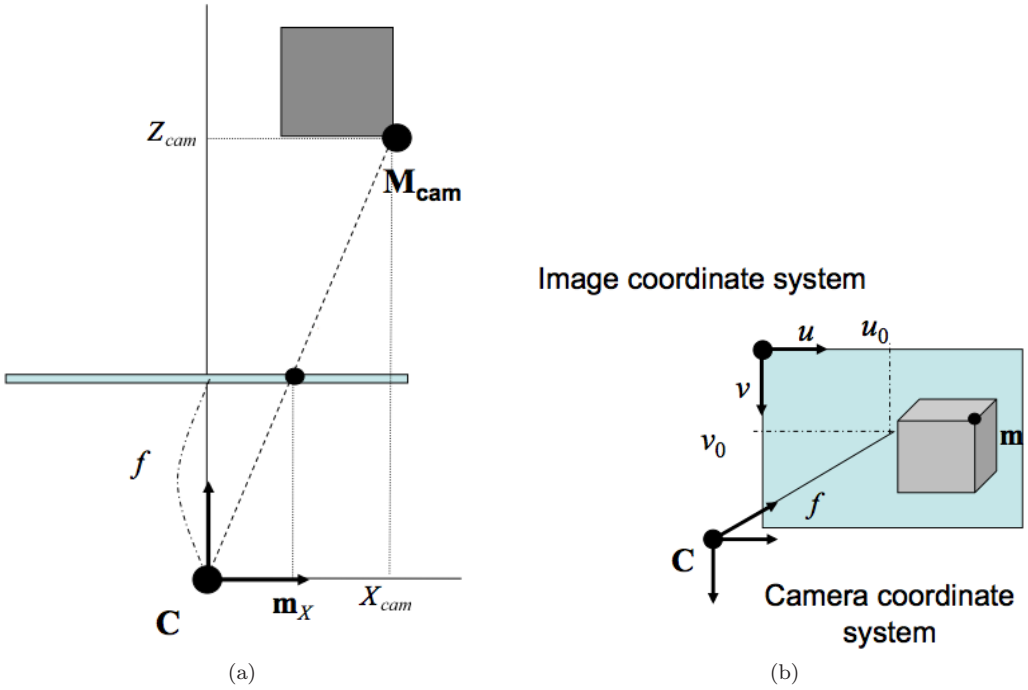


Figure 2.34 — Scale relations in the camera coordinate system is shown on the left (a). From the camera to the image coordinate system is shown on the right (b).

In order to express m in pixel-related units, that is $m_{pix} = (m_u, m_v)$, we first need two unit-converting factors k_u and k_v, where $k_u(k_v)$ is the inverse of the size of a pixel along the x-axis (y-axis), which are expressed in pixels per unit of focal length. Also, since a displacement away from the optical axis is possible, the coordinates (u_0, v_0) of the *principal point*, which is the projection of the camera center C on the image plane, also need to be taken into consideration, as shown in Figure 2.34 (b).

$$m_u = u_0 + k_u m_x \tag{2.25}$$

$$m_v = v_0 + k_v m_y \tag{2.26}$$

In order to rewrite all these relations in a more convenient way, that is in a matrix form, 2D-coordinates of the point m can be written in *homogeneous coordinates*, which add a dimension to the position vector, making the matrix form possible. Let (u, v, w) be the homogeneous coordinates of m. Hence:

$$m_u = \frac{u}{w} \tag{2.27}$$

$$m_v = \frac{v}{w} \tag{2.28}$$

Therefore, the global relation between a point M_{cam} of a 3D-real world object, expressed in the

camera coordinate system, and its projection m on the image plane, expressed in homogeneous pixel-related coordinates, is given by:

$$m = AM_{cam} \Longleftrightarrow \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{pmatrix} \tag{2.29}$$

where A is called the *matrix of intrinsic parameters* of the camera.

To express the real-world coordinate system using camera coordinates, which can be done in four operations, which is three rotations and one translation. Let R be the matrix of rotation coefficients and T be the translation vector, hence:

$$M_{cam} = RM + T \Longleftrightarrow \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \tag{2.30}$$

Using homogeneous coordinates again, M can be written as $(X, Y, Z, 1)$ instead of (X, Y, Z) and the relation becomes:

$$M_{cam} = (R|T)M \Longleftrightarrow \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.31}$$

where $(R|T)$ is the joint rotation-translation matrix, also called *matrix of extrinsic parameters*. Therefore, the global process of projecting a point M of a 3D-real world object,expressed in the real-world coordinate system, onto its projection m on the image plane, expressed in homogeneous pixel-related coordinates, can be described by the *projection matrix* P.

$$m = AM_{cam} = A(R|T)M = PM \tag{2.32}$$

Computation of the extrinsic parameters

To compute the coefficients of the 3×4 projection matrix P, it is necessary to calculate the matrix of extrinsic parameters $R|T$. A set of points M_i belonging to the object as well as a set of their corresponding points m_i in the image plane need to be provided. Several advanced optimization algorithms, such as the Levenberg-Marquardt algorithm, can then be used to minimize the reprojection error ϵ, that is the sum of differences between the actual image points and their computed projections.

$$\epsilon(R, T) = \sum_i ||A(R|T)M_i - m_i||^2 \tag{2.33}$$

Planar homography

In the particular case of 2D-real world objects, such as images, the perspective transformation can be simplified since an object point M, expressed in the object's coordinate system, can have its Z coordinate put to zero without any loss of generality, since this object only has two dimensions. Therefore, the point M becomes $(X, Y, 0)$ or $(X, Y, 0, 1)$ in homogeneous coordinates. This simplification leads to another simplification in the matrix of extrinsic parameters $(R|T)$, as shown hereunder.

$$
\begin{aligned}
m &= PM && (2.34)\\
&= A(R|T)M && (2.35)\\
&= \begin{pmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} && (2.36)\\
&= \begin{pmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} && (2.37)\\
&= HM' && (2.38)
\end{aligned}
$$

Where H is a 3×3 *planar homography* matrix and M' is the two-dimensional equivalent of M.

2.8 Chapter summary

In this Chapter we presented the necessary background to understand object recognition algorithms for this thesis. Several algorithms are inspired and derived from the structure of human brain, however full simulation of a human brain is not feasible now. Several computer vision algorithms are very similar to how the brain process images. We presented an approach for simulation in computer, brain computer vision processing. After the state-of-the-art algorithms are categorized. Most popular algorithms are referred and briefly discussed.

Become the edge on a little blade of grass and you'll be greater than the world's axis.

Légy egy fûszálon a pici él, s nagyobb leszel a világ tengelyénél.

Attila József (1905 — 1937)

Graph based Object Detection - GOD

3

This Chapter introduces a novel graph-based approach for 2D and 3D object duplicate detection in still images. A graph model is used to represent the 3D spatial information of the object based on the features extracted from training images so that an explicit and complex 3D object modeling is avoided. Therefore, improved performance can be achieved in comparison to existing methods in terms of both robustness and computational complexity. Effectiveness of the proposed object duplicate detection algorithm is measured over different object classes. Furthermore, different limitations of our approach are analyzed by evaluating performance with respect to the number of training images and calculation of optimal parameters in a number of applications. This method is shown to be robust in detecting the same objects even when images containing the objects are taken from very different viewpoints or distances.

3.1 Introduction

With the technological evolution of digital acquisition and storage technologies, millions of images and video sequences are captured every day and shared in online services such as Facebook, Flickr, and Picasa. As keyword-based indexing is very time-consuming and inefficient due to linguistic and semantic ambiguities, content-based image and video retrieval systems have been proposed (e.g. [Vajda *et al.*, 2009a, 2010a,c]). Within such systems, a query document is compared to all the documents in the database by making use of content-based features extracted from it. However, since the features are extracted from images which contain two-dimensional projections of three-dimensional scenes, the features may change significantly depending on the view point. Thus, the systems often fail to retrieve relevant content in response to the given queries.

In general, content-based image retrieval can utilize different representations for describing the image content, including global descriptors, feature points, or regions. Recently, interest has turned towards higher-level representations such as objects. Given a query image containing an object, an image retrieval system can perform two tasks: object recognition or object duplicate detection. Object recognition aims at finding all the instances of a certain object class (such as cars, or shoes), while object duplicate detection represents a more specific task of finding only a particular sample of that object class (such as "red Citroen C3 car" or "white Converse sneakers"). Figure 3.1 illustrates the relationship between object duplicate detection and object recognition problems. Therefore, within a complete system object recognition is usually applied first to detect a relevant class of objects (e.g. faces, cars) and then object duplicate detection is used to find a specific instance of that object class. Our object duplicate detection system is able to fulfill both tasks together.

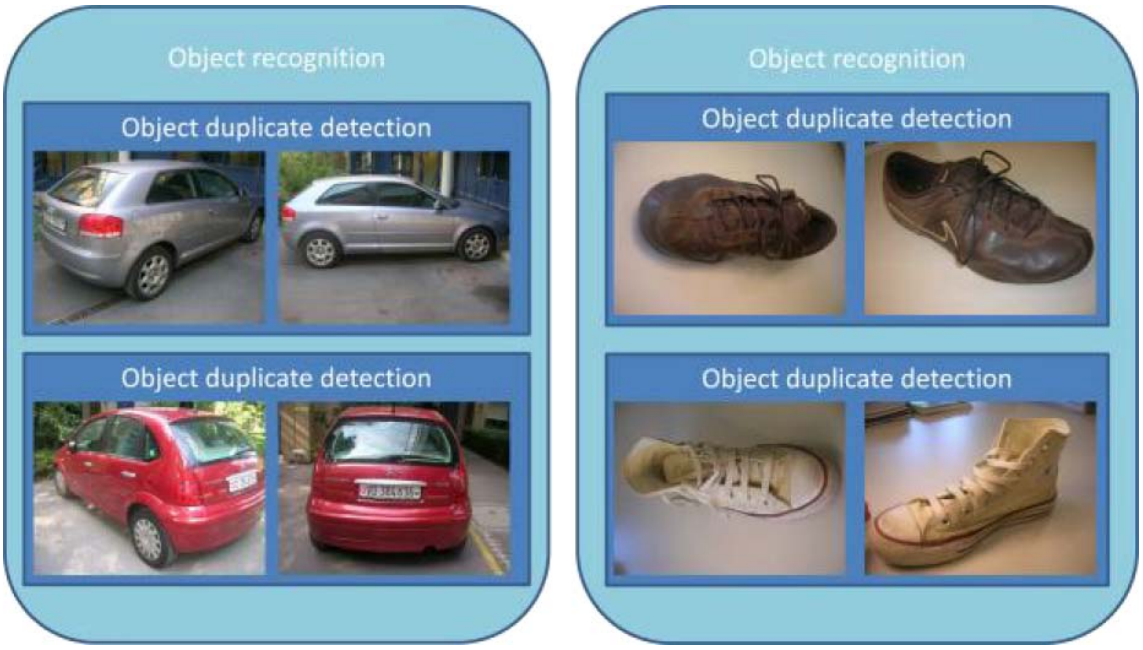


Figure 3.1 — Illustration of relationship between object recognition and object duplicate detection. While the former groups objects into different classes such as cars and shoes, the latter distinguishes between specific shoes or cars.

In this chapter, we are focusing on the object duplicate detection task. The general goal is to detect the presence of a target object in a set of images based on an object model created from a small set of training images. Duplicate objects may vary in their perspective, have different sizes, or be modified versions of the original object after minor manipulations, which do not change their identity. Therefore, object duplicate detection should be robust to changes in position, size, view, illumination, and partial occlusions.

We propose and analyze a novel graph-based approach for 3D object duplicate detection in still images [Vajda *et al.*, 2009b]. This approach combines the efficiency of a bag of words model

with the accuracy of a part-based model, which are described in Section 3.2, i.e. we make an attempt towards 3D modeling, while keeping the efficiency of 2D processing. A graph model is used to represent the 3D spatial information of the object based on the features extracted from the training images so that we can avoid explicitly making a complex 3D object model. Therefore, improved performance can be achieved in comparison to existing methods in terms of robustness and computational complexity. Another advantage of our method is that it requires only a small number of training images in order to build a robust model for the target object. Usually, several images from different views of an object are needed to create its 3D model. However in our approach, only a few common features are necessary to link spatial graphs from different views; therefore fewer training images are needed for the model creation. The method is evaluated through a comprehensive set of experiments, in which an in-depth analysis of its advantages and limitations is performed and optimal algorithm parameters are derived from the analysis. A comparison with three state of the art best-performing methods shows its significant performance improvement, because unlike our method, they consider a 3D object as 2D.

We present the evaluation results of our graph-based object duplicate detection algorithm and analyze them in order to answer the following important questions:

1. What is the impact of the number of training images on the accuracy of our algorithm?
2. How do the optimal algorithm parameter settings change through different numbers of training images?
3. How does the detection performance depend on different object classes?
4. How does its performance compare to the state of the art?

The remaining sections of this chapter are organized as follows. We present the main contribution of the chapter in subsection and introduce related work in the following subsections. Then, we describe our approach for object duplicate detection in detail in Section 3.3. Next, experiments and results are shown in Section 3.4 and 3.5. Finally, we conclude the chapter with a summary in Section 3.6.

3.2 Related work

Typically, any object duplicate detection method contains the following tasks: feature extraction, object representation, similarity measurement and searching tasks. In the following, we review the state of the art of these tasks.

3.2.1 Feature extraction

The first important step of object duplicate detection is to extract salient features from given images. Features for object duplicate detection can be grouped into global and local features as discussed in Section 2.4. Global features usually describe an object as a whole, while local features are extracted from particular parts of the object, such as salient regions. One of the most popular global features can be the Histogram of Oriented Gradient (HOG) [Felzenszwalb

et al., 2008], however it can be used as a local feature on interest regions. Gradient information is very accurate on images in different illumination conditions. HOG is a grid based histogram on this gradient information of the image. Global features usually require an exhaustive search over the whole image for the target object having various scales to localize it. Thus, it is more time consuming when compared to a search using local features, which are typically scale and rotation invariant. The extraction of these local features usually consists of two steps. First, the salient regions are detected in a way robust to geometric transformations. Second, a region description for each of the detected regions is generated to make them distinguishable. Reliable region detectors robust to illumination and viewpoint changes consider affine covariant regions [Mikolajczyk *et al.*, 2005], known to be scale, rotation and translation invariant. Among the most commonly used region descriptors is the Scale Invariant Feature Transform (SIFT) [Lowe, 2004] (The algorithm is patented in the US; the owner is the University of British Columbia: US 6711293), which is based on an approximation of the human visual perception. A faster version of SIFT descriptor with comparable accuracy, called Speeded Up Robust Features (SURF), is proposed in [Bay *et al.*, 2006b] (An application of the algorithm is patented in the US: US 2009238460).

3.2.2 Object representation

Regarding object representation, one can generally distinguish between spatial and non-spatial approaches. The latter does not consider any spatial information with respect to the object and its individual parts, such as the Bag of Words (BoW) model, which is based on a histogram of local features [Fei-Fei and Perona, 2005]. Zhang *et al.* [Zhang *et al.*, 2007] presented a comparative study on different local features on texture and object recognition tasks based on global histogram of features. BoW method gives a robust, simple, and efficient solution for recognition without considering the spatial information of the object. The advantage of our object representation over the BoW model is that spatial information from the object is considered using a graph representation. However, the graph modeling increases the computational complexity.

On the other hand, spatial models such as the Part-Based Model also consider the spatial information of objects for improved performance. Connections between different parts of an object can be achieved using different structures. A star structure has been used to represent objects based on HOG features [Felzenszwalb *et al.*, 2008]. Another common way for considering spatial relationships between individual parts is to employ graph models often referred to as structural pattern recognition [Neuhaus and Bunke, 2006]. These graph matching approaches have been successfully applied to handwriting, character, and contour-line recognition. A generative model based on graph matching is the Random Attributed Relational Graph (RARG), which is able to capture the structural and appearance characteristics of parts extracted from objects [Zhang, 2006]. However these methods are more suitable for object recognition and they need several training images for training the object model.

Most of the object representations consider the objects in the 2D image space only, improved performance can be achieved by using 3D models because real-world objects are inherently 3D. However, the creation of complete 3D models requires a large number of images taken from all possible viewpoints, which may not be available in real applications. Solutions to this problem have

been proposed to consider viewpoint discrepancy between training and query images. An approach described in [Sivic *et al.*, 2006] uses tracking of a target object in an image sequence to retrieve different views of the same object and to group video shots based on object appearance. Several appearances are then used to recognize objects more reliably. In [Rothganger *et al.*, 2004] a full 3D model of the object, created from the video sequence, is used for the detection of objects. Our approach makes an attempt towards 3D modeling, while keeping the efficiency of 2D processing, using a graph model to represent the 3D spatial information so that we can avoid explicitly making a complex 3D object model.

3.2.3 Object duplicate detection

In this section we review representative object duplicate detection methods based on the previously described tasks.

The method presented in [Lowe, 2004] is based on local feature extraction, the general Hough transform for object localization, and pose estimation using the "RANdom SAmple Consensus" (RANSAC) algorithm. Our method is based on this algorithm, but 3D object detection performance is improved by using spatial graph matching and considering more than one training image.

In [Sivic and Zisserman, 2006], descriptors are extracted from local affine-invariant regions and quantized into visual words, reducing the noise sensitivity of the matching. Inverted files technology is used to match the video frames to a query object and retrieve those which are likely to contain the same object. However, this work considers only 2D objects, such as posters, signs, ties, and does not take into account real 3D objects. In this chapter a method is proposed and analyzed for real 3D objects.

An extension of this approach uses key-point tracking to retrieve different views of the same object and to group video shots based on the objects appearance [Sivic *et al.*, 2006]. The tracked object is then used as an implicit representation of the 3D structure of the objects to improve the reliability of the object duplicate detection. This method has proven to be more effective than a query with a single image, but it requires that all the relevant aspects of the desired object are present in the query shot, which limits its applicability.

A useful application of object duplicate detection to image search in a large database is presented in [Gool *et al.*, 2009], where images taken by web cameras are automatically annotated with bounding boxes containing objects.

Detecting buildings in a large image database is presented in [Philbin *et al.*, 2007]. The BoW method is applied for preselecting images from the large database and an efficient spatial verification is considered for further analysis. The database contains up to one million images. To resolve the problem of a high computational complexity due to the size of the database, they use a forest of 8 randomized k-d trees as a data structure for storing and searching features.

3.3 Graph-based object duplicate detection

In this section, our 3D object duplicate detection algorithm will be described in more detail. The goal of this algorithm is to detect the presence of a target object and to predict its location in a

set of images based on an object model created from training images. SIFT features are used to improve the detection speed in large databases thanks to existence of efficient logarithmic nearest neighbor search methods for local features in tree based data structure. We target 3D object duplicate detection by using a graph model, which imposes spatial constraints between features and between the different viewpoints of the whole object to improve the accuracy of the object duplicate detection.

Since objects are considered as 3D objects, in principle more than one training image is needed to create a reliable 3D model for an object. However, as the spatial graph model is only an approximation of a 3D model with multiple 2D models (views) linked with each other, fewer training images (including only one image) are sufficient for model creation. In fact, we will show in the Section 3.4 that only very few images are needed to create a reliable model for 3D object detection.

The system architecture is shown in Figure 3.2. In the training phase, an object model is created from a set of images containing this object. In the testing phase, this object model is used to detect objects and to predict their locations and sizes in test images. Each of the two phases starts with the same feature extraction step, which is shown separately. In the following sections, the feature extraction, the training, and the testing phases are explained in detail.

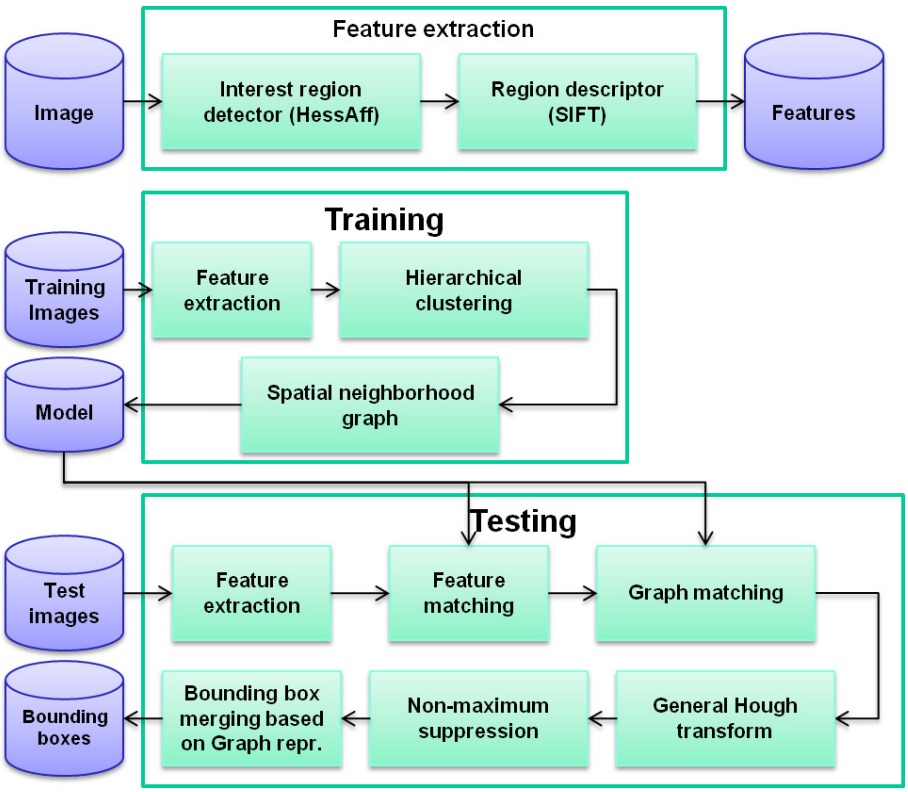


Figure 3.2 — Overview of the object duplicate detection approach with the individual training and testing stages, and the commonly used feature extraction step.

3.3.1 Feature extraction

As explained in the Subsection 2.4, local feature extraction consists of salient region detection and region descriptor generation. The pseudo codes of these steps in our algorithm are shown below:

Algorithm 1 F = Feature extraction(i:image)

$IR :=$ **Hessian affine detector** (i);
where IR_i:(p:position, o:orientation, s:scale), interest region on image i.
$F :=$ **SIFT**(IR);
where F_i: (IR_i, f : feature descriptor), feature.

1. Sparse local features are used to resolve the object localization problem more efficiently. Interesting regions are extracted using the Hessian affine detector [Mikolajczyk and Schmid, 2002], as it has been shown to outperform other detectors dealing with scale and view point changes [Mikolajczyk *et al.*, 2005]. Based on a combination of corner points detected by an Hessian matrix, multi-scale analysis through the Gaussian scale-space, and affine normalization using an iterative affine shape adaptation algorithm, the Hessian affine detector finds regions which are stable across different scales, rotations, translations, as well as changes in illumination and viewpoint. The position, scale and orientation are computed for each of the regions and will be used within the graph model. Implementation and default parameters were provided in [Zynga, 2009a].

2. To describe the detected regions, SIFT features are extracted, as they remain robust to arbitrary changes in viewpoints. The idea of this algorithm is to approximate the human visual perception mechanism through features that share similar properties with the neurons in the inferior temporal cortex used for object recognition in primate vision systems [Serre *et al.*, 2005b]. High contrast candidate points and edge response points are processed for each region and dominant orientations are considered as in the inferior temporal cortex. These steps ensure that region descriptors are more stable for matching. Implementation and default parameters were provided in [Zynga, 2009a].

The complexity of the feature extraction step is $O(n)$, where n is the number of pixels in the query image, as the feature extraction creates $O(n)$ features by making use of pyramids for detection of scale invariant features.

3.3.2 Training

During the training phase, a set of training images containing the target object from different views is processed, and a model for the object is constructed. The training images correspond to a single object filling up the whole field of view. Therefore, we assume that the object is positioned around the center of training images, whose boundaries are used as the bounding box of that object. In the following, each step of the training phase is described. The pseudo code of the training phase is the following:

1. *Features are extracted* from the training images, as described in the previous subsection.

Algorithm 2 $Model$ = Training(I:images)

F := **Feature extraction** (I);
$Storage$:= **Hierarchical clustering**(F);
$Model$:= **Spatial neighborhood graph creation**(F);
where $Model$:(G:Graph, A:Graph attributes);
G:(V:feature nodes, E: edges), edges are between the close features in spatial domain.
$A(E)$: (d:distance, o:orientation), attributes of the edges.
$A(V)$:(p:position, o:orientation, s:scale), relative position, orientation and scale to the object center and object scale.

2. *Hierarchical clustering* is applied to the features, to group them based on their similarity. This improves the efficiency of the feature matching by adopting a fast approximation for the nearest neighbor search. More specifically, hierarchical k-means clustering [MacQueen, 1967] is used with branch factor 10, to derive the vocabulary tree, similarly as described in [Nister and Stewenius, 2006]. It is a balanced tree whose inner nodes correspond to the feature of the cluster centers derived from all its children. The balanced tree is a tree where there are no big differences between the depths of the leaves. The computational complexity of finding a nearest neighbor for a given feature in this tree structure is significantly less than an exhaustive nearest neighbor search. However, it is important to mention that this approximation may occasionally cause erroneous matches, which are discarded by the further validation process.

3. Finally, the *spatial graph model is constructed* from the features and their positions. The nodes of the graph represent the features of the training images. Each node also stores the scale, and orientation of the corresponding region of the feature and the relative position from the object center, normalized by the size of the object. The edges of the graph are the spatial nearest neighbors of two features. The attributes of edges are the distance and orientation of the neighboring nodes. These attributes are important for the matching step in the testing phase.

The computational complexity of the training phase can be even superpolynomial, because in worst case, k-means clustering algorithm can converge very slowly ($2^{\Sigma(N)}$ computation steps, where N is the number of features [Arthur and Vassilvitskii, 2006]). However it is usually fast without guarantee that it will converge to the global optimum.

3.3.3 Testing

In the testing phase, the graph model derived from a small set of training images is used to detect similar objects in other test images and to predict their bounding boxes. The whole testing phase is illustrated in Figure 3.3 and its individual steps will be described in more detail below. The pseudo code of the training phase is the following:

1. *Features are extracted* from the query image, as described before, which results in several robust region descriptors.

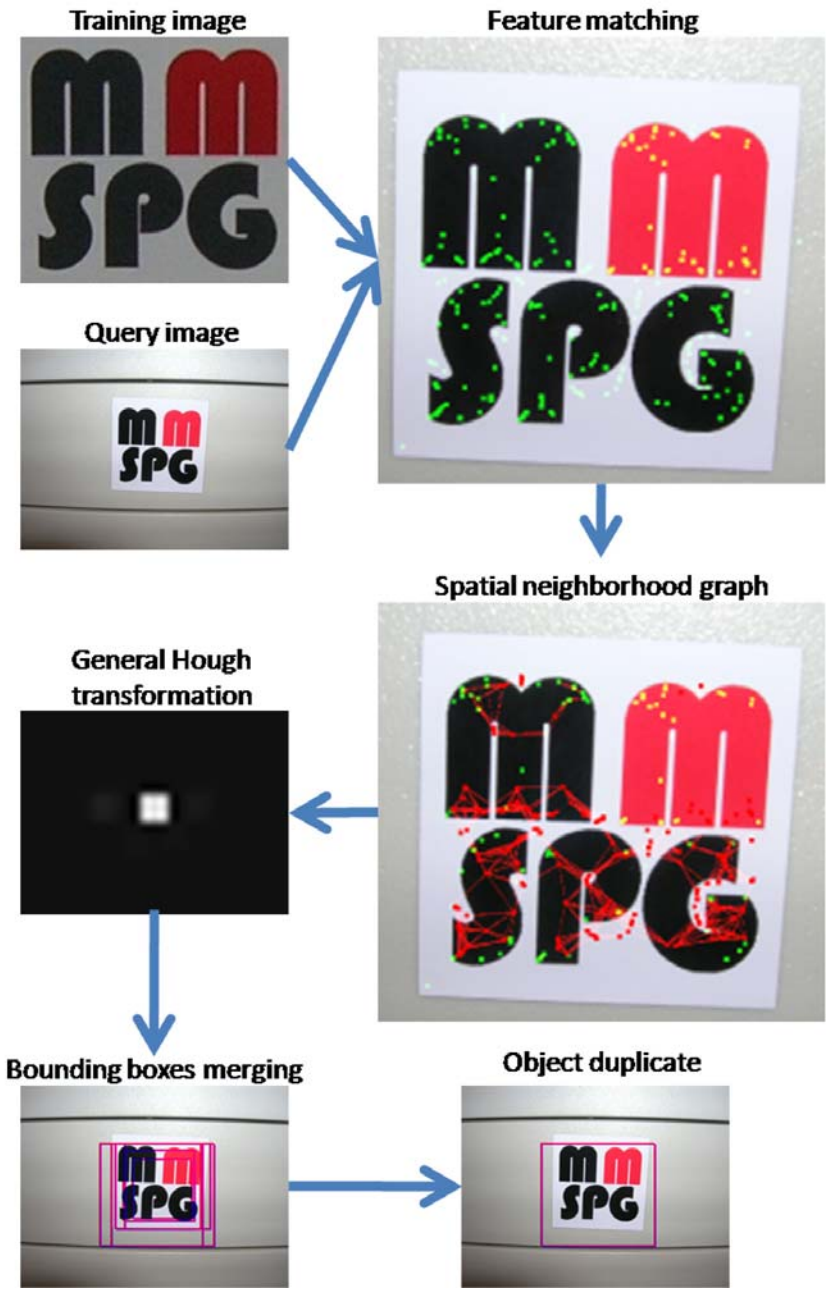


Figure 3.3 — Illustration of the different steps of the testing stage with individual feature matching, spatial graph matching and bounding box estimation.

Algorithm 3 $Boundingboxes$ = Testing(i:image, $Model$)

F := **Feature extraction**(i);
$Matches_{feature}$:= **Feature matching**(F, $Model.G.V$) ∩ **Feature matching**($Model.G.V$, F);
  $Matches$:(f_{query}:feature, f_{model}:feature) one to one matching using Storage.
$Query$:= **Spatial neighborhood graph creation**(F);
  $Query$:(G:Graph, A:Graph attributes)
  G:(V:feature nodes, E: edges), edges are between the close features in spatial domain.
  $A(E)$: (d:distance, o:orientation), attributes of the edges.
  $A(V)$:(p:position, o:orientation, s:scale)
$Matches_{graph}$:= **Graph matching**($Query$, $Model$, $Matches_{feature}$)
  $Matches_{graph}$ = ($V_{query} \times V_{model}$:feature matches, $E_{query} \times E_{model}$:edge matches)
$Boundingboxes$:= **General Hough Transformation**($Query$, $Model$, $Matches_{graph}$)
  $Boundingboxes$ = (c:coordinates, F_{query}:Features)
$Boundingboxes$:= **Non-maximum Suppression**($Boundingboxes$)
$Boundingboxes$:= **Bounding box merging**($Boundingboxes$, $Matches_{graph}$)

2. These features are matched to those in the graph model derived from the training images using a one-to-one nearest neighbor *matching* which is illustrated in Figure 3.4. First, for every feature from the test images the nearest neighbors in the graph model are determined based on the Euclidean distance. Then, for each feature from the graph model the best matching feature in the test image is searched back. If it leads to the original feature in the test image, then this match is a one-to-one match; otherwise the matching features do not correspond to each other and they are discarded. Furthermore, matches with a distance larger than a predefined threshold T_d are also discarded. This procedure ensures the selection of very reliable matching features. Due to the tree representation of features, the complexity of this matching step is $O(N \cdot log(N))$, where N is the number of features.

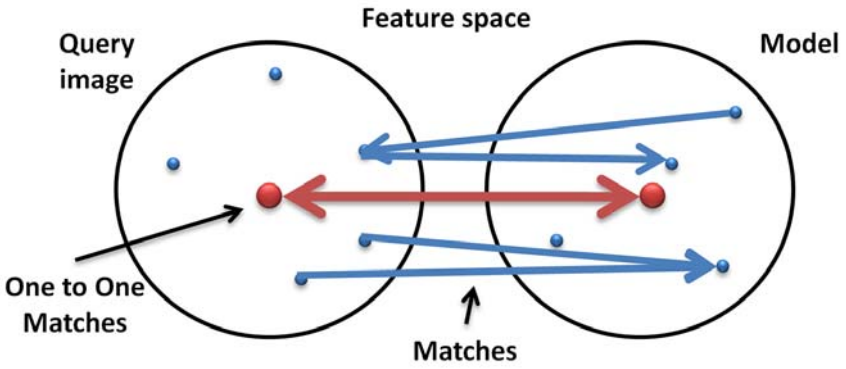


Figure 3.4 — One to one feature matching.

3. A *spatial graph* is constructed from the matched features and their positions from the query image by applying the same method as the graph creation in the training phase. Graph matching is applied between the graph model derived from training images and the graph created from the query image. Figure 3.5 shows the graph matching step. The graph contains spatial information of the potentially matching objects, hence making the algorithm more

robust. The previous feature matching step creates the connections between these two graphs, as shown with red and blue lines in Figure 3.5. First, on the graphs we match the edges, as shown with a red line in Figure 3.5. An edge on the query graph is matched with an edge on the model graph, if the two edges have nodes having similar scales. More precisely, the edges are matched if the ratio between normalized scales is between T_R and $1/T_R$, where the threshold T_R is set manually. The scale values are obtained in the feature extraction step as described in Subsection 3.3.1. The normalized scale is the scale of the feature divided by the scale of the matched feature. Second, the number of edges that go out from each node is examined and only the nodes having at least T_{outdeg} edges are accepted. This produces a rather robust graph of the objects. An expected property of this graph is that, ideally, nodes of a given edge should not be part of two different objects (i.e. each edge is a part of only one object). However, it is possible to have multiple disconnected graphs in a model of an object. With this method, several mismatched features can be discarded, thanks to the spatial position in the object and the normalized scale of features.

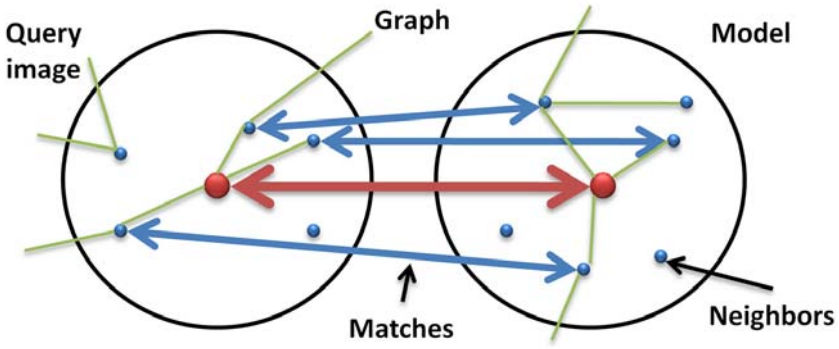


Figure 3.5 — Spatial neighborhood graph matching between the model graph and the query graph. Red line shows finally matched features.

4. To estimate the bounding boxes of the detected objects, the *General Hough transform* is applied on the nodes of the matched graph [Ballard, 1981]. Each node in this graph votes for the center and the size of the bounding box in the query image, using the orientation and scale of the extracted features. Hierarchical Hough space is created for each size of the object by of 1.2. The Hough bin sizes of each scale set to third of the object size. Each vote is weighted by the number of edges connected to the node. A histogram of the object's center and its scale is obtained from the voting result. Then, the local maxima whose values are greater than a threshold T are searched for in the obtained histogram.

5. As the above procedure may return several bounding boxes with spatial overlaps, *non-maximum suppression* is applied to discard duplicate bounding boxes [Neubeck and Gool, 2006]. The overlap threshold is set to 0.4. This method leads to good results when using one training image; however if different views of the same object in training images are used, then multiple bounding boxes can be obtained for that object.

6. Bounding boxes obtained due to the different views of an object in training images are *merged*

based on graph intersection as shown in Figure 3.6. This is done by considering the number of edges of the graphs intersecting two bounding boxes. The ratio between the number of intersected edges and the number of edges in both bounding boxes are threshold by a parameter T_{bb}. Thus, if a sufficient number of edges intersect, the two bounding boxes are merged [Warshall, 1962]. Finally, each bounding box represents a target object in the test image. This graph based method solves the problem of multi-view images in which separate bounding boxes may be obtained for the same object. Even if more than one target object is present in the test image, our algorithm still successfully detects them, as each object produces a separate graph.

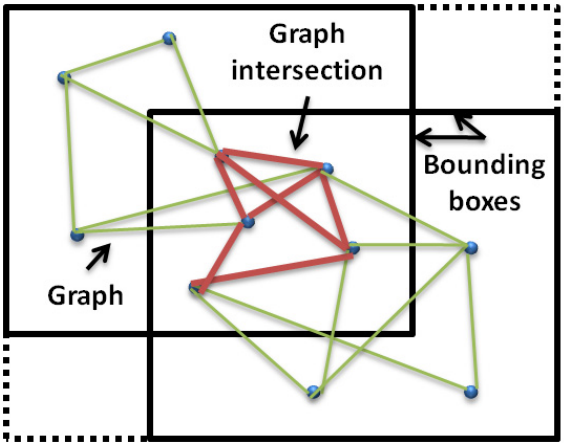


Figure 3.6 — Bounding box merging based on graph intersection.

The complexity of the feature extraction step is $O(n)$ as discussed in 3.3.1, where n is the number of pixels in the query image, as the feature extraction creates $O(n)$ features. General Hough Transformation is also linear to the number of features, because scale and orientation information are used for voting in Hough space. The complexity of feature matching, spatial neighbor graph construction and matching are $O(n \cdot log(n))$, because for each feature a tree path has to be computed. Therefore the overall cost of our algorithm is $O(n \cdot log(n))$.

3.4 Experimental setup

The created dataset is described in the following Subsection and the parameter setup is introduced. As evaluation methodology we are following the general detection evaluation scheme as presented in Appendix A.

3.4.1 Database

We created a new database for 3D objects in different points of view and distances, due to lack of realistic databases in this area.

Table 3.1 — Summary of the database of images taken under challenging conditions.

class	#images	#samples	condition	planar
bag	103	3	monotone	3D
bike	50	4	various features	3D
body	43	3	textured cloths	3D
book	76	13	shiny	2D
building	57	3	artificial	3D
can	60	3	small	3D
car	37	3	shiny	3D
workbook	49	9	text	2D
face	39	3	deformable	3D
logo	130	4	small	2D
motor	31	3	shiny	3D
newspaper	67	17	text	2D
poster	39	7	text,images	2D
shoes	40	3	small	3D
stone	32	3	monotone	3D
total	850	≥ 3	various	2D,3D

We collected 850 images including ten 3D and five 2D object classes: bag, bicycle, body, face, shoes, stone, can, car, building, motor, poster, logo, newspaper, book and workbook (Figure 3.7, Table 3.1).

Each of the 15 classes contains at least three samples. Figure 3.8 provides three images for two selected classes: building and shoes. As it can be seen from these samples, images with a large variety of view points and sizes are included in each class.

The information of the relative size and viewing angle of the object in each image was obtained and attached to the image in order to perform analysis on scale- and orientation-invariant detection by the proposed algorithm. For this, the four corner points of an object were manually selected.

The database contains different instances of the same object class (e.g. red shoes and white shoes) in order to make the object duplicate detection task more challenging. Since the task is detect duplicate objects, retrieving a different instance (red shoes) from the one given in the query image (white shoes) is regarded as a wrong result produced by the system.

3.4.2 Parameter setup

In this section the evaluation environments are assessed.

The parameter settings of our algorithm are set based on several experiments and heuristics. The number of clusters in the vocabulary tree was set to 128, as this provided a good tradeoff between complexity and accuracy. For the feature matching, the distance threshold was selected to be $T_d = 10^5$. In the spatial neighborhood graph $T_{outdeg} = 4$, to accept robust matching features. The normalized scale ratio was set to $T_R = 2$. Finally, the threshold $T_{bb} = 0.1$ was chosen to merge bounding boxes. The optimal detection threshold T for the algorithm was derived based on various experiments as summarized below.

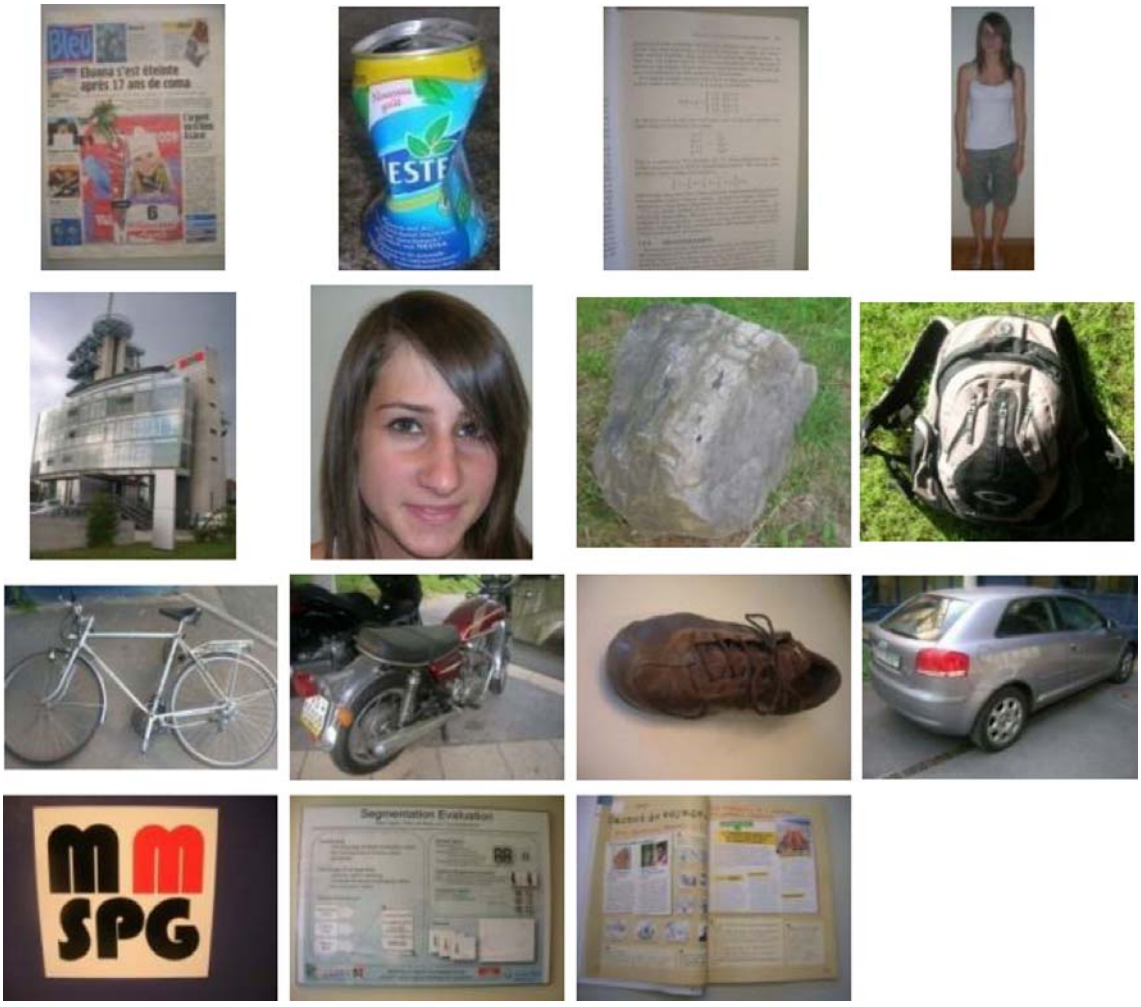


Figure 3.7 — Samples of the different 2D and 3D object classes within the dataset used in evaluations.

Figure 3.8 — Samples for two objects under diverse viewing conditions within the dataset used in evaluations.

3.5 Results and analysis

Multiple training images

In this experiment, we trained our system with one, two or three images, and tested it with the remaining images in order to analyze the performance of the object duplicate detection as a function of the number of training images. Negative images are all images which do not contain the ground truth object. In our experiments we define as negative images all different images from the same class of the object and several not related images.

Figure 3.9 (a) plots the corresponding ROC curves. The results show that the performance of the algorithm varies according to the number of training images. One can notice that using more than one training image can significantly improve the performance because it makes the object model robust against different points of views. As an example, the results show that for $FPR = 0.10$ a $TPR = 0.85, 0.92, 0.97$ is achieved when the system is trained with one, two and three images, respectively.

The object duplicate detection algorithm was also evaluated using PR curves as shown in Figure 3.9 (b). These curves complement the previously discussed results and provide a better visualization of the opposing effects (high precision vs. high recall) which are inherent to any detection task. For instance, the results show that for $R = 0.90$ a $P = 0.28, 0.67, 0.78$ is achieved when the system is trained with one, two and three images, respectively. For a high recall value greater than 0.8, higher precision values are obtained when more than one training image are used.

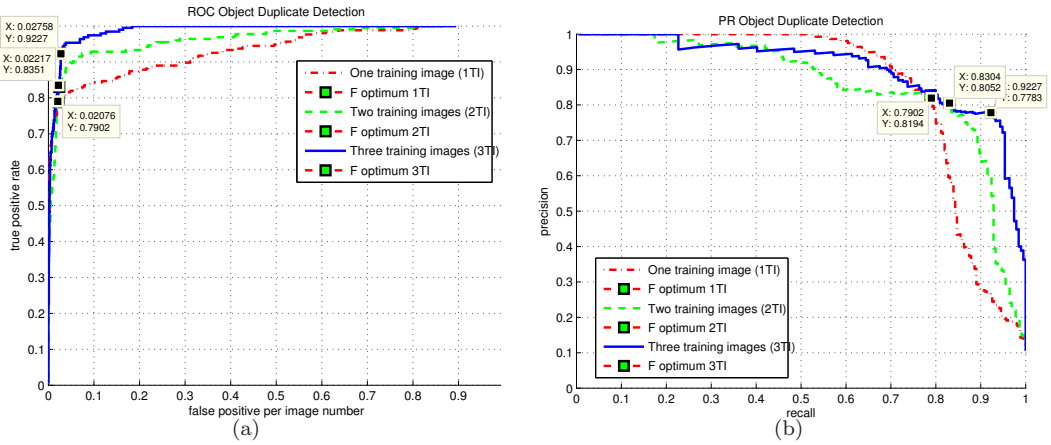


Figure 3.9 — Receiver operating characteristic (ROC) curves for different numbers of training images (1, 2 and 3) per object is shown on the left. A larger number of training images leads to an increased TPR for a fixed FP value. Recall versus precision curves for different numbers of training images (1, 2 and 3) per object is shown on the right.

Optimal parameter settings

In this experiment, we estimated the optimal parameter settings of our system. The F-measure is calculated in order to determine the optimal threshold value, T.

The results are shown in Figure 3.10 (a). For each of the cases where one, two and three training images are used, the maximum F-measure is found and shown with a marker in this figure. According to the results in this figure, F-measures of 0.80, 0.82 and 0.84 can be reached with thresholds equal to 51, 140 and 288, if our system is trained with one, two or three images respectively. Therefore, the optimal threshold is highly correlated with the number of training images. The optimal threshold increases significantly if more training images are used.

Figure 3.10(b) sshows the optimal threshold values producing the largest general F-measure values for each β value. Using the general F-measure, the importance of the precision and recall can be balanced according to the requirement of a given application using the object duplicate detection. If the β parameter is equal to one, the precision is as much important as the recall. Two other commonly used F-measures are the $F_{0.5}$ measure, which weights the precision twice as much as the recall, and the F_2 measure, which weights the recall twice as much as the precision. The optimal F-measure parameter settings, for which the precision and the recall are equally weighted, are also shown as markers in Figure 3.10.

3.5.1 Performance different classes

The goal of this experiment was to measure the performance of the object duplicate detection algorithm as a function of different object classes. The F-measure is computed for each of the 2D and 3D object classes and different classes are compared with each other.

The results are shown in Figure 3.11. According to these results, there are big differences

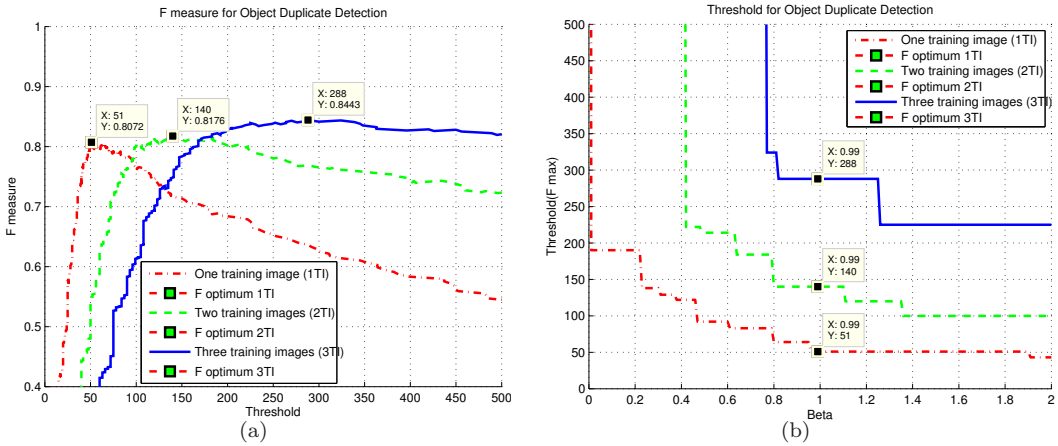


Figure 3.10 — F-measure versus detection threshold for different number of training images (1, 2 and 3) is shown on the left. Both the F-measure and the detection thresholds increase with a larger number of training images. Detection threshold versus β parameter of the general F-measure for different numbers of training images is shown on the right.

in detection performance between different classes, which are caused by various factors such as reflection properties, amount and presence of textures, or number of salient features. The object duplicate detection algorithm performs well with newspapers, thanks to the large number of pictures and textures in such objects. Duplicate detection of human bodies considers only the texture of the clothes and not their shape, which gives a surprisingly good result. Face identification is also a possible application, although its performance should be compared to those of state of the art face detection algorithms. Shiny objects, such as motor bicycles and cars, are hard to detect due to the changing reflections depending on lighting condition. Books are also among the classes showing the worst performance due to large illumination variations during the image acquisition of our dataset, which was not the case for newspapers.

Finally, an interesting example is shown in Figure 3.12. The object duplicate detection algorithm is performed on the class of cars. Interestingly, even the opposite side of a car can be correctly recognized when only one training image is used. This is due to the fact that the license plate of a car is the most salient region on both the front and the back sides of the car. Indeed, the quality of the duplicate object detection is partly due to the fact that the front and back plates of the car are identically shaped and formulated, which is not necessarily the case depending on the given Swiss Canton, or foreign policies. Nevertheless, the location of the car is shifted upwards due to the different position of the license plate with respect to the overall object.

3.5.2 Comparison with state of the art

The goal of this experiment was to assess the quality of proposed method for object duplicate detection in relation to the state of the art. We have compared it to the BoW method [Fei-Fei and Perona, 2005], RANSAC based geometry validation [Valle *et al.*, 2009] and Lowe's object duplicate detection [Lowe, 2004]. Parameter settings are selected as dedicated in the original papers and the

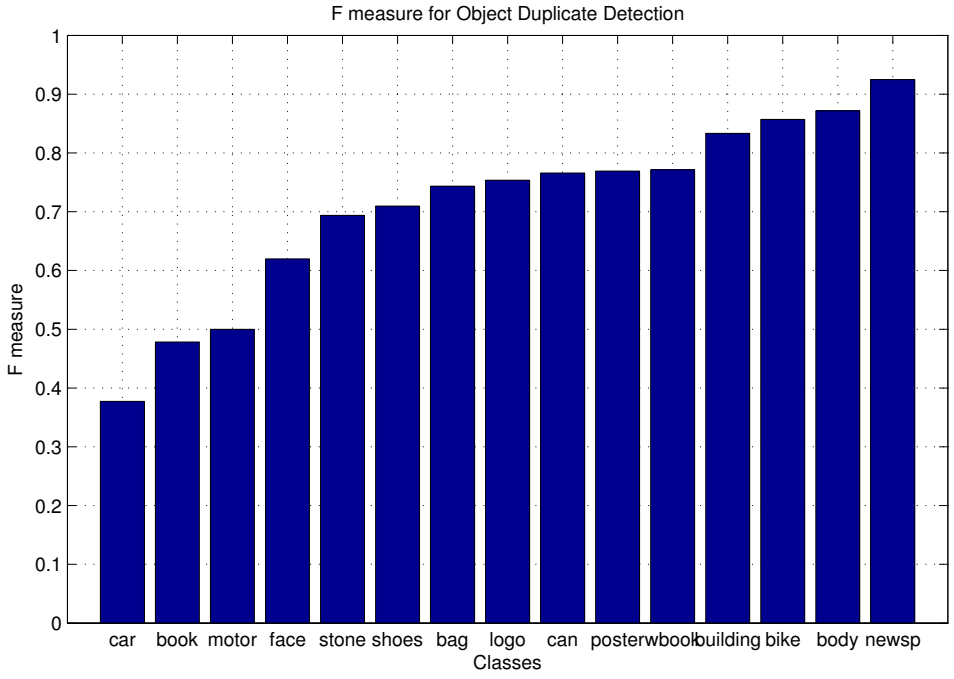


Figure 3.11 — Performance of the object duplicates detection in F-measure for each object class. The difference between classes is caused by various factors such as reflection properties, amount and presence of textures, or number of salient features.

Figure 3.12 — An example where our object duplicate detection algorithm detects the back side of a car thanks to its license plate. The training image is shown in the bottom left corner.

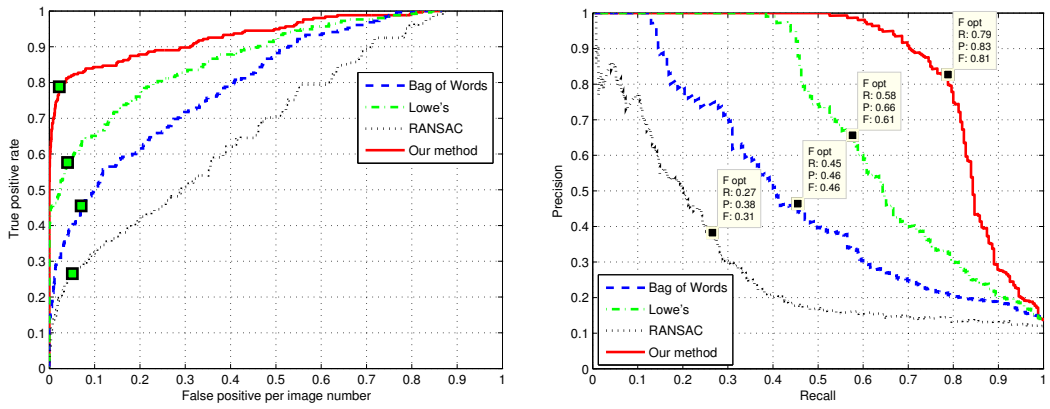
implementation is based on OpenCV open source library.

A drawback of the BoW method is that it creates a general histogram from local features extracted from the object region and does not consider their spatial information. However it is still very accurate and fast. It is not suitable for object localization, but it is still sensitive for large local object matching in constant searching time ($O(1)$). While various distance measures can be used to compare histograms obtained from the training and test images, it was shown that the histogram intersection distance performed the best [Swain and Ballard, 1991]. Thus, we implemented the BoW method using the histogram intersection distance for the performance comparison.

Lowe's object duplicate detection [Lowe, 2004] algorithm considers the spatial information by using General Hough Transformation similar to our approach. However, our method improves the quality of the matched features, due to the graph matching. The computational complexity of this algorithm is $O(n \cdot log(n))$.

RANSAC is a very popular geometry checking method. It improves the accuracy significantly, by validating affine transformation between the query and training images [Valle *et al.*, 2009]. Therefore we applied the RANSAC geometry checking algorithm on the SIFT based feature matching algorithm [Lowe, 2004]. RANSAC is a heuristic iterative method and it's complexity depend on the maximum number of iteration, which can make this algorithm slower than the others, mentioned above.

In Figure 3.13, ROC curves and PR curves are shown for the four methods: Bag of Words, Lowe's, RANSAC and our graph based method. The results show significant differences between the algorithms. The BoW method performs bad, since it does not consider any spatial information. Lowe's method shows improved performance by considering this information. Our method provides another considerable performance gain due to the graph matching. RANSAC performs the worst, which is surprising taking into account its popularity on general object duplicate detection.



(a) (b)

Figure 3.13 — Comparison of our method with BoW method, RANSAC based and Lowe's object duplicate detection algorithms (Lowe, 2004). On the left side, ROC curve and on the right side PR curve are shown.

For a more detailed analysis we compared the performance of the methods across the different classes shown in Figure 3.14. The advantage of our algorithm is observable for most of the classes. However, there are some cases where our method does not perform as good as the other methods. The BoW method works well on "cars","bags" and "lgoos" compared to the other methods, since it is more robust to the spatial information changes caused by varying view point and distance. In these objects there are only a few features and the available spatial information is not very reliable. However, for objects where the spatial information is crucial such as "books", "shoes", "posters", "buildings", "bodies" and "newspapers", the BoW method shows very bad results. The RANSAC based method performs the worst in many cases, because real 3D objects sometimes cannot be detected only by considered affine transformation, whereas it is significantly better than the BoW algorithm for flat objects, such as "newspapers", "workbooks" and "posters". Our method significantly outperforms these methods due to the graph matching in both 2D and 3D objects.

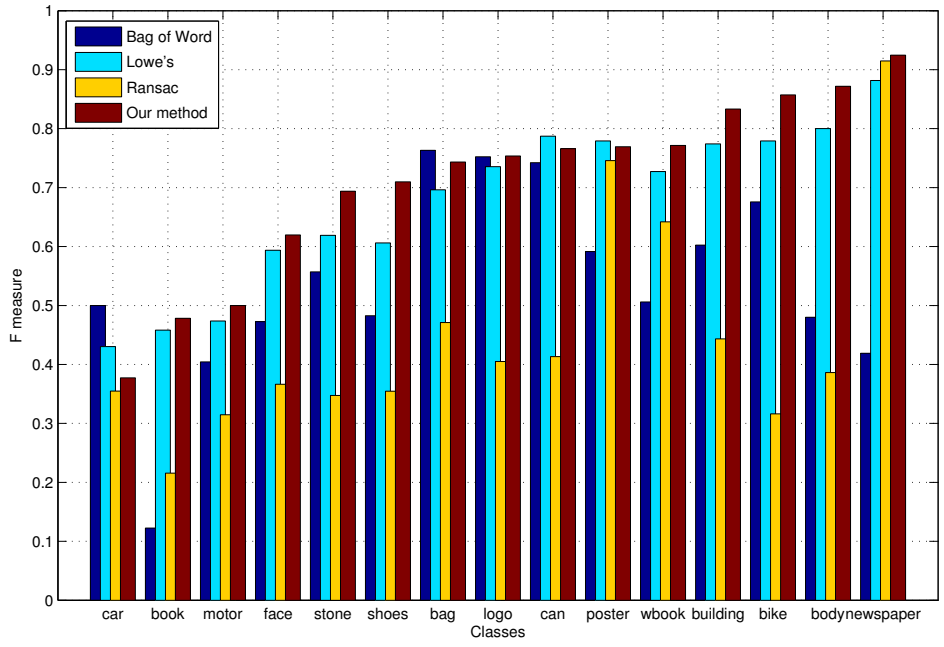


Figure 3.14 — Comparison is shown between our method, BoW, RANSAC and Lowe's object duplicate detection algorithm through different classes of objects.

3.6 Chapter summary

Image and video retrieval systems are becoming increasingly important in many applications.

A large number of applications can benefit from object duplicate detection. For example, in the popular photo sharing web pages, untagged images can be automatically annotated based on

the detection of the same objects from a smaller set of images with associated tags. Also, object duplicate detection may be used to search a specific object in a large collection, such as a suspect car in a video surveillance database. Moreover, when a user takes a picture of an object with his/her mobile phone, additional information about the object can be retrieved from the web, such as the price of a product, or the name and location of a monument.

In this chapter, we have proposed and analyzed a robust graph-based object duplicate detection algorithm for 2D and 3D objects. The experiments were performed on various classes of objects. The main conclusions that can be drawn from our experiments are:

- The performance of our object duplicate detection algorithm shows that more than one training image can significantly improve the detection accuracy.
- Considering F-measure and F_β-measure values, the optimal threshold was determined and found to vary with the requirement of specific applications.
- The comparison of performance between various classes of objects shows that our algorithm performs well with textured objects, while shiny objects are most difficult to detect (car, book, motor).
- We showed that our method works significantly better than Bag of Words method, RANSAC based and Lowe's object duplicate detection method on our dataset, due to the way the spatial information is considered through graph matching.

That's not me shouting, it's the earth that roars.

Nem én kiáltok, a föld dübörög.

Attila József (1905 — 1937)

Robust 3D object duplicate detection

4

In this chapter, we extend the previously proposed graph-based approach for 3D object duplicate detection in order to enhance its robustness against variations in the viewpoint and size. We determine how many training images, taken from which points of view, are necessary in order to achieve certain efficiency. Moreover, the performance of the algorithm is improved by generated training images, where the original training images are scaled and rotated in 3D space. Our experiments show that four training images are enough for 3D object duplicate detection from a planar view point and ten training images for omnidirectional detection. An algorithm on 3D captured images is shown to reduce feature size for mobile devices. Furthermore, we apply our object duplicate detection method to video sequences, where the training images are added in an intelligent way during the detection process. The goal of the proposed method is to retrieve key frames and shots of a video that contain a particular object. The key idea is to apply the proposed object duplicate detection method iteratively to the video sequence in order to compensate for 3D view variations, illumination changes and partial occlusions.

4.1 Introduction

A large number of applications can benefit from a precise object duplicate detection. For example, when a user takes a picture of an object with his/her mobile phone, additional information about the object can be retrieved from the web, such as the price of a product, or the name and location of a monument, once the object in the picture is accurately detected and recognized. It can apply in searching for counterfeit objects, or for abusive usage of logos, trademarks etc. Moreover, object

duplicate detection may be used to search a specific object in a large collection of image or video data, such as a suspect car in a video surveillance database. In this case, objects should be detected from any view point and at any size with a certain efficiency, because the objects in training and test images may appear in different viewpoints and/or sizes. Therefore, it is important to understand the limits of multi-view object duplicate detection which is the focus of this chapter.

We extend and analyze our proposed graph-based approach, described in Section 3.3, for 3D object duplicate detection in still images and video, considering detections from any view point and with any scaling factor. A graph model is used to represent the 3D spatial information of the object based on features extracted from the training images so that a complex 3D object processing is avoided. Therefore, improved performance can be achieved in comparison to existing methods in terms of robustness and computational complexity 3.5.2. The main goal is to analyze and determine how many images of the object of interest should be captured in order to detect it with a certain precision. We also analyze the positions of the cameras from which the images should be captured in order to reach the optimal (minimal) number of training images. Furthermore, we show how synthetic training images can be created through an affine transformation in order to decrease the number of captured training images. A database is created and used for an in-depth analysis of omnidirectional object duplicate detection, as described in Section 3.4.1. The database contains images of several object classes taken from various points of view and distances.

Applications for object duplicate detection can increase the transmitted data from mobile to a server significantly. It can increase the energy consumption and traffic cost of mobile phones. Therefore, it is important to reduce the submitted information size. A novel algorithm is proposed to reduce this information by making use of 3D capturing device.

As an application of robust detection, object duplicate detection in video content is presented. Objects are detected in video content iteratively in order to compensate for 3D view variations, illumination changes and partial occlusions. Given a query image with the object of interest, the proposed system retrieves key frames with duplicates of that object. Due to invariance of the object duplicate detection approach to minor appearance changes, the retrieved frames usually contain also variations from the object of interest. Therefore, the retrieved objects are considered as iterative queries to retrieve object duplicates with larger variations. For example, given the frontal view of a car as the initial query, the iterative query mechanism can retrieve the back side of the car if intermediate views of the car are available in the video clip.

The remaining sections of this chapter are organized as follows. Omnidirectional detection is handled in Section 4.2 and the extension with syntectic training images is described in Section 4.3. Object duplicate detection in video content is introduced in Section 4.5. Finally, we conclude the chapter with its summary in Section 4.6.

4.2 Omnidirectional detection

4.2.1 Introduction

In this section, we determine the number of training images of the object which are necessary to detect it with a certain accuracy.

Imagine a scenario where a surveillance system should detect a knife, a gun, a stolen bag or any other suspicious object. For a reliable system, the object duplicate detection should achieve a certain F-measure, even if these objects are shown from an arbitrary direction. However, it is difficult to create a system which can detect an object shown in arbitrary directions and distances in test images with a satisfactory precision, if only a limited set of training images is available and they are not selected so as to cover all possible variations.

We evaluate the performance of our proposed object duplicate detection algorithm (details in Section 3.3) with respect to angle and size deviations between training and test images in order to derive requirements for omnidirectional object duplicate detection. We use our database containing 15 classes of objects with high diversity, and we set the target precision as F-measure values of 0.7 and 0.8. However, the analysis method presented in this section is a general framework that can be used for any other database and any target precision to estimate a required number and conditions of the training images to obtain the target precision for the given database and detection algorithm.

4.2.2 Related work

Typically, most object duplicate detection methods contain the following steps: feature extraction, object representation, and matching. In this section we review representative object duplicate detection methods based on these steps, considering view-point variation.

Most of the object representations consider the objects in the 2D image space only, e.g [Sivic and Zisserman, 2006], where descriptors are extracted from local affine-invariant regions and quantized into visual words, reducing the noise sensitivity of the matching operation. But, since real-world objects are inherently 3D, a higher performance can be achieved using 3D models. However, the creation of complete 3D models requires a large number of images from all possible angles, which may not be feasible in real applications. Despite this difficulty, interesting solutions have been proposed for multi-view retrieval of objects from a set of images or video. In [Rothganger *et al.*, 2004] a full 3D model of the object is used for the detection of objects in video sequences. The approach of [Sivic *et al.*, 2006] uses key-point tracking to retrieve different views of the same object and to group video shots based on the object's appearance. The tracked object is then used as an implicit representation of the 3D structure of the object to improve the reliability of object duplicate detection. This method has proven to be more effective when compared to a query with a single image, but it requires that all the relevant aspects of the desired object are present in the query shot, which limits its applicability. Our approach makes an attempt towards 3D modeling, while keeping the efficiency of 2D processing, using a graph model to represent the 3D spatial information.

Different scale and orientations of the objects can be considered for better performance of the feature extraction and the salient region detection tools. Mikolajczyk [Mikolajczyk *et al.*, 2005] analyzed different affine region detectors considering different angles and distances. However, region detectors are just one small part of an object duplicate detection method and it is also necessary to consider orientation- and scale-invariance of the feature descriptors used for local feature matching. In the original papers presenting the SIFT [Lowe, 2004] and SURF [Bay *et al.*,

Table 4.1 — Summary of the database of images taken under challenging conditions.

class	#images	#samples	condition	planar
bag	103	3	monotone	3D
bike	50	4	various features	3D
body	43	3	textured cloths	3D
book	76	13	shiny	2D
building	57	3	artificial	3D
can	60	3	small	3D
car	37	3	shiny	3D
workbook	49	9	text	2D
face	39	3	deformable	3D
logo	130	4	small	2D
motor	31	3	shiny	3D
newspaper	67	17	text	2D
poster	39	7	text,images	2D
shoes	40	3	small	3D
stone	32	3	monotone	3D
total	850	≥ 3	various	2D,3D

2006b] feature descriptors, their effectiveness was analyzed for different viewpoints and scales. However, the performance of the algorithms in robustness by the proposed method evaluated for a very limited number of planar objects. This chapter shows significant improvement in robustness by the proposed method over the original SIFT description. As we will show later, the number of training images can be decreased, using optimal camera positioning. We also use a significantly larger database with more than 80 different objects. Moreover, we describe an optimal strategy for training image creation for full omnidirectional detection.

4.2.3 Experimental setup

The evaluation methodology used in this chapter, such as ROC, PR and F-measurement, is described in Appendix A.

In this chapter the database which was previously introduced in Section 3.4.1, is used for evaluation of robustness of our object duplicate detection method for still images. This database contains 850 images and it was created in order to evaluate the object duplicate detection method from different viewpoints and distances, as described in Table 4.1. It consists of ten 3D and five 2D object classes: bag, bicycle, body, face, shoes, stone, can, car, building, motor, poster, logo, newspaper, book and workbook. Each of the 15 classes contains at least three different objects of the same class (e.g. Adidas bag, Oakle bag, North Face bag), and all together 85 objects are contained in the database with ten sample photos per object. For more details about the database, refer to Subsection 3.4.1.

4.2.4 Results and analysis

The results of the analysis for our database are shown in Figures 4.1. Using only the original training images, the F-measure starts to decrease considerably when the object size in the test

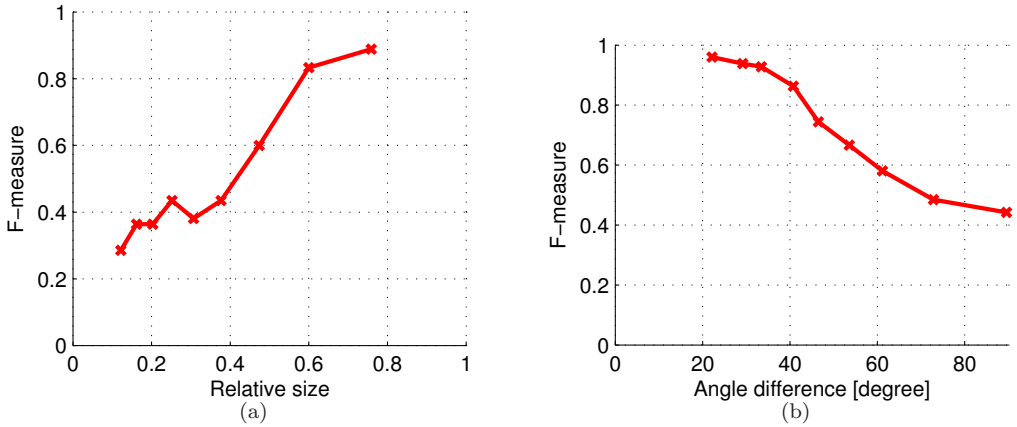


Figure 4.1 — (a) F-measure for various relative object sizes in test mage in comparison to training image. (b) F-measure for various viewing angle differences between training and test images.

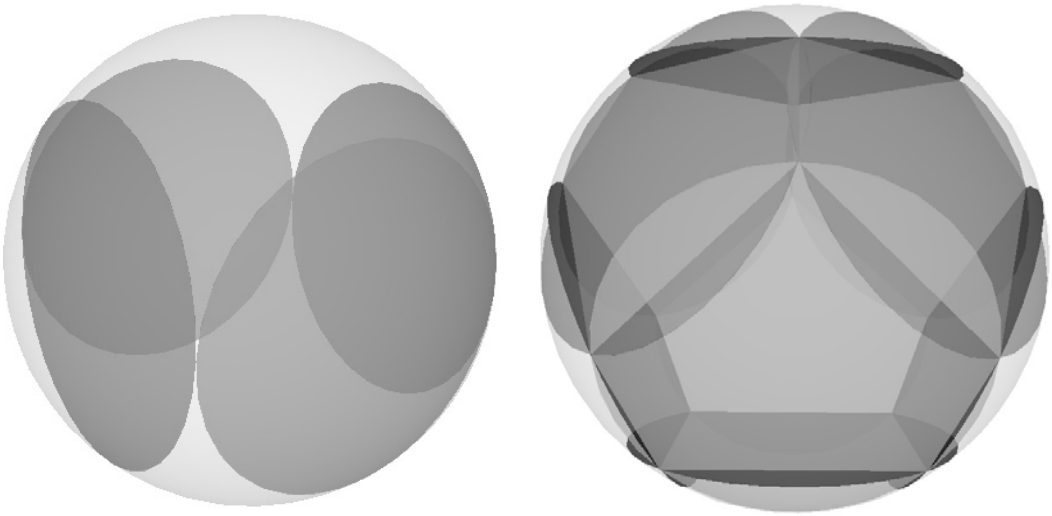
image is less than 60% of the original size, or when the viewing angle differs by more than 40° from the training image. It is interesting to see that even when the viewing angle differs by 90°, the F-measure still remains around 0.45 whereas conventional methods based on 2D object representations would get almost zero. This shows that target objects have to be considered as convex 3D objects rather than planar 2D objects, which can lead to an increased tolerance to angle deviations. When the relative size is too small (e.g. below 30%), the performance is degraded significantly (e.g. below 0.4 in terms of F-measure).

Based on these results, it is possible to derive the minimum number of training images and the required angles and distances of the objects in the images from our database in order to achieve a certain F-measure value. In order to achieve an F-measure of at least 0.8 by using an object model trained with only one training image, the test images may differ from the training image up to an angle of ±45° and up to a relative size of 59%.

If we want to detect at least 80% of the test objects in our database for all possible rotations around a single axis, four training images are enough because one image can cover 90° of 360° as shown in Figure 4.2.

On the other hand, if we consider omnidirectional object duplicate detection in the 3D space, it is necessary to solve the problem of positioning disks (or, equivalently, cameras) to cover a sphere. More precisely, the problem is to find the minimum number of congruent disks that cover a sphere for a given radius of the disks, or conversely, to find the minimum radius of the disks to cover a sphere for a given number of disks so that every point of the sphere belongs to at least one disk. Although a general solution of the problem for an arbitrary number of disks is not available, the solutions for some cases has been given by Fejes Tóth [Fejes Tóth, 1972]. For different numbers of cameras, the required coverage radius of the cameras is shown in Table 4.2 [Hardin *et al.*, 1997].

Therefore, to cover a sphere with disks having a radius of 45°, 10 training images are enough, if the positioning of the cameras is as shown in Figure 4.2, where radius of 45° is assumed to achieve at least 0.8 for F-measure. The positions of the cameras in this case are shown in Table 4.3.



(a) (b)

Figure 4.2 — Required camera positions for planar (a) and omnidirectional (b) object duplicated detection with a F-measure of 0.8, where each circle represents a camera and its covering area.

Table 4.2 — Solutions for the problem of covering a sphere with *#cameras* congruent, overlapping disks. The second row shows the radius of the disks in degree. Each disk can represent a camera and its coverage angle.

#cameras	4	5	6	7	8	9
radius [*degree*]	70.53	63.43	54.74	51.03	48.14	45.88

#cameras	10	11	12	13	14	15
radius [*degree*]	42.31	41.43	37.38	37.07	34.94	34.04

Figure 4.3 combines the previous results and shows how many training images in our scenario are necessary for a certain F-measure. If we want to detect at least 70% of the test objects contained in images taken from arbitrary directions, based on the previously discussed estimations, it is allowed to have angle differences up to 50° and thus it is sufficient to use 8 images for training, as shown in Figure 4.3.

4.2.5 Conclusion

A novel methodology of determining the number of training images for viewpoint-invariant detection is presented in this section. We define problem of determining the number of training images as a problem of covering a sphere with congruent disks. Assuming that a specific object is detected with a sufficiently high precision, for some angle and scale factors, the following conclusions can be drawn from our discussion:

Table 4.3 — Ten 3D coordinates of the centers of the disks which cover a unit sphere when the radius of the disks are 45°.

Axis/Cam	1	2	3	4	5
x	−0.521	0.449	−0.577	−0.526	0.526
y	0.576	0.879	0.684	−0.345	0.345
z	−0.630	0.160	0.446	0.778	−0.778

Axis/Cam	6	7	8	9	10
x	0.468	−0.904	0.957	−0.013	0.142
y	0.072	−0.357	−0.290	−0.594	−0.970
z	0.881	−0.236	−0.015	−0.805	0.198

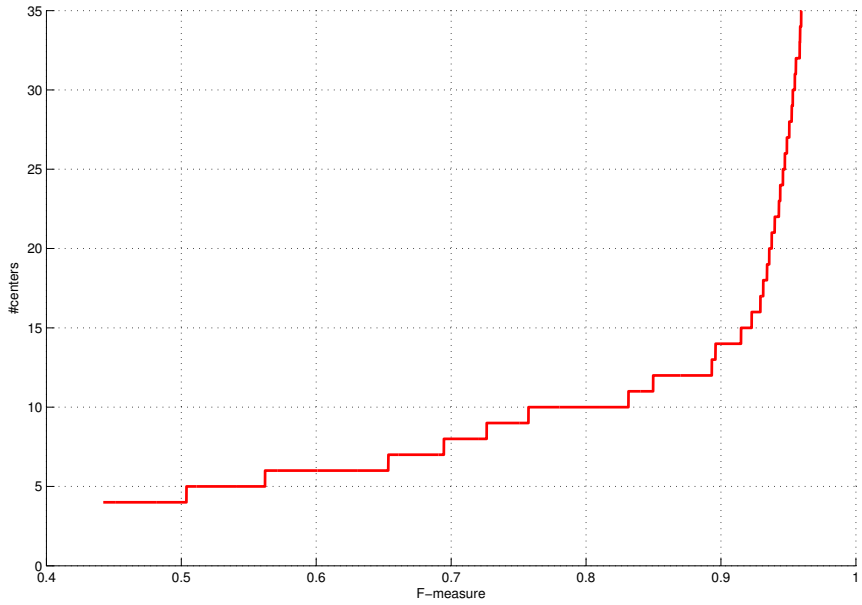


Figure 4.3 — F-measure vs. number of cameras needed for omnidirectional object duplicate detection.

- Four training images are enough for 3D object duplicate detection from planar view point.
- Eight and ten training images are necessary for full omnidirectional detection by keeping F-measure above 0.7 and 0.8 respectively.

4.3 Synthetic training images

4.3.1 Introduction

One way to improve the accuracy of the object duplicate detection algorithm is to generate synthetic images using affine transformations on the original training images to enrich our database. Affine transformed synthetic images can create automatically, without any user interaction and it can improve the imperfect affine invariant feature detection. To generate synthetic images we scaled

the original images by s^n, where $n \in [0 \dots 10]$ and s is a parameter that was set to 0.85 in our experiments. The scaling factor is exponential, assuming, that training on the generated image will have the same detection accuracy over shifted viewpoint. Synthetic rotated images are generated by scaling the training image in the horizontal direction by s^n. In object detection we consider only one direction of rotation, assuming that the results for other directions do not change much. All synthetic images are used as training images for object duplicate detection in the experiments.

4.3.2 Realted work

Considering various scale and orientation, visual features are evaluated in several articles [Bay *et al.*, 2006b; Lowe, 2004; Mikolajczyk *et al.*, 2005]. However, in [Morel and Yu, 2009], the authors go a step further and create new features which include several affine transformed generated images as input. This paper shows significant improvement over the original SIFT description. As we will show later, the number of generated images can be decreased, using optimal camera positioning. We also use a significantly larger database with more than 80 different objects. Moreover, we describe an optimal strategy for training image creation for full omnidirectional detection.

4.3.3 Experimetal setup

Database and evaluation methodology of this section is the same than it was in the previous Section 4.2.3. The detailed evaluation methodology is described in Appendix A and the database is introduced first in Section 3.4.1.

4.3.4 Results and analysis

We explore the benefit of synthetic training images generated through affine transformations.

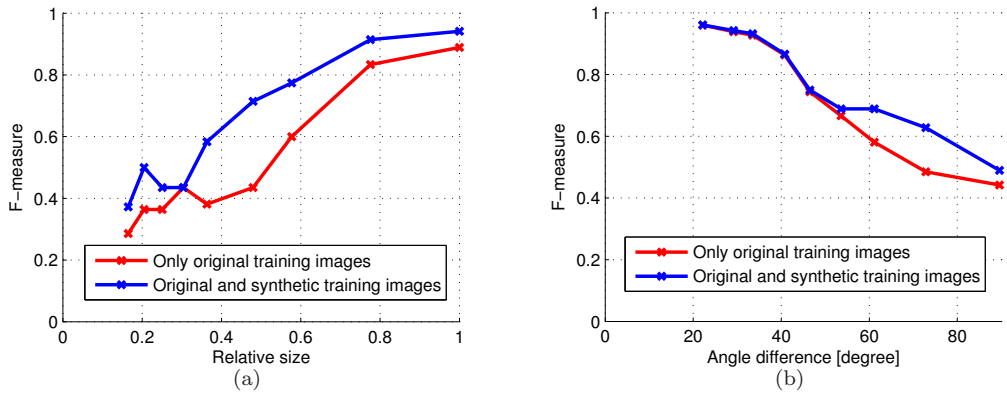


Figure 4.4 — (a) F-measure vs. relative size of the object in the test image. (b) F-measure vs. viewing angle difference between the training and test images.

Figure 4.4 shows that, adding synthetic training images generated by affine transformations leads to a significant improvement of the F-measure (up to 0.2) over the whole range of size

deviations. However, for the angle deviations the F-measure improves significantly (up to 0.15) only for angles larger than 50°. These results are expected since the scaling of the training images needed for different sizes causes much smaller distortions in the synthetic images than the affine transformations required for the different angles.

Based on these results, it is possible to derive the minimum number of training images and the required angles and distances of the objects in the images from our database in order to achieve a certain overall F-measure value. Using synthetic training images, the relative size improves from 59% to 50% for an F-measure of 0.8, while the angle difference does not change.

Figure 4.5 combines the previous results and shows how many training images in our scenario are necessary for a certain F-measure using automatically generated syntectic images. If we would like to detect at least 70% of the test objects contained in images taken from any direction, based on the previously discussed estimations, it is allowed to have angle differences up to 50° and thus it is sufficient to use eight images for training. However, if we use synthetic training images, seven images are enough as shown in Figure 4.5.

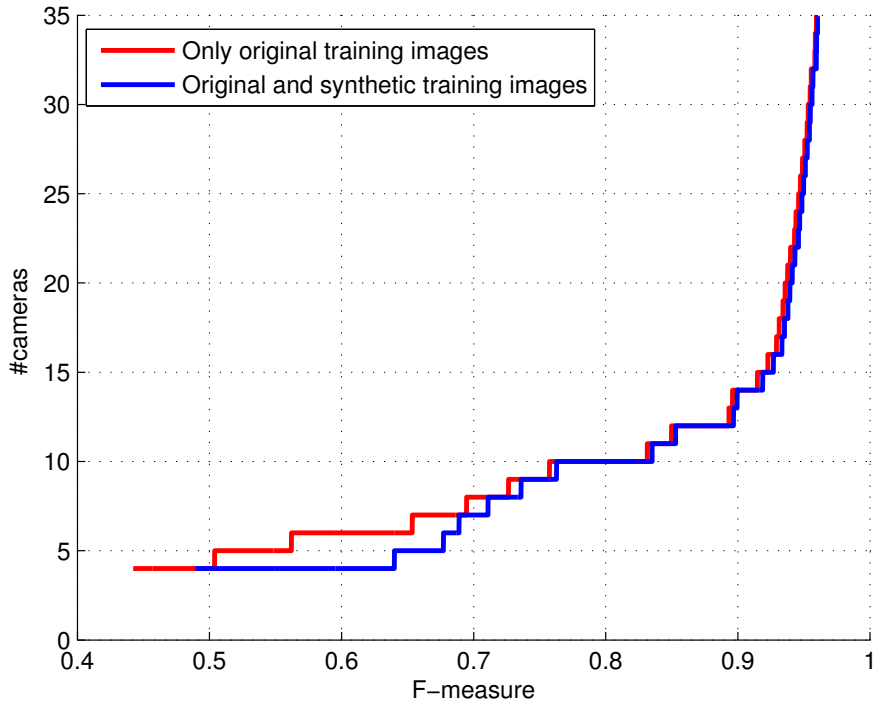


Figure 4.5 — F-measure vs. number of cameras needed for omnidirectional object duplicate detection using only original or additionally synthetic training images.

4.3.5 Conclusion

In this work, we explored the benefit of synthetic training images generated through affine transformations. Assuming that a specific object is detected with high enough precision, for some angle and scale factors: Synthetic rotated training images improve mainly the accuracy of omnidirectional

detection when the viewpoints in the training and test images are largely different, however synthetic scaled training images improve significantly the detection of objects from different distances in all cases.

4.4 Stereo view for mobile platform

4.4.1 Introduction

Mobile image search and retrieval has become increasingly popular and several commercial applications and services have been developed, including Kooaba, Google Goggles and Snaptell. *Kooaba*[*] is based on SURF features and it detects specific objects, such as posters, CDs, DVDs, books, and game covers. *Snaptell*[†] detects objects through local features and Accumulated Signed Gradient matching. Therefore Snaptell is robust to changing viewpoints and partial occlusions. *Goggles*[‡] is the most recent commercial application from Google. It can detect logos, book covers, artworks, places and wines using visual and GPS information.

These application can increase the transmitted data from mobile to a server significantly. It can increase the energy consumption and traffic cost of mobile phones. Therefore, it is important to reduce the submitted information size. In this Section a novel algorithm is proposed to reduce this information by making use of 3D capturing device.

4.4.2 Stereo GOD

A novel object duplicate detection method in stereo images is proposed based on the GOD method. The overall size of SIFT features is comparable to the image size itself. This can cause delay in processing of the object duplicate detection method on mobile and tablet platforms, due to the slow network bandwidth. Therefore, new feature compression approaches have been developed, such as PCA-SIFT [Ke and Sukthankar, 2004] and CHOG features. To compress the features, principal component analysis (PCA) decreases the dimension of feature, from 128 bytes SIFT feature to 36 bytes. CHOG quantizes the gradient information for feature extraction to reach around 10 times smaller features per image by reducing 256 possible values (1 byte) in each histogram bins to 8 different values (3 bit). Stereo GOD approach uses stereo images to decrease the number of feature by selecting only important features. The proposed method, first, evaluates GOD algorithm, using CHOG features on the stereo images and eliminates features which were not matched. Therefore the number of features can be reduced, by keeping the accuracy of the detection algorithm. In the last step, selected features are matched to the features extracted from a query image by GOD algorithm. This idea can be used to eliminate unimportant features. As it is explained in Section 4.4.4, this method shows 25 times smaller size of overall features compared to the original image size.

[*] http://www.kooaba.com/
[†] http://www.snaptell.com/
[‡] http://www.google.com/mobile/goggles/

4.4.3 Experimetal setup

The detailed evaluation methodology is described in Appendix A.

A new 2D and 3D image dataset was created and different object duplicate detection methods were systematically evaluated using collected images. The dataset contains images of particular objects, both indoor and outdoor (such as buildings, dolls, shoes, books and products), captured manually from different view points and distances, and in different lighting conditions, using different types of cameras (such as professional, mobile, 3D and web camera). All angles, distances and luminance are precisely measured and images are appropriately annotated with these information. Furthermore, the dataset contains 3D images: stereo images which include left and right view from the point of capturing, and images from Microsoft Kinect accompanied with depth map.

Objects

The novel dataset contains 16076 pictures for five 3D object classes as shown in Figure 4.6: buildings, dolls, shoes, books and products. Images of shoes, dolls, books and products were taken in indoor environment, while buildings were considered as outdoor objects. Each class contains at least three sample objects. The number of images for each of the objects with respect to the type of camera used for capturing is presented in Table 4.4.

Creation

Four types of cameras are used for capturing images: professional, mobile phone, 3D and web camera. Cannon EOS 400D is used as a professional camera which captures images with a resolution of 3888×2592 pixels. Mobile phone Samsung Galaxy I9000 on Android OS platform is used for capturing mobile images with a resolution of 2560×1920 pixels. 3D Fujifilm Finepix REAL 3D W1 is used to collect stereo images, left and right views. A resolution of images is 3648×2736 pixels. Microsoft Kinect is considered as a web camera, which collects images with a resolution of 640×480 pixels accompanied with depth maps. All images were captured using automatic camera adjustments, such as exposure or white balance.

The following describes how images were captured from different points of view. Each sample of indoor objects was fixed on a rotational chair equipped with protractor (with a resolution of $10°$) and images were captured by rotating the chair along with precisely measuring the angle distance from the initial position of the chair. To take images of objects at different distances, the chair was moved along a ruler on the floor which was used to precisely measure the distance. For outdoor objects, Google Maps was used to determine the points of shooting. From the center of an object, concentric circles and straight lines are drawn as in Figure 4.7. Intersections between straight lines and one of the circular lines represent shotting points with $10°$ of difference between two consecutive ones. The scale of the map is used to measure the distance from the object, as shown in Figure 4.8. The professional camera was used in two setting modes: automatic and manual, for taking images of outdoor objects.

Figure 4.6 — Samples of the different 3D object classes within the dataset used in evaluations.

Figure 4.7 — Google Maps was used to determine points of shooting for outdoor objects with respect to different angles. Intersections between straight and circular lines represent shotting points with 10° of difference between two consecutive ones.

Table 4.4 — Summary of a novel database of images taken under challenging conditions. This table shows the conditions under which images of a particular object were captured: the numbers of angles, distances and lighting conditions. Also, it provides the numbers of images taken using specific cameras.

Object	#angles	#distances	#lighting conditions	Professional camera	Mobile phone camera	3D camera	Kinect camera	#images per object
building1	19	14	4	264	132	264	0	660
building2	19	12	4	248	124	248	0	620
building3	17	14	4	248	124	248	0	620
building4	19	13	4	256	128	256	0	640
book1	36	11	4	188	188	376	376	1128
book2	36	11	4	188	188	376	376	1128
book3	36	11	4	188	188	376	376	1128
doll1	36	11	4	188	188	376	376	1128
doll2	36	11	4	188	188	376	376	1128
doll3	36	11	4	188	188	376	376	1128
product1	36	11	4	188	188	376	376	1128
product2	36	11	4	188	188	376	376	1128
product3	36	11	4	188	188	376	376	1128
shoes1	36	11	4	188	188	376	376	1128
shoes2	36	11	4	188	188	376	376	1128
shoes3	36	11	4	188	188	376	376	1128
$\sum$				3272	2764	5528	4512	16076

Conditions

For each sample of indoor objects, images were taken from 36 equidistant angles (10° of difference between consecutive points of view) and 11 different distances (50-350 cm) from the center of the object. The same strategy was followed for samples of outdoor objects, where images were taken from at least 17 angles with 10° of difference between consecutive points of view and at least 12 different distances (20-200 m) from the object. The number of images for outdoor objects was limited due to natural obstacles in outdoor environments at the point of shooting, such as trees or other buildings.

Further, different lighting conditions were considered, as shown in Figure 4.9 for buildings, dolls and shoes. Images of indoor objects were taken in four different lighting conditions using a homogenous light with specific luminance: dark (2 lux), bright (23 lux), very bright (154 lux), and side-light (108 lux). The first three indoor lighting conditions were made using homogenous light directed towards the top of the object, while side-light was directed towards one side of the object. Outdoor images were taken under sunny ($>$ 300 lux), evening/morning (150-300 lux), twilight (3-150 lux) and night ($<$ 3 lux) lighting conditions. The exact luminance of scene was measured using a light-meter at the point of capturing an image. Different lighting conditions include not

Figure 4.8 — Google Maps was used to determine points of shooting for outdoor objects with respect to different distances. A straight line and a scale of the map were used to measure the distance from an object.

only the luminance, but also the influence (e.g. color) of the automatic adjustment in a particular camera, such as exposure or white balance. Also, blurring can happen when the environment is dark, due to the shaking hand.

Use of the novel database

The proposed novel dataset is suitable for several research areas:

- *Object duplicate detection in images* in which the goal is to detect the presence of a target object in a set of images based on an object model created using visual features from a small set of training images.

- *Image quality assessment* in which subjective or objective assessment is performed to judge the influence of the automatic adjustments done by different cameras and in different lighting conditions on the quality of the produced images. Also, the influence of the different coding schemas on the image quality can be explored using the novel dataset.

- *3D image processing* in which, for example, evaluation of different techniques for depth map generation can be applied on the novel dataset, as well as, the quality assessment of 3D images from different aspects.

However, in this work, we are focusing on object duplicate detection task, which will be further elaborated in the following sections.

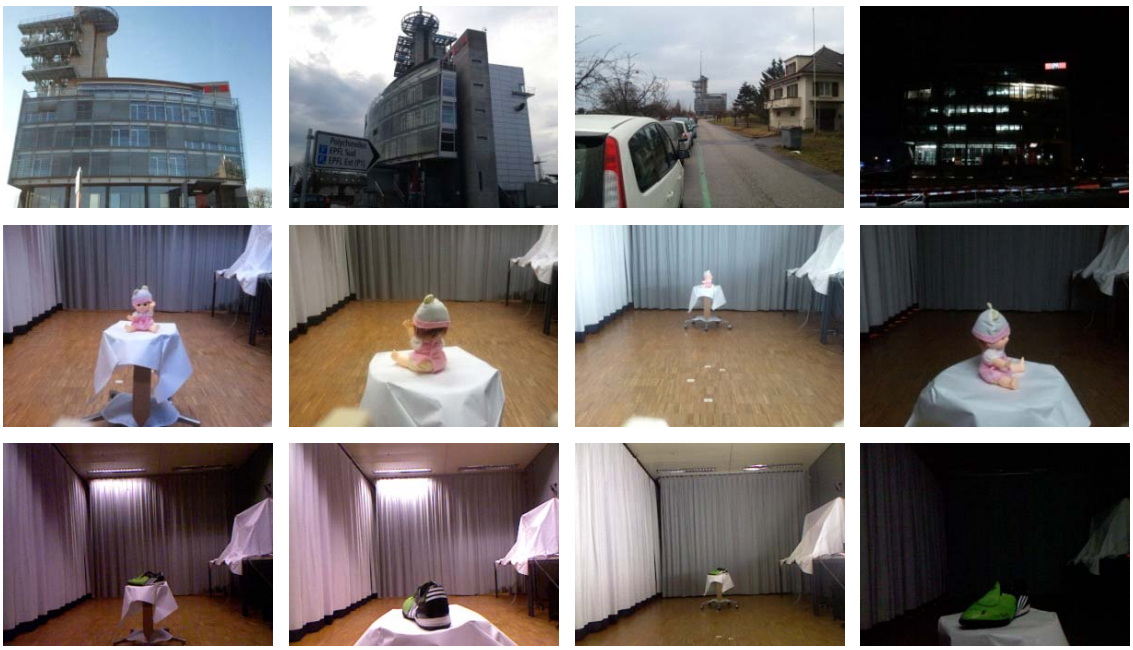


Figure 4.9 — Samples for three objects in diverse viewing conditions within the dataset used in evaluations. Images in each row were taken using a professional camera, a mobile phone camera and a web camera, respectively.

4.4.4 Results and analysis

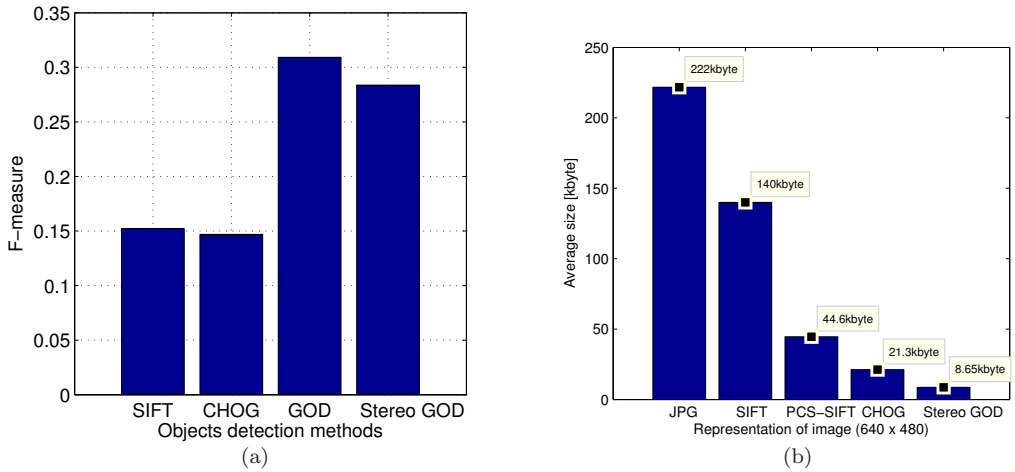


Figure 4.10 — Performance difference (a) and size deviation (b) for different representation and detection of image.

In the near future several photo and mobile cameras will be able to capture 3D images, therefore object recognition using 3D images will become more important. The proposed database

contains 3D images, especially stereo and depth images. In this Section we demonstrate the Stereo GOD algorithm, which saves bandwidth and therefore decreases detection time and energy. The algorithm compares left and right images from stereo images and takes into account only common features in both images, which reduces the number of features passed by mobile device. The overall size of features of images in the proposed database is compared against different algorithms, as shown in Figure 4.10. The overall size of features in the proposed method is 2.5 times less than number of CHOG features in 640×480 images. The Stereo GOD algorithm outperforms all other methods, except GOD algorithm which is 10% more accurate. The idea of feature selection using stereo image can be adapted in many applications, such as object retrieval in video.

4.4.5 Conclusion

In this work, we explored the benefit of stereo cameras on mobile phone. Novel database was captured containing more than 16000 well annotated images. All angles, distances and luminance are precisely measured and images are appropriately annotated with these information. Furthermore, the dataset contains 3D images: stereo images which include left and right view from the point of capturing, and images from Microsoft Kinect accompanied with depth map.

A novel algorithm, Stereo GOD is designed for perform object duplicate detection on low bandwidth devices, such as mobile phone. This algorithm was evaluated on the proposed database. Stereo GOD approach uses stereo images to decrease the number of feature by selecting only important features. This method shows 25 times smaller size of overall features compared to the original image size, which saves large amount of bandwidth, transmission time and energy while mobile application sends query to the server application.

4.5 Iterative object duplicate detection in video

4.5.1 Introduction

In the past few years, sharing photos and video in social networks has become very popular. The number of video content grows rapidly in social networks like YouTube*, DailyMotion† and Blip.tv‡. For instance, 20 hours of video are uploaded to YouTube every minute [Youtube, 2010] and DailyMotion contains over 975 million video clips [DailyMotion, 2010]. Therefore, fast video retrieval systems, which allow users to search desired video clips in an efficient way, are becoming increasingly important. Most of the existing popular video search engines rely on text-based annotations and manual descriptions of the video content. However, recent developments have shown that content-based video retrieval (CBVR) using visual features extracted from the video content itself provides a promising alternative. Most of the CBVR approaches rely on the query by example paradigm where a user is required to provide a query video which is compared to other video in a database. Since a representative query video may not be available, either a single query image or an object of interest may be used to describe the scene.

*http://www.youtube.com
†http://www.dailymotion.com
‡http://www.blip.tv

In this section, we propose an efficient 3D object-based video retrieval system which requires only a single query image in order to overcome the constraint of previous approaches requiring a large number of query images. This work is based on the algorithm presented in the previous chapter. The key idea is to apply this method iteratively to the video database in order to compensate for 3D view variations, illumination changes and partial occlusions. Given a query image with the object of interest, the proposed system retrieves key frames with duplicates of that object. Due to invariance of the object duplicate detection approach to minor appearance changes, the retrieved frames usually contain also variations from the object of interest. Therefore, the retrieved objects are considered as iterative queries to retrieve object duplicates with larger variations. For example, given the frontal view of a car as the initial query, the iterative query mechanism can retrieve the back side of the car if intermediate views of the car are available in the video clip. Therefore, the novelty of this work comes from an iterative approach for object duplicate detection in video and a key frame extraction method which is specialized for this application.

4.5.2 Related work

Over the last decade, tremendous attention has been given to developing systems which are able to automatically analyze and index video clips, and retrieve its relevant parts. The proposed system is related to different research fields including visual analysis, shot-boundary detection and key frame extraction, and 3D object modeling. Here we will discuss only the most related work.

Studies on shot-boundary detection are typically based on extracting visual features (color, edge, motion, and interest points) and comparing them among successive frames. Lienhart in [Lienhart, 1999] and Cotsaces *et al.* in [Cotsaces *et al.*, 2006] provide comparison by taking different algorithms into account, and by measuring their ability to detect the type and temporal extent of the transitions. Grana *et al.* [Grana *et al.*, 2005] propose a two-step iterative algorithm, unique for both cuts and gradual transitions detection, in the presence of fast object motion and camera operations. Huang *et al.* [Huang *et al.*, 2008] propose an approach based on local keypoint matching of video frames to detect abrupt and gradual transitions. By matching the same objects and scenes using contrast context histogram in two adjacent frames, the method decides whether there is shot change or not.

After shots are segmented, key frames can be extracted from each shot. A key frame is the frame which can represent the salient content of the shot. Depending on the content complexity of the shot, one or more key frames can be extracted from a single shot. In [Zhang, 1997], Zhang *et al.* propose a method how to use multiple visual criteria to extract key frames, such as color, motion or shot based features. Wolf proposed a motion based approach for key frame extraction [Wolf, 1996]. The method consists of computing the optical flow for each frame in the shot and calculating a simple motion metric based on the optical flow. Finally, key frames are selected at the local minima of motion over the time. In our case, choosing only one frame of the shoot as the key frame seems to be the natural choice in order to decrease computational complexity, as all the rest of the frames in the shot can be considered to be logical and continuous extensions of that frame.

Text-based engines search for the desired clips by matching the keywords entered by the user

against a large set of annotations. However, this approach always needs extensive time and intensive man-power to annotate and describe those video clips. Furthermore, ambiguous search results usually occur due to the fact that different people usually have different interpretations on the video contents.

Recent developments in multimedia technology introduced content-based video retrieval as a new research field in contrast to the traditional text-based approaches for indexing and searching. This approach highly reduces the time and man-power required during the indexing and annotation phases. It performs video retrieval according to the similarity measurements between video shots based on low-level visual features. Visual features generally contain a large amount of information which is hard to capture using keywords [Smeulders *et al.*, 2000]. Although, it has been demonstrated that this approach can give good results, user is usually required to provide a query video clip, which may not be always possible.

Initially most CBVR approaches were based on global representations such as color, texture and motion characteristics. Recent approaches have turned towards object-based representations to facilitate the search of objects or regions of interest within a video. However, the retrieval of objects is a challenging problem because an object's visual appearance may change considerably due to variations in viewpoint, illumination, deformation and partial occlusions. Different approaches have been developed to handle these multiple visual aspects of an object. Sivic and Zisserman [Sivic and Zisserman, 2003] proposed a method where descriptors are extracted from local affine-invariant regions and quantized into visual words (more details can be found in Section 2.4.7), which reduces the noise sensitivity of the matching. Inverted files are used to match the video frames to a query object and retrieve those which are likely to contain the same object. However, the latter considers only 2D objects, such as posters, signs, ties, and the front side of clocks, and does not take into account real 3D objects. An extension of this approach by Sivic *et al.* [Sivic *et al.*, 2006] uses keypoint tracking to retrieve different views of the same object and to group video shots based on the object's appearance. The tracked object is then used as an implicit representation of the 3D structure of the object to improve the reliability of the object recognition. This method has been proven to be more effective than a query with a single image, but it requires that all relevant visual aspects of the desired object are present in the query shot, which limits its applicability. The system by Rothganger *et al.* [Rothganger *et al.*, 2004] is based on a rigid 3D model of the object of interest which is created from several instances of the object within a single shot. The 3D object model is matched to a shot by reprojecting it to the 2D video. While 3D models provide a more reliable description of a real-world objects, their creation requires a large number of images from various angles, which may not be available during a query. In this section, we propose an algorithm which uses only one image of the query object, while keeping good detection rate.

4.5.3 Proposed algorithm

In this Section, we present our solution for object duplicate detection in video clips.

The main innovation is to apply object duplicate detection in an iterative way by considering retrieved objects within key frames as additional queries beside the initial query object. The system architecture which consists of two phases, namely key frame extraction and iterative object

duplicate detection, is illustrated in Figure 4.11. In the proposed system, a user is able to search for video clips containing the desired object by providing a snapshot or photo of the object.

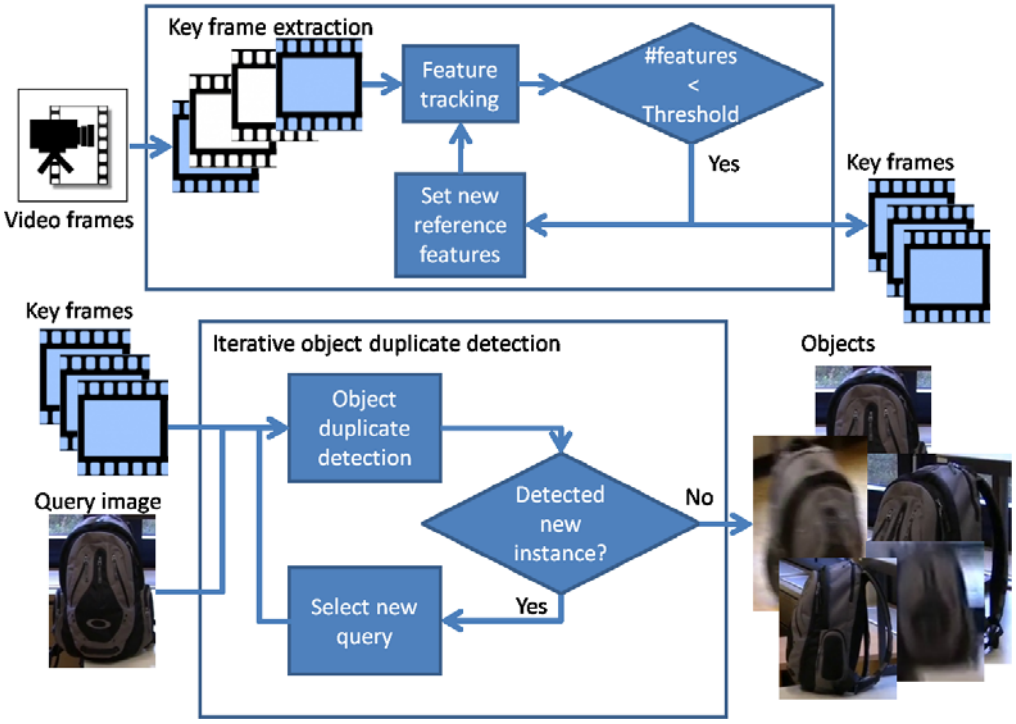


Figure 4.11 — Overview of the system for object duplicate detection in video. It runs an iterative search for the target objects on extracted key frames.

Key frame extraction

The goal of the key frame extraction module is to detect representative frames of the video which contain a considerable change in comparison to previous key frames, e.g. significantly different views or a completely different object or scene. This definition is specific for object duplicate detection in video and differs from that typically used for key frames of video shots.

Our approach is to detect stable and robust salient points in the video and to track them using optical flow. Harris corner detection is applied to detect salient points and an iterative Lucas-Kanade method [Bouguet, 2002] is used to compute the optical flow. If a tracked point disappears in further frames or moves very close to another salient point (≤ 5 pixels), it is considered as lost and not tracked anymore. If the ratio between the number of the tracked points of the previous and the current frame decreases more than a threshold ($T_k = 0.5$), then this frame is saved as a key frame. Otherwise the point tracking continues. This method results in several key frames extracted from a given video, which contain significant changes of the object or scene.

Iterative object duplicate detection

The goal of the object duplicate detection module is to detect the presence of a target object in video based on an object model created from a query image. Duplicate objects may vary from their perspective, have different size, or be modified versions of the original object.

Our approach for object duplicate detection, previously described in Section 3.3, is robust to minor appearance changes, viewpoint variations, and partial occlusions due to the combination of invariant local features and a graph model which describes their relationships. Given a training (query) image, features are extracted and a spatial graph model is used to improve the detection accuracy. We use sparse features in order to resolve the localization problem efficiently. These features are robust to arbitrary changes in viewpoint. A spatial graph model is created for the object of interest, which considers the scale, orientation, position and neighborhood of features. To detect the presence of the object in a test image, the features are extracted and a graph matching algorithm is applied to match the created graph model to these features and derive a matching score. As a result, a match score matrix is produced, which represents the pair-wise comparison of training and test images.

The appearance of a 3D object in a video sequence may vary a lot due to different viewpoints and deformations. Therefore, the detection based on a single query image may fail at some point due to the large difference between the trained object model and the object present within the considered key frame. In order to solve this problem, we apply the object duplicate detection method iteratively, as shown in Figure 4.12.

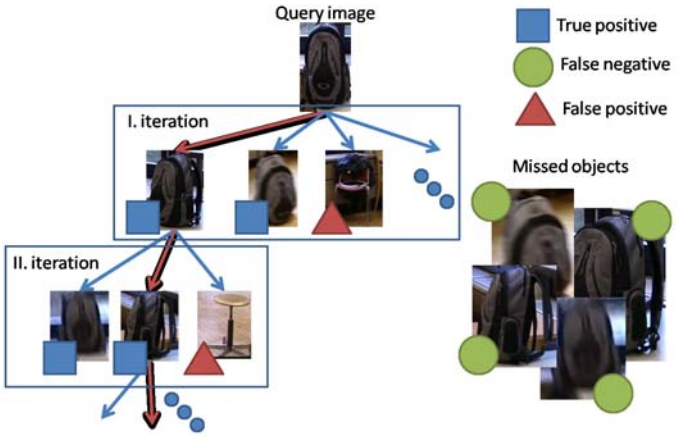


Figure 4.12 — Illustration of the iterative object duplicate detection on the key frames of the video. Predicted objects of an iteration are used as query objects in the next iteration.

In other words, we use a detected instance of the object, which may have a slightly different viewpoint from that of the current query image, as a new query image for the object duplicate detection of the next iteration. At each iteration, we randomly choose one of the objects for the next iteration. In the new iteration, object duplicates are searched only in the key frames that were not predicted to contain the object before.

4.5.4 Experimental setup

During the iterative detection, the precision rate decreases with the number of iterations (N), while the recall rate increases until the algorithm retrieves all key frames and the recall reaches 100%. We estimate the expected precision as a function of N, and a proper value of N is determined so that the expected precision remains higher than a target precision rate $T_{prec} = 0.75$.

The precision can be seen as a probability function, which determines the probability that a randomly selected object among the detected objects is true. If the object duplicate detection algorithm selects a false object as the next query, all the predicted objects will be false in the next iteration. Here, we assume that, in each iteration, the number of predicted objects is the same. Then, the expected precision E_{prec} can be written as

$$E_{prec}(N) = \sum_{i=1}^{N} \frac{1}{N} \cdot P_{odd} \cdot P_{it}^{i-1} \tag{4.1}$$

where P_{odd} and P_{it} are the precisions of the object duplicate detection algorithm for the first iteration and the subsequent iterations, respectively. These values can be obtained a priori based on the results in Section 3.5. They are dependent on the threshold parameter that is applied to the matching score between the object model and the query image. We set different values of the threshold for the first and subsequent iterations, i.e., $T_{odd} = 60$ and $T_{it} = 80$, because we target a higher precision for less reliable query objects selected among the key frames. As a result, we obtain $P_{odd} = 85\%$ and $P_{it} = 92\%$. By using these values, the maximum value of N satisfying the inequality $E_{prec} \geq T_{prec}$ is obtained as $N = 3$.

Video dataset

In order to evaluate the proposed method in video content, a video sequence was recorded in which a "bag" was chosen as the target object. In contrast to the movies used in [Sivic *et al.*, 2006], generating a new video enabled us to have the object in various challenging conditions, such as changes of the background and the distance from the object to the camera, different viewpoints, changes in the illumination of the room, and partial occlusions. Some examples of the key frames extracted from the video are shown in Figure 4.13. The movie was recorded in a resolution of 1440×1080 pixels, with a frame rate of 25 fps. It lasts 44 minutes, which results in 66000 frames overall. The object "bag" appears approximately during 40% of the total length of the sequence.

4.5.5 Results and analysis

The precision and recall (definition is in Appendix A) calculated on key frame level for each iteration are shown in Figure 4.14 as crosses. Each of the points within the figure represents the result of a single query which has been either manually or randomly selected. For the first iteration the query object corresponds to the manually selected one (the first image in Figure 4.13) and for the further iterations to those randomly selected from the retrieved objects. Depending on the selected query object, the performance for the same iteration varies.

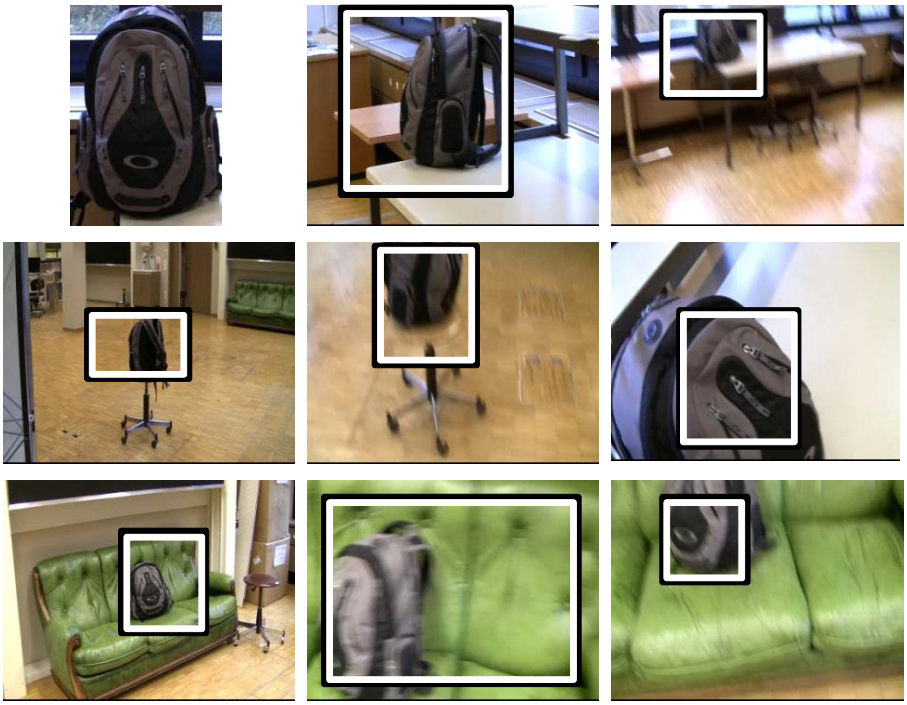


Figure 4.13 — Representative key frames containing the object "bag" extracted from the used video. The first image represents the manually selected and cropped training image. Detected objects are marked with bounding boxes.

The result shows that the first iteration has the highest precision and recall values and the precision and recall tend to decrease through the second and the third iterations. This can be explained by the fact that the bounding box of an automatically detected object is usually less precise than the manually selected. Therefore it may contain a considerable part of the background, which may lead to false detections in the next iteration. A more precise segmentation of the detected object could solve this problem and improve the performance.

The overall performance of the iterative object duplicate detection is calculated by considering all selected query objects for a certain iteration. This leads to circles in Figure 4.14 which shows the recall improvement due to the iterations. After the third iteration we obtain a precision of 89%, a recall of 34%. From these results, F-measure of 50% is calculated. As estimated before, the final precision rate remains higher than the lower bound which was set to $T_{prec} = 75\%$.

The dataset contains fast camera movements and thus some of the key frames are blurred. The viewpoints of the object also significantly vary across the whole video. However, our algorithm robustly detects instances of the target object which are blurred or acquired from different viewpoints. Figure 4.13 shows successfully detected instances of the target object with viewpoint changes of more than 90 degrees, partial occlusions of more than 50% and a large amount of blurring.

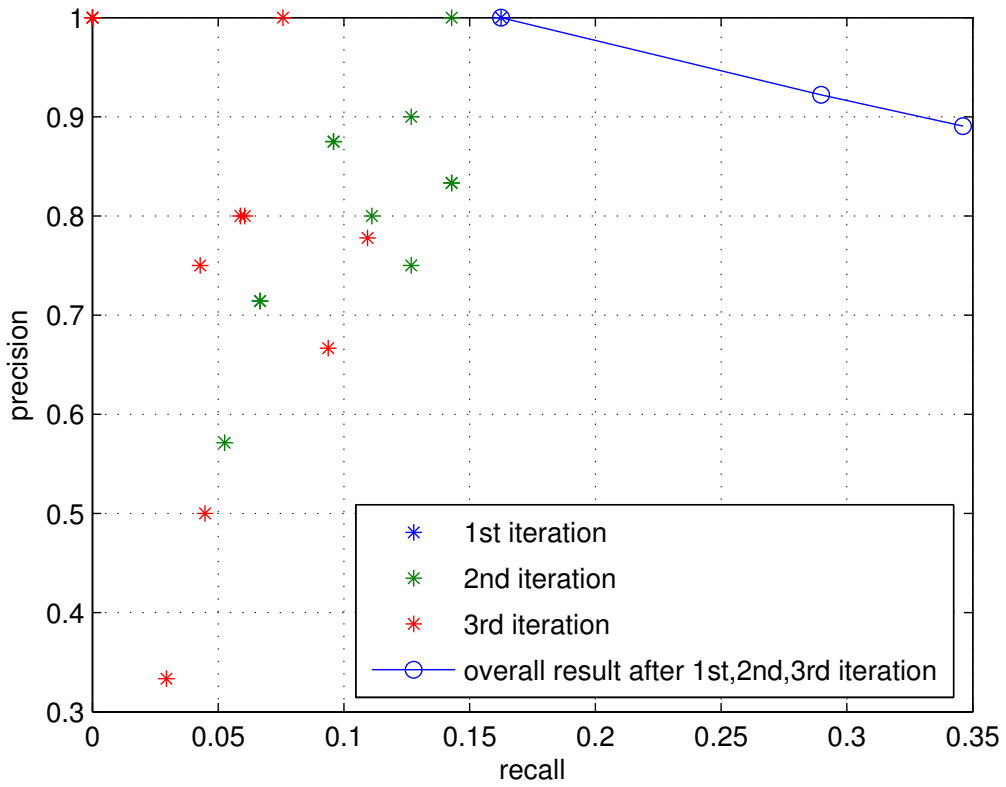


Figure 4.14 — Precision vs. recall for the individual queries (marked with "x") and the overall system (marked with "o") for the first three iterations.

4.5.6 Conclusion

In this Section, we have proposed a robust 3D object duplicate detection algorithm for video retrieval. An iterative procedure and a special key frame extraction have been introduced to detect robustly objects in different conditions, such as significant variations of viewpoint, size, lighting conditions and motion blur. The results show that the recall is improved by a factor of 2 using the iterative detection procedure in comparison to the non-iterative object duplicate detection algorithm, while the precision value is kept around 90%.

4.6 Chapter summary

Image and video retrieval systems are becoming increasingly important in many applications. Automatic video and image tag propagation, video surveillance, and high level image or video search are among some of the applications which require accurate and efficient object duplicate detection methods. In this chapter, we have analyzed and extended our robust graph-based object duplicate detection algorithm for 2D and 3D objects. The main idea of this chapter was to increase detection rate by automatically generated training images using synthetic images, video frames and stereo images. These methods are based on orthogonal ideas and can be combined to improve the final accuracy of the object duplicate detection algorithm. The experiments were performed on various classes of objects. The main conclusions that can be drawn from our experiments are:

- *Synthetic* rotated training images improve mainly the accuracy of omnidirectional detection when the viewpoints in the training and test images are largely different, however synthetic scaled training images improve significantly the detection of objects from different distances in all cases.
- The recall value of the object duplicate detection in *video* is improved by a factor of 2 using the iterative detection procedure in comparison to the non-iterative object duplicate detection algorithm, while the precision value is kept around 90%.
- Proposed Setero GOD algorithm shows 25 times smaller size of overall features compared to the original image size, which saves large amount of bandwidth, transmission time and energy while mobile application sends query to the server application.

Our method has shown to be robust in detecting the same objects in images and videos even if the images or videos with objects are taken from very different viewpoints or distances.

Everything has its beauty, but not everyone sees it.

Attributed to the Greek philosopher Confucius (*cerca* 551 B.C. — 479 B.C.)

Applications for large scale object duplicate detection

5

In the past few years sharing photos in social networks has become very popular. In order to easily explore a huge collection of images, they are usually tagged with representative keywords such as persons, events, objects, and locations. In this chapter, we present two large scale object duplicate detection algorithms for efficient object tag and geotag propagation. In both systems tags are propagated by training a graph based object model for each of the objects on a small tagged image set and finding its duplicates in a large untagged image set. Based on the established correspondences between these two image sets, tags are propagated from the tagged objects to the untagged images. The effectiveness of the proposed methods is demonstrated through a set of experiments on an image database containing various objects and landmarks.

5.1 Introduction

The past few years have witnessed an increasing popularity of social networks, digital photography and web-based personal image collections. A social network service typically focuses on building online communities of people who share interests and activities, or who are interested in exploring the interests and activities of others. Most social network services are web-based and provide a variety of ways for users to interact. They have become also a popular way to share and disseminate information, e.g. users upload their personal photos and share them through online communities asking other people to comment or rate their content. This has resulted in a continuously growing volume of publicly available photos, e.g. Flickr* contains over 3.6 billion photos [Wikipedia, 2010] and more than 2 billion photos are uploaded to Facebook† each month [Facebook, 2010].

*http://www.flickr.com
†http://www.facebook.com

As the popularity of social networking is on a constant rise, new uses for the technology are constantly being observed. To manage a large number of photos, tagging is one of the popular methods, which enables us to search our photo collections with keywords. However, tagging a lot of photos by hand is a time-consuming task. Users typically tag only a small number of the shared photos, leaving most of the other photos with incomplete metadata. This lack of metadata seriously impairs search, as photos without proper annotations are typically much harder to retrieve than correctly annotated photos. Therefore, robust and efficient algorithms for automatic tagging (or tag propagation) are desirable to help people organize and browse large collections of personal photos in a more efficient way.

In this chapter we propose two large scale object duplicate detection algorithms for object tag and geo tag propagation, respectively. We developed an interactive online platform which is capable of performing semi-automatic object tag propagation and tag recommendation. We consider object-based tagging in the system, therefore the query and the resulting objects are marked by its bounding boxes. We also consider the use of object duplicate detection for the propagation of geotags from a small set of images with location names (IPTC) to a large set of non-tagged images. The motivation behind this idea is that images of individual locations usually contain specific objects such as monuments, buildings or signs. The effectiveness of the approach is demonstrated through a set of experiments considering various locations.

The remaining sections of this chapter are organized as follows. Section 5.2 describes our approach for the interactive online platform and large scale object search for tag propagation and recommendation. Section 5.3 describes our approach for geotag propagation and discusses two application scenarios. Finally, Section 5.5 concludes the chapter.

5.2 Object duplicate detection for tag recommendation and propagation

5.2.1 Introduction

The main novelty of this section comes from the large scale evaluation of object duplicate detection in an interactive service system that minimizes the users' tedious and time-consuming manual annotation process. We propose an interactive online platform which is capable of performing manual or semi-automatic image annotation and tag recommendation for an extensive online database of images containing various object classes. Since the most salient regions in images usually correspond to specific objects, we consider object-based tagging. First, when the user marks a specific object in an image, the system performs object duplicate detection and searches images containing similar objects. Then, the annotation of the object can be performed in two ways, i.e., tag recommendation and tag propagation. In the tag recommendation mode, the system recommends tags for the object in the query image. The corresponding tags of all matched objects in the retrieved images are shown to the user who can then select appropriate tags among them. In the tag propagation mode, when the user enters a tag for the object, it is propagated to other images containing similar objects.

5.2.2 Related work

Tagging images is a very time consuming process and tagging objects in images even more. Therefore, it is necessary to understand and increase the motivation of users to annotate images. Ames and Naaman [Ames and Naaman, 2007] have explored different factors that motivate people to tag photos in mobile and online environments. One way is to decrease the complexity of the tagging process through tag recommendation which derives a set of possible tags and let the users select suitable ones. Another way is to provide incentives for the users in form of entertainment or rewards. The most famous examples are the ESP Game and Peekaboom, developed for collecting information about image content. The *ESP Game* [von Ahn and Dabbish, 2004] randomly matches two players who are not allowed to communicate with each other. They are shown the same image and asked to enter a textual label that describes it. The goal of a player is to enter the same word as the other player in the shortest possible time. In *Peekaboom* [von Ahn *et al.*, 2006] one player is shown an image and the other sees an empty black space. The first user is given a word related to the image, and the aim is to communicate that word to the other player by revealing portions of the image. Peekaboom not only collects the semantic description of images similarly to the ESP Games, but also the location information of the objects in the images. *LabelMe* is a web-based tool that allows easy image annotation and sharing of such annotations [Russell *et al.*, 2008]. Using this tool, a large variety of annotations are collected spanning diverse object categories (cars, people, buildings, animals, tools, etc.).

However, manually tagging a large number of photos is still a tedious and time-consuming task. Thus, automatic image annotation has received a lot of attention recently. It is a challenging task which has not been solved in a satisfactory fashion for real-world applications. Most of the solutions are developed for a specific application and usually consider only one tag type, e.g., faces, locations, or events.

Another application that combines textual and visual techniques has been proposed by Quack *et al.* [Quack *et al.*, 2008]. They developed a system that crawls photo collections on the internet and identifies clusters of images referring to common objects (physical items on fixed locations), and events (special social occasions taking place at certain times). The clusters are created based on the pair-wise visual similarities between the images, and the metadata of the photos in a cluster is used to derive a label for each clusters. Finally, Wikipedia* articles are attached to the corresponding images and the validity of these associations is checked. Philbin *et al.* [Philbin *et al.*, 2007] applied the Bag of Words method for detecting buildings in a large database. To resolve the problem of large database they use a forest of 8 randomized k-d trees as a data structure for storing and searching features.

Lindstaedt *et al.* [Lindstaedt *et al.*, 2008] developed a tag recommendation system for pictures of fruits and vegetables, called *tagr*. The system combines three types of information: visual content, text and user context. At first, it groups annotated images using global color and texture features. The user-defined annotations are then linked with the images. The resulting set of tags for visually similar images is then extended with synonyms derived from WordNet. When a user uploads an untagged image, it is assigned to one of the classes and corresponding tags

*http://www.wikipedia.org

are recommended to the user. In addition, this system analyzes tags which the user assigns to the images and returns the profiles of users with similar tagging preferences. This method has been proven to be effective to recommend appropriate tags images containing selected fruits and vegetables, but it cannot be applied to other classes of objects, which limits its applicability. Some other approaches for automatic image annotation consider only the context. Sigurbjörnsson and van Zwol [Sigurbjörnsson and van Zwol, 2008] developed a system which recommends a set of tags based on collective knowledge extracted from Flickr. Given a photo with user-defined tags, a new list of candidate tags is derived for each of the user-defined tags, based on tag co-occurrence. The lists of candidate tags are aggregated, tags are ranked, and a new set of recommended tags is provided to the user.

The tag propagation and recommendation are nowadays very important in environments such as social networks, since they enrich efficient information for grouping or retrieving images. The system proposed in this section provides this functionality in an interactive way. The novelty is the image annotation which is performed at the object level by making use of content based processing. It does not consider context, such as text or GPS coordinates, which may limit its applicability. This approach is suitable for any kinds of objects, such as trademarks, books, newspapers, and not just buildings or landmarks.

5.2.3 System overview

In this section, we present our method for object-based tag recommendation and propagation. The system performs tag propagation of marked and tagged objects. Image annotation is performed at the object level by outlining an object with a bounding-box. The system architecture is illustrated in Figure 5.1. In the following we will describe the offline and online parts of the proposed system separately.

5.2.4 Offline part

The goal of the offline processing is to preprocess uploaded images in order to allow efficient and interactive object duplicate detection. It starts by describing each image with a set of sparse local features. In order to speed up the feature matching, the features of all images are grouped hierarchically into a tree representation.

For a robust and efficient object localization sparse local features are adopted to describe the image content. Salient regions are detected using the Fast-Hessian detector [Bay *et al.*, 2006b] which is based on approximation of Hessian matrix detector. The position and scale are computed for each of the regions and will be used for the object duplicate detection (described in Section 5.2.5).

The detected regions are described using Speeded Up Robust Features (SURF) [Bay *et al.*, 2006b], which are approximations of Scale Invariant Feature Transform (SIFT) [Lowe, 2004]. They can be extracted very efficiently and are robust to arbitrary changes in viewpoints.

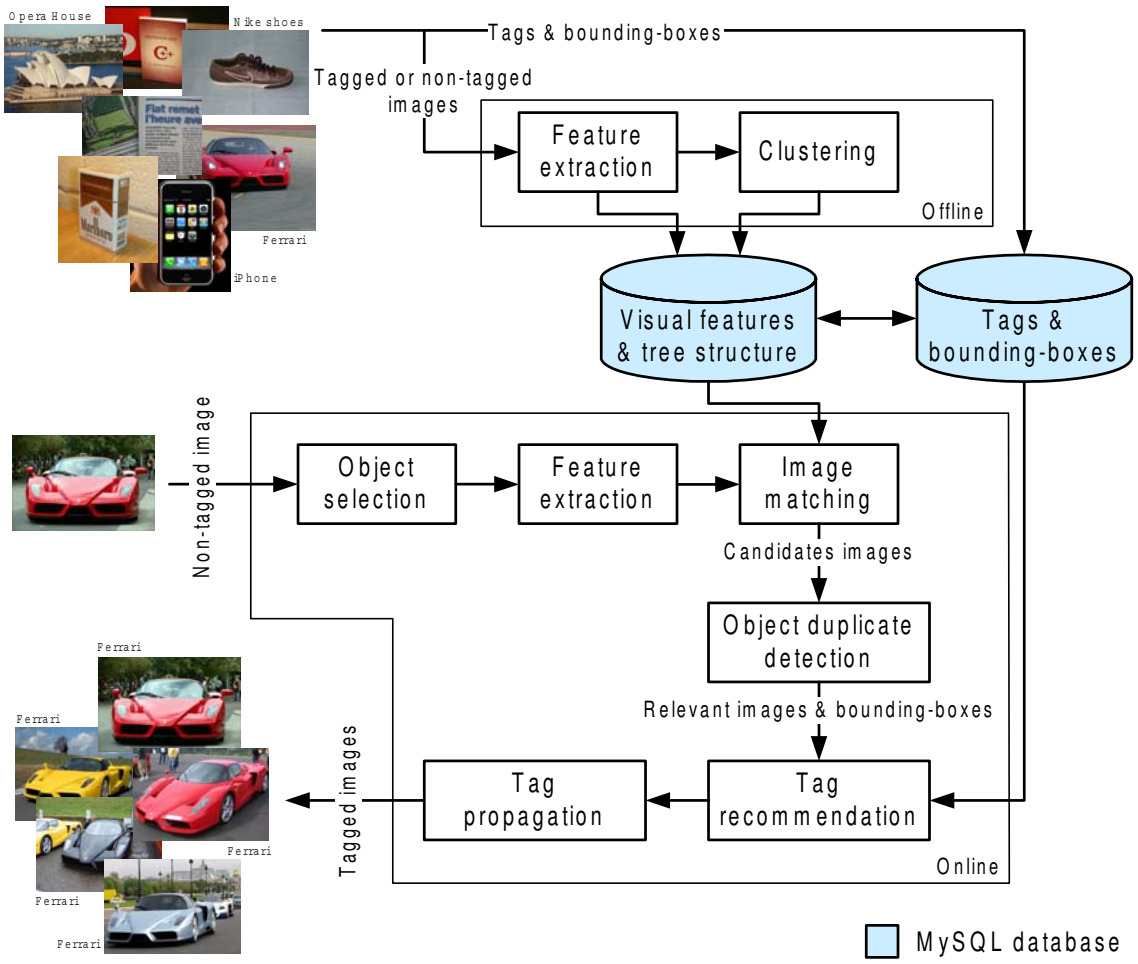


Figure 5.1 — Overview of the system for semi-automatic annotation of objects in images.

Clustering

For the object detection, features of a selected object have to be matched against the features of all the images in the database. Therefore, a fast matching algorithm is required to ensure interactivity of the application.

In our system, hierarchical k-means clustering is applied to group the features according to their similarity and to derive the vocabulary tree [Nister and Stewenius, 2006]. Then, a fast approximation of the nearest neighbor search can be used in the feature matching step. Within the tree, a parent node corresponds to the cluster centers derived from the features of all its children node and a leaf nodes corresponds to a SURF feature vector in the images. The clustering leads to a balanced tree with a similar depth for all the leaves.

In order to account for unequal importance of the visual words (i.e. nodes in the tree structure), a weight is assigned to each node. This weight is equivalent to the inverse document frequency (IDF) commonly used in text retrieval. The weight for node i is defined as

$$w_i = \log\left(\frac{N}{N_i}\right) \tag{5.1}$$

where N is the number of images in the database and N_i is the number of images which have features in the subtree when the i-th node is considered as a root of this subtree. The basic idea of IDF is that the importance of visual word is higher if it is contained in fewer images. Furthermore, the importance of a visual word i in relation to image j is considered using the term frequency (TF), which is defined as

$$m_{ij} = \frac{N_{ij}}{\sum_k N_{kj}} \tag{5.2}$$

where N_{ij} is the number of occurrences of visual word i in image j and the denominator is the number of occurrences of all features in image j.

Given this TF-IDF weighting scheme, the overall weight d_{ij} for visual word i in image j is given as

$$d_{ij} = m_{ij} \cdot w_i \tag{5.3}$$

which can be combined into a vector $\mathbf{d}_j$. This vector will be matched to the one extracted from the query image to compute the similarity in the image matching step described in Section 5.2.5.

The computational complexity of the offline phase may be even superpolynomial ($2^{\Omega(N)}$, where N is the number of features [Arthur and Vassilvitskii, 2006]), because in worst case, the k-means clustering algorithm can converge very slow. However, it is usually fast and accurate approximate algorithm without guarantee that it will converge to the global optimum.

5.2.5 Online part

The goal of the online processing is to detect duplicates of a selected object in the entire image database and then to propagate its tag to the other images containing the same object. Once a user marks a desired object in the image by placing a bounding-box around it, the system performs image matching by making use of local features and selects a set of candidate images which are

most likely to contain the target object then, the object duplicate detection is applied to detect and to localize the target object in the candidate images. The corresponding tags of all matched objects are suggested to the user and he/she can then select an appropriate tag among them. Once an object has been tagged, the user can ask the system to propagate it automatically to other images in the database.

Object selection

The user can annotate any photo in the database, which is either uploaded by himself/herself or by any other user. Images are annotated on the object level. The database used in this work covers a wide range of object classes, which will be described in more details in Section 5.2.6. Once the user chooses a photo which he/she wants to annotate, the user is free to label as many objects depicted in the image as he/she chooses. The user interface used in this work is shown in Figure 5.2. After clicking the button "add note", the user can place a bounding box around an object. This process is commonly used in many photo sharing services, such as FaceBook. When a user browses a particular image from the dataset, bounding boxes and tags which are previously entered by other users, are shown on the image. If there is a mistake in the annotation (either the outline or the text of the label is not correct), the user may either edit the label by renaming the object, redrawing the bounding box or deleting labels for the chosen image. Once the target object is correctly marked, the tag recommendation process can start.

Image matching

In order to speed up the object duplicate detection process, image matching that is less complex than the object duplicate detection, is applied a priori to select a set of candidate images which are most likely to contain the target object. By making use of the local features, target images can be efficiently distinguished from non-target images even if the target object is just a small part of an image.

Given the local features in the selected region in the query image and the vocabulary tree, a weighting vector $\mathbf{q}$ is computed in the same way as the weighting vector $\mathbf{d}_j$ for image j, as described in Section 5.2.4. Then, the matching distance between the query image and the j-th image of the database, s_j score, is computed as [Nister and Stewenius, 2006]:

$$s_j = \|\mathbf{q} - \mathbf{d}_j\| = 2 - 2 \cdot \sum_{\forall i: q_i \neq 0 \cap d_{ij} \neq 0} \frac{q_i \cdot d_{ij}}{\|\mathbf{q}\| \cdot \|\mathbf{d}_i\|}. \tag{5.4}$$

Only the images whose scores are less than a predefined threshold T_I are considered in the following object duplicate detection step.

The complexity of the search step for similar images is $O(n)$, where n is the size of the query image, as the feature extraction creates $O(n)$ features by making use of pyramids for detection of scale invariant features [Bay *et al.*, 2006b].

Object duplicate detection

The object duplicate detection step localizes the target object in the images returned by the image matching step. The outcome of this step is a set of predicted objects described through their bounding-boxes in each of the images.

Local features are used for object duplicate detection in [Lowe, 2004]. General Hough Transformation is then applied for object localization. Our object duplicates detection method is based on this algorithm and the detection accuracy is improved by using inverse document frequency. Inverse document frequency has been used for the similar purpose in [Sivic and Zisserman, 2006]. Descriptors are extracted from local affine-invariant regions and quantized into visual words, reducing the noise sensitivity of the matching. Inverted files are then used to match the video frames to a query object and retrieve those which are likely to contain the same object.

The detection and localization starts by matching the features in the selected region of the query image to those in the candidate image. Again the hierarchical vocabulary tree is used to speed up the nearest neighbor search. Matches whose distances are larger than a predefined threshold T_F are discarded.

In order to detect and to localize target objects based on these matched features, the general Hough transform [Ballard, 1981] is applied. Thereby, each matched feature in the candidate image votes for the position (center) and the scale of a bounding-box based on the position and scale of the corresponding feature in the query image. Since a unique feature may provide a more reliable estimate of the bounding-box, the vote of a feature is equal to its IDF value. This leads to a three dimensional histogram that describes the distribution of the votes across the bounding-box parameters (position and scale). To obtain the set of predicted objects the local maxima of the histogram are searched and thresholded with T_O.

The complexity of our method for object duplicate detection is $O(n)$, where n is the size of the query image, since the SURF feature extraction uses image pyramids for detection of scale invariant features [Bay *et al.*, 2006b] and the general Hough transformation has the same computational complexity, since we do not consider rotated objects in the database.

Tag recommendation

After selecting an object in the current image the user can press the "recommend" button to ask for suitable tags for this object (Figure 5.2). The system tries to find duplicates of the selected object using the algorithms described in the previous sections. If there is more than 50% overlap between the bounding boxes of the object selected by the user and the one found by the system, the found object is considered as a match. Tags for all matched objects are displayed to the user in form of tags and associated thumbnails. Duplicate tags may appear in the recommendation list, when multiple images contain visually similar objects accompanied by the same tag, which can be seen in Figure 5.2.

The user has then the choice to select one or more of the recommended tags or to provide his own tags for the object selected by him/her. At the end of this step, the bounding-box of the objects and the chosen/entered tags are stored in the database.

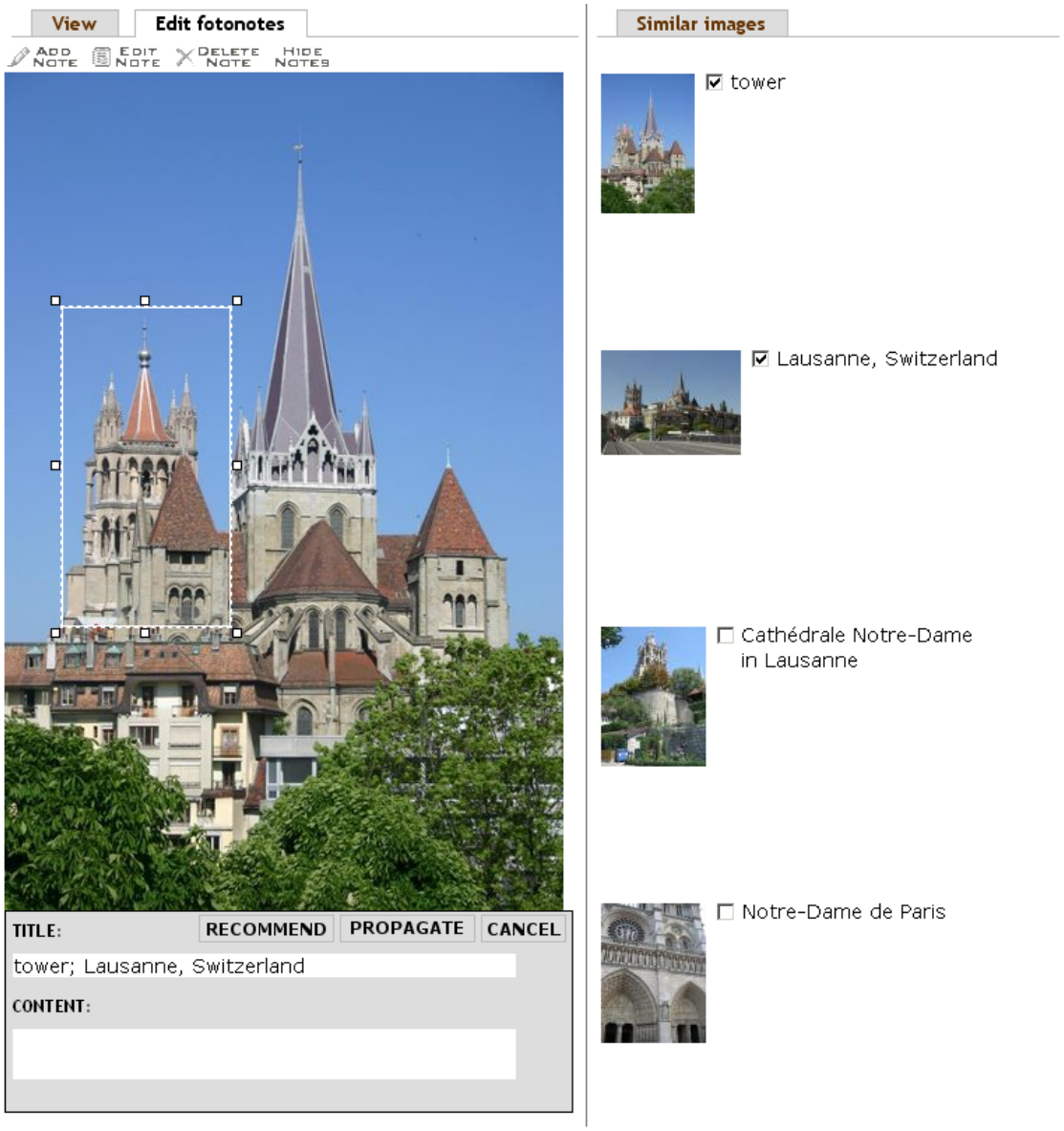


Figure 5.2 — Screenshot of the web application during the tag recommendation step. Based on the selected object the system automatically proposes tags from which the user can select suitable ones.

Tag propagation

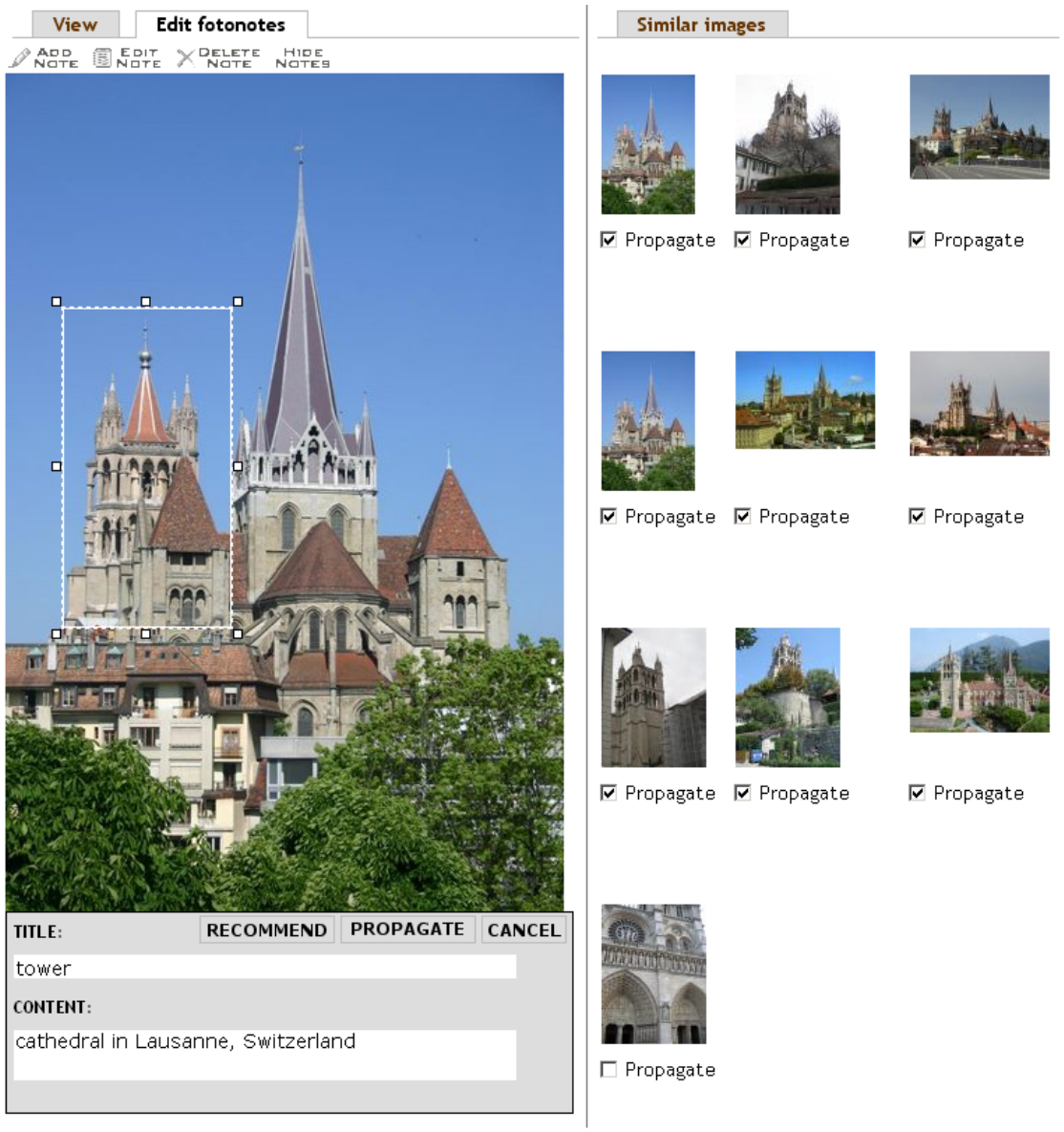


Figure 5.3 — Screenshot of the web application during the tag propagation step. The system automatically propagates the tagged object to the images in the database and asks the user to verify the result.

Once an object has been marked and tagged in a query image by the user, he/she can ask the system to propagate it automatically to the other images in the database by pressing the "propagate" button as shown in Figure 5.3. The system performs object duplicate detection and returns images containing object duplicates from the database to get confirmation from the user. If an object duplicate matches an already tagged object, the two bounding-boxes and sets of tags

have to be merged. The system can either ask the user or apply some heuristics automatically, to resolve the conflict and merge the two objects. In our system, since manually tagged objects are usually more reliable than automatically found ones for tag propagation, the new bounding-box is discarded but the two sets of tags are combined.

5.2.6 Experimental setup

Database

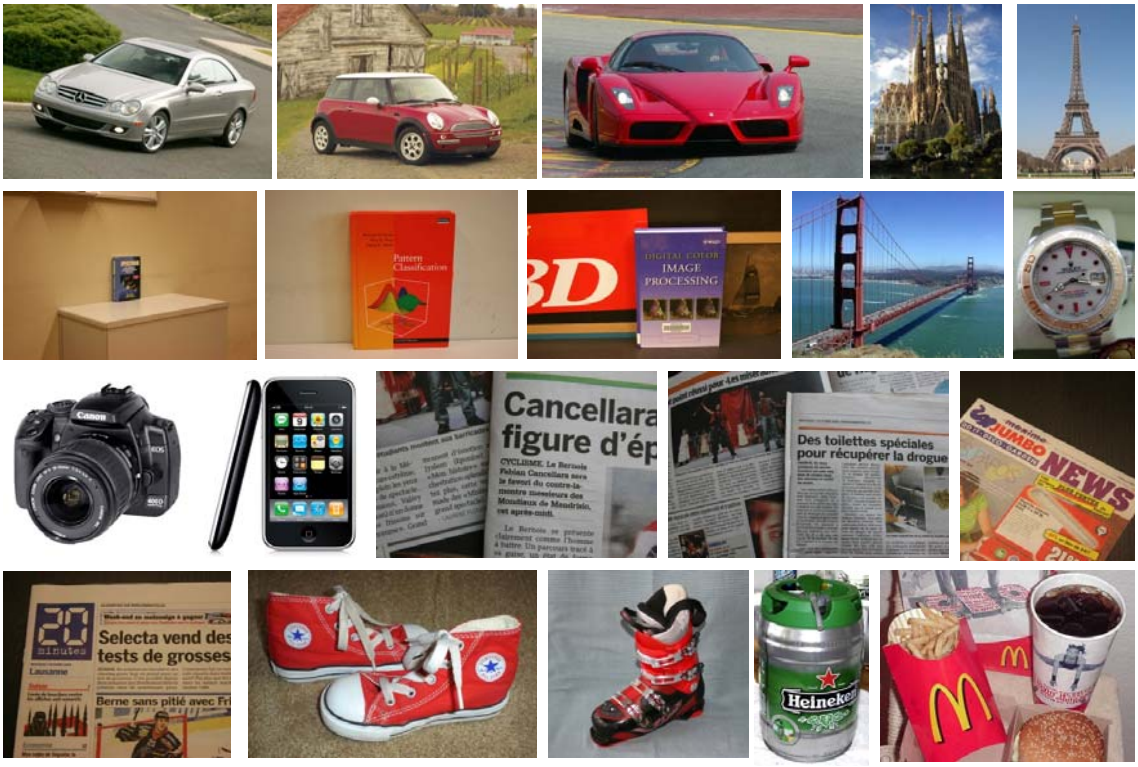


Figure 5.4 — Samples from the 160 objects in the dataset.

A new dataset was created in order to evaluate the tag propagation and tag recommendation methods. Part of the dataset is obtained from Google Image Search*, Flickr and Wikipedia by querying the associated tags for various classes of objects. The rest of the dataset is formed by manually taken photos of particular objects using digital camera Canon EOS 400D.

The resulting dataset consists of 3200 images: 8 classes of objects, 20 objects in each class, and 20 sample images of each object. Summary of the considered classes and some example objects are shown in Table 5.1. Figure 5.4 shows a single image for a single object from some of the 160 objects, while Figure 5.5 provides several images for three selected objects (e.g., Merrell Moab hiking shoes, Golden Gate Bridge, and Heineken trademark).

*http://images.google.com

Table 5.1 — Summary of the classes and some example objects

Classes	Example objects	#images
Cars	BMW Mini Cooper, Citroen C1, Ferrari Enzo, Jeep Grand Cherokee, Lamborghini Diablo, Opel Ampera, Peugeot 206, Rolls Royce Phantom	400
Books	"Digital Color Image Processing", "Image Analysis and Mathematical Morphology", "JPEG2000", "Pattern Classification", "Speech Recognition"	400
Gadgets	Canon EOS 400D, iPhone, Nokia N97, Sony Playstation 3, Rolex Yacht-Master, Tissot Quadrato Chronograph	400
Buildings	Sagrada Familia (Barcelona), Brandenburg Gate (Berlin), Tower Bridge (London), Golden Gate Bridge (San Francisco), Eiffel Tower (Paris)	400
Newspapers	MobileZone, Le Matin Bleu, 20 Minutes, EPFL Flash	400
Text	Titles, paragraphs and image captions in newspapers	400
Shoes	Adidas Barricade, Atomic Ski Boot, Converse All Star Diego, Grubin Sandals, Merrell Moab, Puma Unlimited	400
Trademarks	Coca Cola, Guinness, Heineken, McDonald's, Starbucks, Walt Disney	400
8	160	3200

Figure 5.5 — Selected objects for three different objects: Merrell Moab hiking shoe, Golden Gate Bridge (San Francisco), and Heineken trademark.

As it can be seen, images with a large variety of view points and distances, as well as with different background environments, are considered for each object. The dataset is split into training and test subsets. Training images were chosen carefully so that they provide frontal wide angle views of the objects depicted in the images, where the objects were selected using bounding-boxes. One sample image for each object is chosen as a training image. All other images from the dataset are test images. T_I and T_F thresholds for image preselection, feature matching and object detection were chosen considering previous works and experiments. T_O object detection threshold is determined in a new experiment as shows in the following section.

5.2.7 Results and analysis

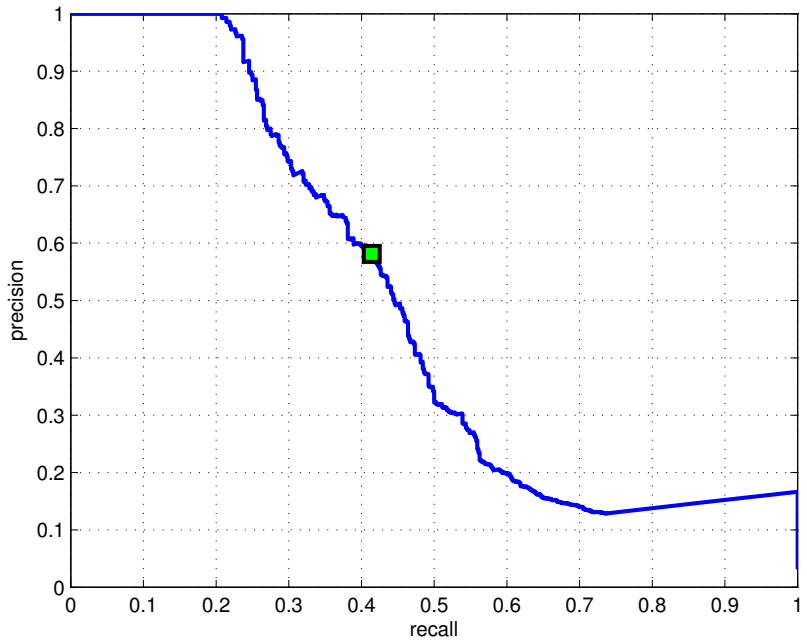


Figure 5.6 — Performance of the object duplicate detection as precision vs. recall (PR) curve averaged over all the classes.

Figure 5.6 shows the performance of the object duplicate detection in form of the average PR curve computed over all the test images. It provides a good visualization of the trade-off relationship between the precision and recall, which are inherent to any detection task as shown in Appendix A. The results show that when both measures are considered with equal importance, then the optimum of the F-measure is achieved at $R = 0.4$ and $P = 0.6$. However, the precision can be greatly increased if $R = 0.2$ is considered for the tag recommendation and propagation.

In order to determine the optimal threshold T_O for the object duplicate detection, the F-measures across the different thresholds has been computed. Figure 5.7 shows the threshold versus F-measure curve. The optimal threshold of 50 is chosen for the maximum F-measure of 0.49 and shown in all figures by green markers. The final F-measure averaged over the whole dataset is 0.49. The tag recommendation will be more supported than the tag propagation because the propagation is more sensitive to the performance of the object duplicate detection.

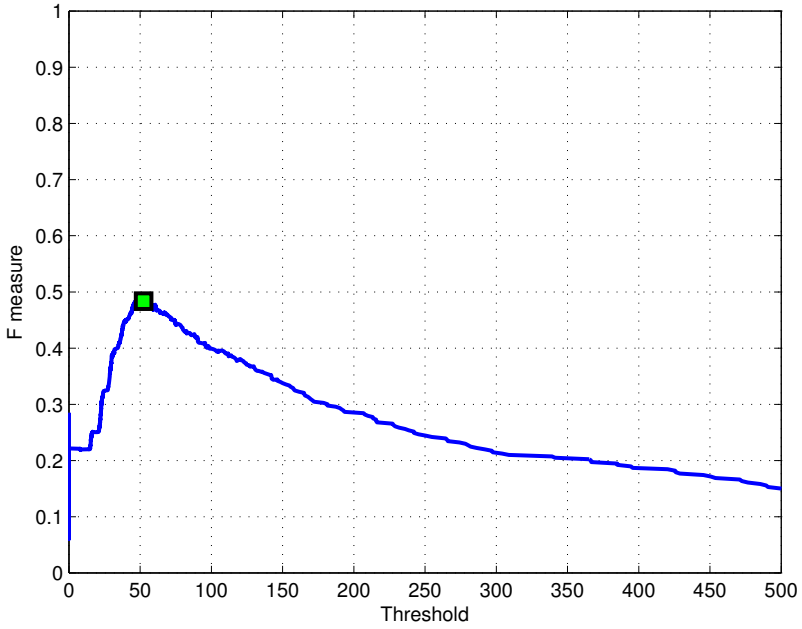


Figure 5.7 — Performance of the object duplicate detection as average F-measure vs. object duplicate detection threshold T_O.

Figure 5.8 compares the performance of ODD for different classes. The best performance is obtained for trademarks, thanks to the large number of salient regions. In the case of text or cover pages of newspapers, books or gadgets, the proposed tag propagation algorithm performs worse because the objects do not have enough discriminative features. Shiny or rotated objects, such as cars, shoes or buildings, are hard to detect due to the changing reflections and varying viewpoints.

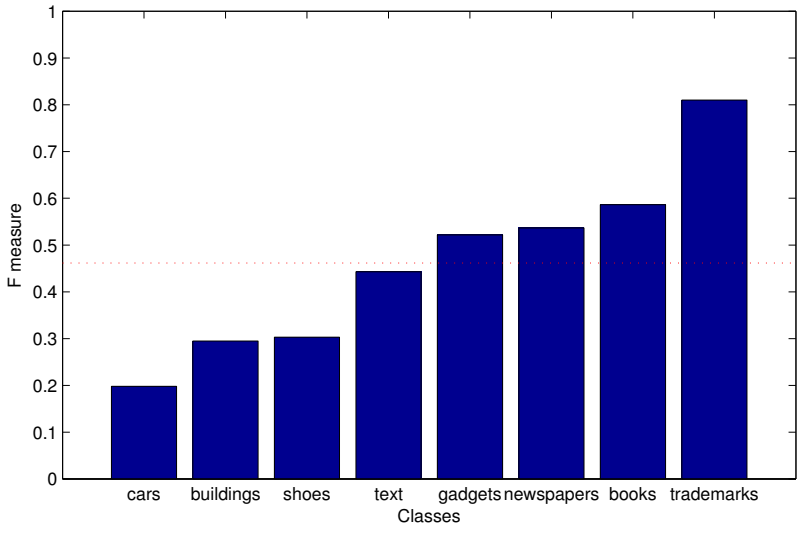


Figure 5.8 — Performance of the object duplicate detection in terms of F-measure for different classes.

5.2.8 Conclusion

In this work we have developed an application of large scale object duplicate detection for semi-automatic image tagging. After marking a desired object in an image, the system performs object duplicate detection in the whole database and returns the search results with images containing similar objects. Two levels of detection are proposed for efficient search in a large scale database. First, a fast image selection algorithm selects images which contain objects similar to the query object, then an accurate object duplicate detection algorithm provides their bounding boxes. The annotation can be performed through a tag recommendation process, in which the system recommends tags associated with the object in the images of the search results, or through a tag propagation process, when the user enters his/her tag for the object and it is propagated to the images in the search results.

The performance of the system has been assessed by evaluating the performance of the object duplicated detection step, since both tag recommendation and propagation rely on its outcome. It has been shown that the detection works reliably for salient objects such as trademarks, books, newspapers, and gadgets.

5.3 GPS retrieval from landmarks

5.3.1 Introduction

The most popular tags from photo sharing sites, such as Flickr, are mostly related to either persons, objects, events or locations. For a large portion of images, the association to their geographical locations provides a powerful cue for grouping and indexing. This is especially true for a large number of images depicting famous places from all over the world. Usually, the most salient region in such an image corresponds to a specific landmark or object to which a geotag is associated by a user. In this scenario, large scale object duplicate detection proposed in this section can be used effectively for propagation of geotags in order to reduce users' efforts for manual annotation, because it is robust in detecting the same object, as shown in the previous chapter. Untagged images can be automatically annotated based on the detection of the same object from a small set of training images associated with tags.

5.3.2 Related work

The proposed system is related to different research fields including visual analysis, geographic information systems, social networking and tagging. In particular, this section reviews existing studies on joint analysis of visual content and geographical context.

Google Image Search* is a popular image retrieval system, which is based on keywords extracted from the filename of an image, link text pointing to the image, and text adjacent to the image in web pages. Such contextual information may contain geographical locations related to the image, such as places, streets or cities. Besides, support for content based image search using visual features, such as color or image size, was recently added.

*`http://images.google.com`

Most of the photo sharing websites (e.g. Flickr, Picasa*, Panoramio†, Zoomr‡), provide information about where images were taken in form of maps or groups. This information is either provided by an external GPS sensor and stored as image metadata (Exchangeable Image File Format (EXIF) [Technical Standardization Committee on AV & IT Storage Systems and Equipment, 2002], International Press Telecommunications Council (IPTC) [International Press Telecommunications Council, 2009]) or manually annotated via geocoding. Our goal is to derive this information from the content of the image by comparing it to a small set of already tagged images.

Carboni *et al.* [Carboni *et al.*, 2006] have developed a web-based application called *GeoPix* that supports photo sharing through mobile phones. It allows mobile users equipped with camera phones to capture, annotate and finally upload images on the GeoPix web page. Important information about the locations assigned to images is provided by GPS devices or mobile phones.

Kennedy and Naaman [Kennedy and Naaman, 2008] presented a method to search representative landmark images from a large collection of geotagged images. This method uses tags and the geographical location representing a landmark. Then, it analyzes the visual features (global color and texture features), as well as SIFT to cluster landmark images into visually similar groups. This method has been proven to be effective to extract representative image sets of selected landmarks, but it cannot be applied to untagged images, which limits its applicability.

The recent work of Zheng *et al.* [Zheng *et al.*, 2009] finds frequently photographed landmarks automatically from a large collection of geotagged photos. They perform clustering on GPS coordinates and visual texture features from the image pool, and extract landmark names as the most frequent tags associated with a particular visual cluster. Additionally, they extract landmark names from the travel guide articles, such as Wikitravel§, and visually cluster photos gathered by querying Google Image Search. However, the test set they used is quite limited – 728 total images for a 124-category problem, or less than 6 test images per landmark. While they focused on mining landmark names and photos, we perform recognition of landmarks.

A pioneering paper in this area by Hays and Efros [Hays and Efros, 2008] proposed an algorithm called *IM2GPS* to estimate the locations of a single image using a purely data-driven scene matching approach. Given a test image, their approach finds the visual nearest neighbors in the database and estimates a geolocation of the image from those GPS tagged nearest neighbors. The estimated image location is represented as a probability distribution over the Earth's surface. However, the IM2GPS approach showed low recognition accuracy due to low-level features. While IM2GPS uses a set of more than 6 million training images, its general applicability is inconclusive because the performance was verified only on 237 hand-selected test images. Another drawback is limited availability of GPS coordinates associated to images. Our approach differs from their method in the way that we focus more on recognizing specific locations (landmarks), instead of geographic scenes and areas (such as forest or savanna). We aim at achieving high recognition rates considering the problem as a high-level object duplicate detection task, rather than a low-level image matching task used in IM2GPS.

*http://picasa.google.com
†http://www.panoramio.com
‡http://www.zooomr.com
§http://www.wikitravel.com

Crandall *et al.* [Crandall *et al.*, 2009] also considered the problem of estimating the geographic locations of photos. In addition to the visual features, they used the spatial distribution of popular places where photos were taken considering GPS coordinates. They found representative images for popular cities and landmarks by matching the SIFT interest points among the photos and considering temporal information, as photos taken in a short period of time are often different shots of the same landmark. In contrast, we target landmark matching by using a graph model, which imposes spatial constraints between SIFT features and thus improves the accuracy of the image matching.

Another application that combines textual and visual techniques has been proposed by Quack *et al.* [Quack *et al.*, 2008]. They developed a system that crawls photos on the internet and identifies clusters of images referring to a common object (physical items on fixed locations), and events (special social occasions taking place at certain times). The clusters are created based on the pair-wise visual similarities between the images, and the metadata of the clustered photos are used to derive labels for the clusters. Finally, Wikipedia* articles are attached to the images and the validity of these associations is checked. Gammeter *et al.* [Gammeter *et al.*, 2009] extends this idea towards object-based auto-annotation of holiday photos in a large database that includes landmark buildings, statues, scenes, pieces of art, with help of external resources such as Wikipedia. In both [Quack *et al.*, 2008] and [Gammeter *et al.*, 2009], GPS coordinates are used to pre-cluster objects which may not be always available.

Most of the recent systems rely on GPS coordinates to derive geographical annotation, which is not available for the majority of web images and photos, since only a few camera models are equipped with GPS devices. Furthermore, a GPS sensor in a camera provides only the location of the photographer instead of that of the captured landmark, which may be up to several kilometers away. Therefore, the GPS coordinates alone may not be enough to distinguish between two landmarks in a city. Describing landmarks through location names rather than GPS coordinates is not only more reliable but also more expressive. A recent study by Hollenstein and Purves [Hollenstein and Purves, 2010] indicated that geotagging should follow the way people actually describe locations, i.e. it is more convenient to use: Belgrade - Church of Saint Sava, rather than: latitude 44.798083, longitude 20.46855. Therefore, there is growing interest in the research community to infer geographic locations of the scenes in photos based on visual and text features.

5.3.3 System overview

In this section, we present our solution for geotag propagation between images. The main innovation is the combination of object duplicate detection for accurate and reliable large scale geotag propagation. The system architecture is illustrated in Figure 5.9. It contains two functional modules, each of which has a specific task: object duplicate detection and tag propagation [Ivanov *et al.*, 2010a; Vajda *et al.*, 2010b]. The object duplicate detection is based on the method described in Section 3.3.

The system takes a small set of training images with associated geotags to create the corresponding object (landmark) models. These object models are used to detect object duplicated in a set

*http://www.wikipedia.org

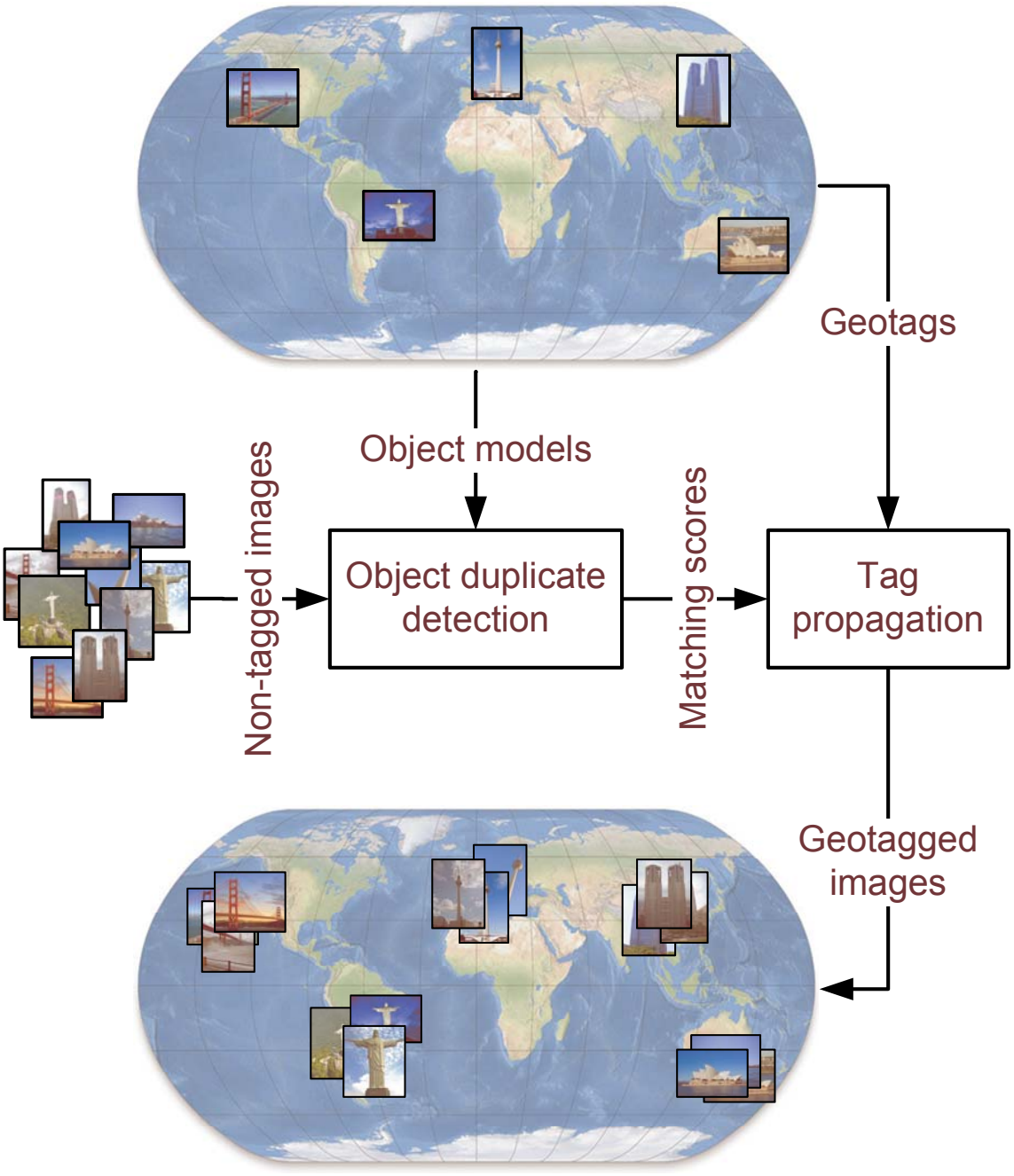


Figure 5.9 — Overview of the system for geotag propagation. The object duplicate detection is trained with a small set of images with associated geotags. The created object (landmark) models are matched against non-tagged images. The resulting matching scores serve as an input to the tag propagation module, which propagates the corresponding tags to the untagged images.

of untagged images. As a result, matching scores between the models and the images are obtained. According to the scores, the tag propagation module makes decisions about which geotags should be propagated to the individual images.

Object duplicate detection

The goal of the object duplicate detection module is to detect the presence of a target object in an image based on an object model created from training images. Duplicate objects may vary from their perspective, have different size, or be modified versions of the original objects after minor manipulations, as long as such manipulations do not change their identity. The basic idea of applying object duplicate detection for geotag propagation is that images of places typically depict distinctive landmarks (buildings, mountains, bridges, etc.) which can be considered as object duplicates.

The adopted approach is described in Section 3.3. Given a set of training images, features are extracted and a spatial graph model is created for each of the objects. We use sparse features in order to resolve the localization problem efficiently. These features are robust to arbitrary changes in viewpoints. A spatial graph model is used to improve the accuracy of the detection, which considers the scale, orientation, position and neighborhood of features. To detect the presence of the object in a test image the same features are extracted and a graph matching algorithm is applied to match the created graph model to these features and derive a matching score. As a result, a matching score matrix is produced which represents the pair-wise comparison of training and test images.

As specially for landmarks and building, our method significantly outperforms the state of the art methods as shown in Section 3.5.2. A reason is that our approach considers buildings as 3D objects which is an advantage against existing planar geometry checking methods. Another reason is the use of spatial information, which is very important with artificial objects, like landmarks, because it contains several repeated similar features in the object.

Tag propagation

The geographical metadata (geotags) embedded in the image file usually consist of location names and/or GPS coordinates, but may also include altitude, viewpoint, etc. Two of the most commonly used metadata formats for image files are EXIF and IPTC. In this chapter, we consider the existing IPTC schema and introduce a hierarchical order for a subset of the available geotags, namely: city (name of the city where image was taken) and sublocation (area or name of the landmark).

Our system supports two application problems as shown in Figure 5.10. In the *closed set problem*, each test image is assumed to correspond to exactly one of the known (trained) landmarks. Therefore, the test image gets assigned to the most probable trained landmark and the corresponding tag is propagated to the image. This is comparable to an identification task in biometrics. However, in the *open set problem* the test image may correspond to an unknown landmark. This problem is comparable to a watchlist task in biometrics where the goal is to distinguish between known and unknown persons (landmarks). For example, in Figure 5.10 we assume that the system is trained with only three known landmarks: Budapest (Parliament), Belgrade (Church St. Sava)

and Tokyo (Tower). Given the input test image of Paris (Eiffel Tower), the system gives different results for the closed and open set problems. In the closed set problem, our system finds that Tokyo (Tower) is the most suitable model for the test image. If we consider the open set problem, the system does not retrieve any of the trained models since the matching scores between the object models and the test image do not exceed a predefined threshold. The open and closed set problems are separately evaluated in Section 5.2.6 as detection and recognition tasks, respectively.

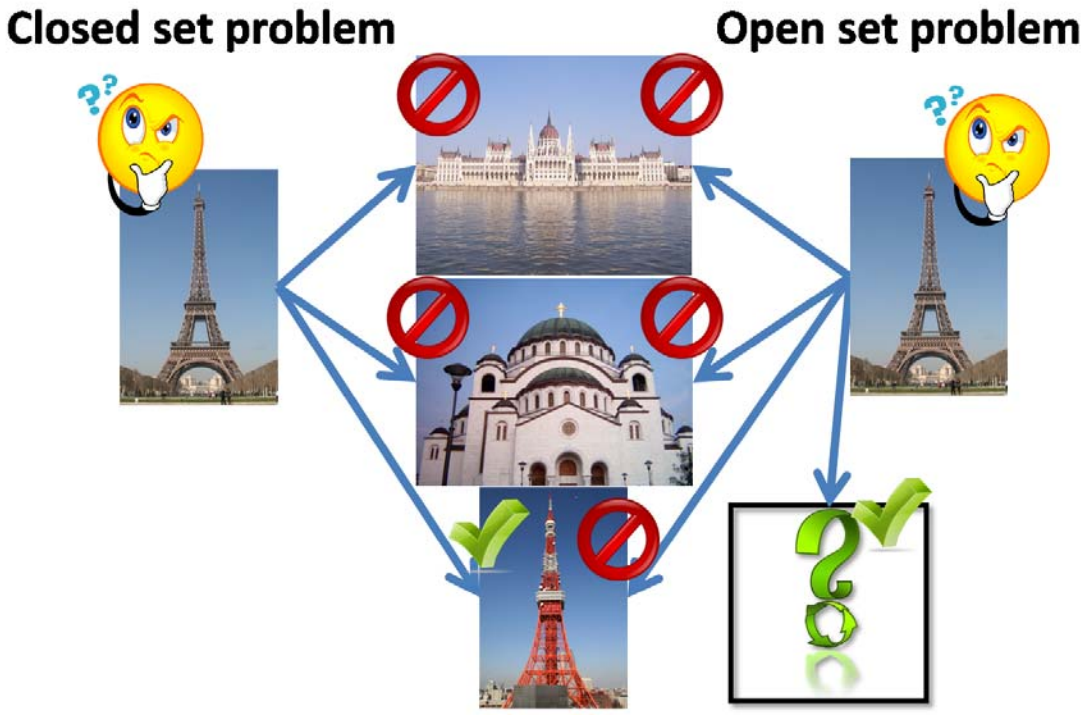


Figure 5.10 — The closed and the open set problems. In the closed set problem, each test picture is assumed to correspond to one of the known (trained) landmarks. However, in the open set problem the test picture may also correspond to an unknown landmark.

In a more detailed way, the object duplicate detection module provides a matching score matrix $S_{i,j}$. It represents the pair-wise comparison of the trained images (landmarks) i, $i \in [1, M]$, and the test images j, $j \in [1, N]$, where M and N are number of training and test images, respectively.

In the closed set problem, we find the maximum score for each test image j and propagate the geotag of the corresponding training image i to the test image. The assignment matrix $C_{i,j}$, is formed in the following way:

$$C_{i,j} = \begin{cases} 1, & \text{if } S_{i,j} = \max\limits_{i \in [1,M]} \{S_{i,j}\}; \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

In this case, each test image gets assigned with exactly one tag from the training photo dataset.

In the open set problem the tag propagation is only done if the corresponding score exceeds a

predefined threshold $\hat{S}$. The assignment matrix $O_{i,j}$, is defined as:

$$O_{i,j} = \begin{cases} 1, & \text{if } S_{i,j} = \max\limits_{i \in [1,M]} \{S_{i,j}\} \wedge S_{i,j} \geq \hat{S}; \\ 0, & \text{otherwise.} \end{cases} \tag{5.6}$$

In this case, each test image can get assigned zero or one tag from the training set depending on the value of the threshold $\hat{S}$.

Based on the assignment matrix $C_{i,j}$ or $O_{i,j}$ the tags are propagated. If the corresponding value is 1, the tag associated with training image i is propagated to the test image j. If the corresponding value is 0, no tag is propagated.

5.3.4 Experimental setup

We evaluate our automatic geotag propagation algorithm in the annotation process.

Database

We are interested in images that depict geographically unique landmarks. For instance, pictures taken by tourists are ideal because they often focus on the unique and interesting landmarks of a place. The dataset used in this experiment was obtained from Google Image Search, Flickr and Wikipedia by querying the associated tags for famous landmarks.

The resulting dataset consists of 1320 images: 22 cities (such as Amsterdam, Barcelona, London, Moscow, Paris), 3 landmarks for each of them (objects or areas in those cities, such as Bird's Nest Stadium, Sagrada Familia, Reichstag, Golden Gate Bridge, Eiffel Tower) and 20 image samples for each landmark. Figure 5.11 shows a single image for a single landmark from each of the 22 considered cities, while Figure 5.12 provides several images for 3 selected landmarks (e.g. Berlin - Reichstag, San Francisco - Golden Gate Bridge and Paris - Eiffel Tower). As can be seen from these samples, images with a large variety of view points and distances are considered for each landmark. Figure 5.13 provides all cities and landmarks in our dataset.

The dataset is split into a training and a test set. Training images are chosen carefully so that they provide a wide angle view of those landmarks without other dominating objects. Moreover, for each training image, negative and positive test pictures are selected. For each landmark there are 19 positive images in the test set. Negative images are all images which do not contain the ground truth landmark, namely all images which depict one of the other 65 landmarks. This leads to $19 \times 65 = 1235$ negative images in the test set.

In order to make our approach more computationally feasible, all images are downsized to a maximum size of 500×500 pixels and JPEG compressed before further processing.

Evaluation

For the evaluation, a ground truth matrix is defined as:

$$GT_{i,j} = \begin{cases} 1, & \text{if } Landmark(i) = Landmark(j); \\ 0, & \text{otherwise.} \end{cases} \tag{5.7}$$

Figure 5.11 — Sample landmarks for each of the 22 cities in the dataset. The dataset covers a large variety of landmarks including buildings, bridges, monuments, etc.

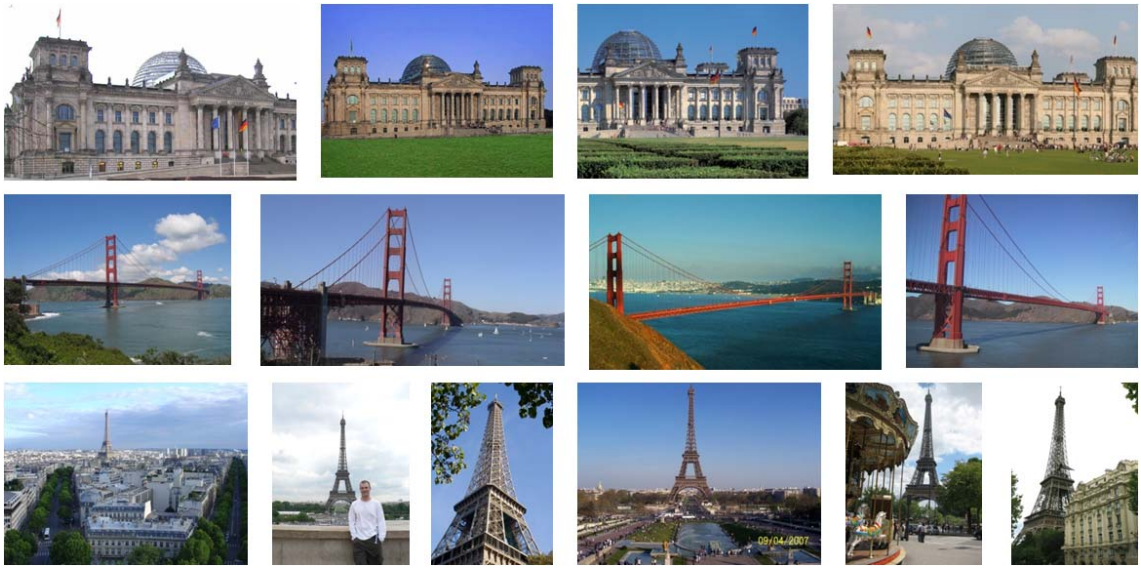


Figure 5.12 — Images for 3 selected landmarks: Berlin (Reichstag), San Francisco (Golden Gate Bridge) and Paris (Eiffel Tower). The images contain a large variety of views, distances, and partial occlusions.

where *Landmark* is geographically unique landmark name, $i \in [1, M]$, $j \in [1, N]$, M is the number of training images and N is the number of test images.

An *open set problem* can be evaluated as a typical detection task, where an image has to be classified as known or unknown. Given the ground truth and the predicted labels, the confusion matrix can be calculated as shown in Appendix A.

Given the assignment matrix $O_{i,j}$ for the open set problem, TP, FP and FN can be calculated as

$$TP = \sum_{i,j} GT_{i,j} \cdot O_{i,j}, \tag{5.8}$$

$$FP = \sum_{i,j} \Big(1 - GT_{i,j}\Big) \cdot O_{i,j}, \tag{5.9}$$

$$FN = \sum_{i,j} GT_{i,j} \cdot \Big(1 - O_{i,j}\Big). \tag{5.10}$$

A *closed set problem* can be evaluated using the recognition rate (RR). It is defined as the ratio between the numbers of correctly suggested tags T and overall number of samples A:

$$RR = \frac{T}{A}. \tag{5.11}$$

For tag propagation, T and A can be calculated as

$$T = \sum_{i,j} GT_{i,j} \cdot C_{i,j}, \tag{5.12}$$

$$A = \sum_{i,j} GT_{i,j}. \tag{5.13}$$

Results and analysis

First, the *closed set problem* is evaluated as a recognition task. The recognition rate for all landmarks is shown in Figure 5.13. The average recognition rate for each city is presented in the first column of this figure after sorting. In our dataset, there are three landmarks in each of the cities which are represented in the right columns of the figure.

The performance of the tag propagation varies considerably between different cities, but also across the individual landmarks in a city. According to common visual properties, all landmarks can be split into different groups, such as castles, churches, bridges, towers/statues, stadiums and ground structure, and further evaluation of the tag propagation scenario is based on these groups. Interestingly, by taking a closer look at the results in Figure 5.13, one can perceive that the performances of landmarks in the same group show small variations.

Figure 5.14 provides the recognition rates averaged over the different groups, which shows variation across the groups. Our approach performs the best for the group of castles, while stadiums show the worst results. The average RR across all landmarks is 71%. The recognition errors are solely caused by the object duplicate detection.

Second, the *open set problem* is evaluated as a detection task through the PR curves shown in

Figure 5.13 — The recognition rates for all landmarks. Each row represents one city from our dataset and the right three columns represent three landmarks in each of the cities. The first column shows the sorted average recognition rates for each city.

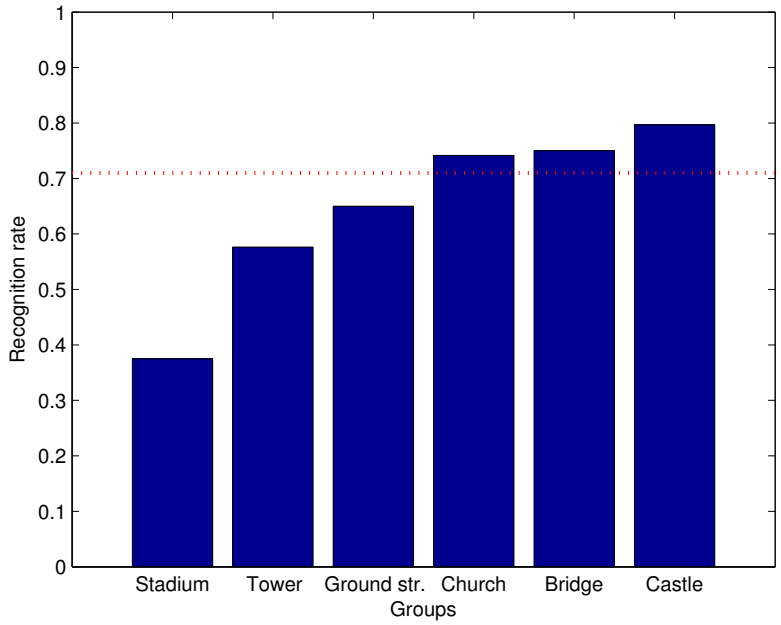


Figure 5.14 — The recognition rates across the different landmark groups in the closed set problem (bars) and the average recognition rate for all landmarks (dashed line). Landmarks have been grouped according to their visual characteristics.

Figure 5.15. The PR curves show significant difference between the different groups of landmarks. The proposed tag propagation method performs well with castles or other buildings which have more salient regions. In case of towers, it performs worse because the landmark does not have enough discriminative features. However, in case of stadiums, the performance is low due to the large variety of viewpoints.

The F-measures for the different detection thresholds $\hat{S}$ were calculated and shown in Figure 5.16 to determine the optimal threshold value. The optimal threshold is chosen for the maximum F-measure and shown by green markers. The optimal threshold value does not vary much depending on landmarks (standard deviation of 13%). The final F-measure for the open set problem averaged over the whole dataset is 73%.

5.3.5 Conclusion

In this work, we have developed an efficient system for automatic geotag propagation by associating locations with distinctive landmarks and using object duplicate detection for tag propagation. The adopted graph based approach reliably establishes the correspondence between a small set of tagged images and a large set of untagged images.

For assessing the quality of the tag propagation system we have considered the open and closed set problems. We have analyzed the performance of the tag propagation alone, which leads to a promising average accuracy of 71% over all the landmarks. Furthermore, we have shown that

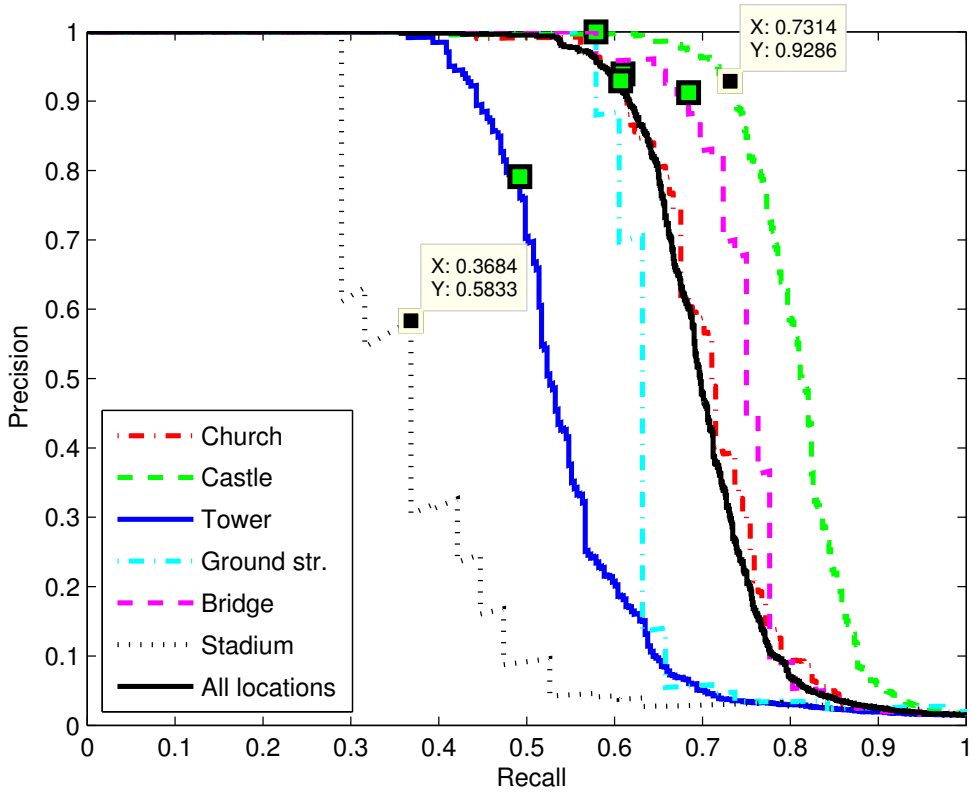


Figure 5.15 — Precision vs. recall curves for the open set problem across the different landmark groups. Markers indicate the cases when the maximal F-measure values are obtained.

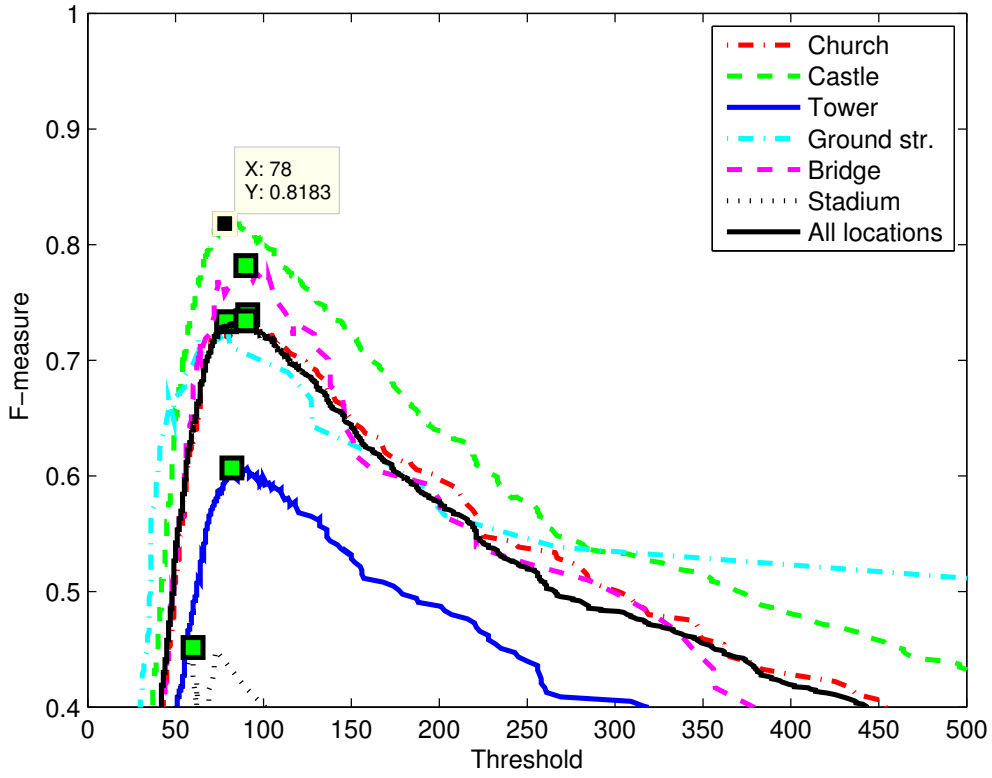


Figure 5.16 — F-measure versus detection threshold $\hat{S}$ across different landmark groups. Green markers show the optimal thresholds.

the performance varies considerably among different landmark types depending on their visual characteristics.

5.4 Cheese demonstration tool

As demonstration tool of this Chapter, an advanced image management platform for online use and mobile devices is developed, called Cheese*, as shown in Figure 5.20. Beside standard features such as image upload, tagging and keyword based search, it offers the user visual similarity based search, object based tagging and semi-automatic tag propagation. For improved interoperability between different image repositories and applications, the platform supports the export and import of image files with embedded metadata in JPSearch - Part 4 compliant format. Database contain more than a million of images.

5.4.1 User-friendly Interface

The user can annotate any photo in the database, which is either uploaded by himself/herself or by any other user. Images are annotated on the object level. Web and mobile platform is designed as shown in Figure 5.17. Also similar images can be search form through both platform.

Object can be marked and tagged in a query image by the user, he/she can ask the system to propagate it automatically to the other images in the database by pressing the "propagate" button. The system performs object duplicate detection and returns images containing object duplicates from the database to get confirmation from the user, as shown in Figure 5.18.

5.4.2 JPSearch - Part 4 compliant

Our framework is the first in the world, which is JPSearch - Part 4 compliant. File format for metadata embedded in image data (JPEG and JPEG 2000) adopts the well known image data formats for embedding metadata information. The benefit of such an integration and combination of metadata and raw data is the mobility of metadata and its persistent association with the image itself.

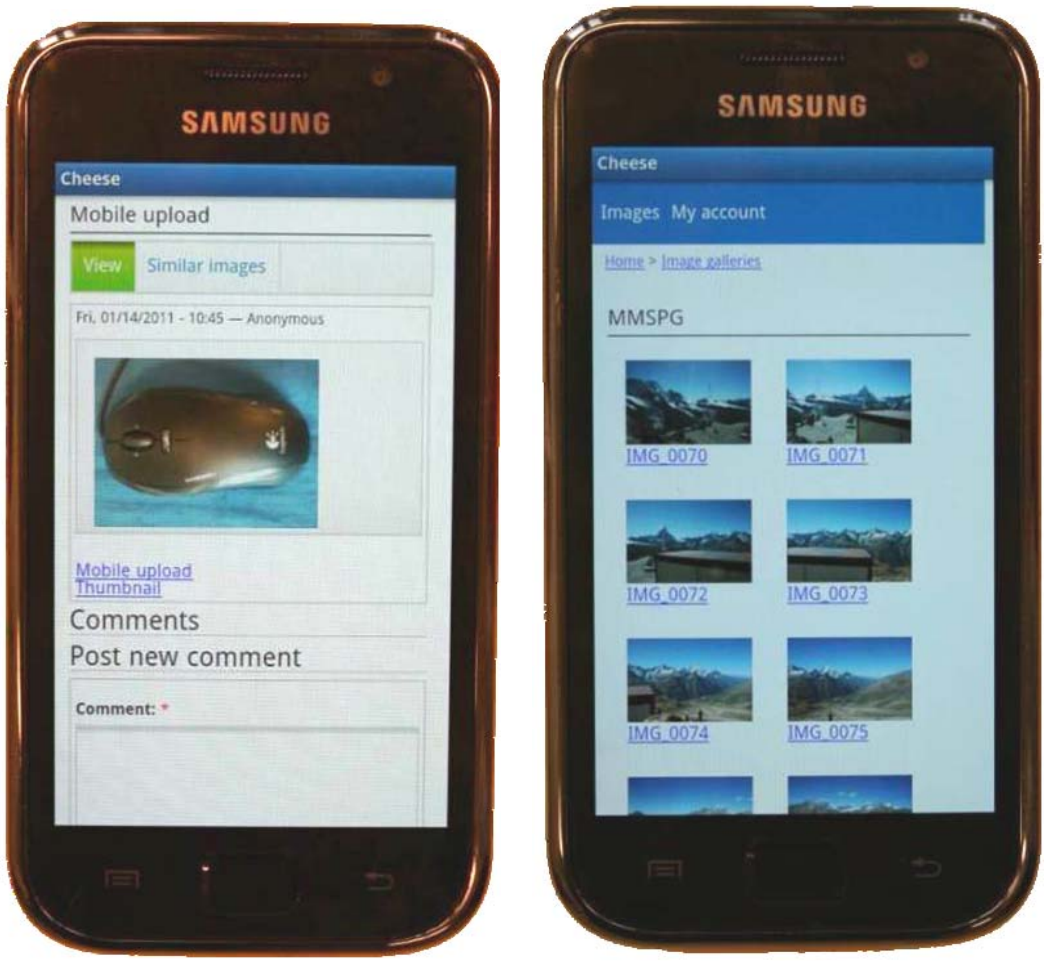
An example for an embedded metadata can be seen in algorithm 5.4.2.

5.4.3 Database

The database contains evaluation, distraction images, personal images from all over the world. Part of the dataset is obtained from Flickr and from conferences. The rest of the dataset is formed by manually taken photos of particular objects for evaluation and test purpose. The database continuously growing, thanks to the active users.

By uploading a new image or any other content, the user allows others to distribute, remix, reuse, and build upon their work, even commercially, as long as they credit the user for the original creation (According to the Creative Commons license: Attribution CC BY).

*http://cheese.epfl.ch/



(a) (b)

Figure 5.17 — Mobile interface and image similarity search in Cheese platform.

Algorithm 4 Embedded object annotation in JPEG file

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="JPSearch:schema:coremetadata jpcore.xsd">
  ...
<RegionOfInterest>
  <RegionLocator>
    <Region dim="2"> 50 30 100 100 </Region>
  </RegionLocator>
  <Description> Zermatt, Switzerland </Description>
  <Keyword> MMSPG, EPFL </Keyword>
</RegionOfInterest>
  ...
```

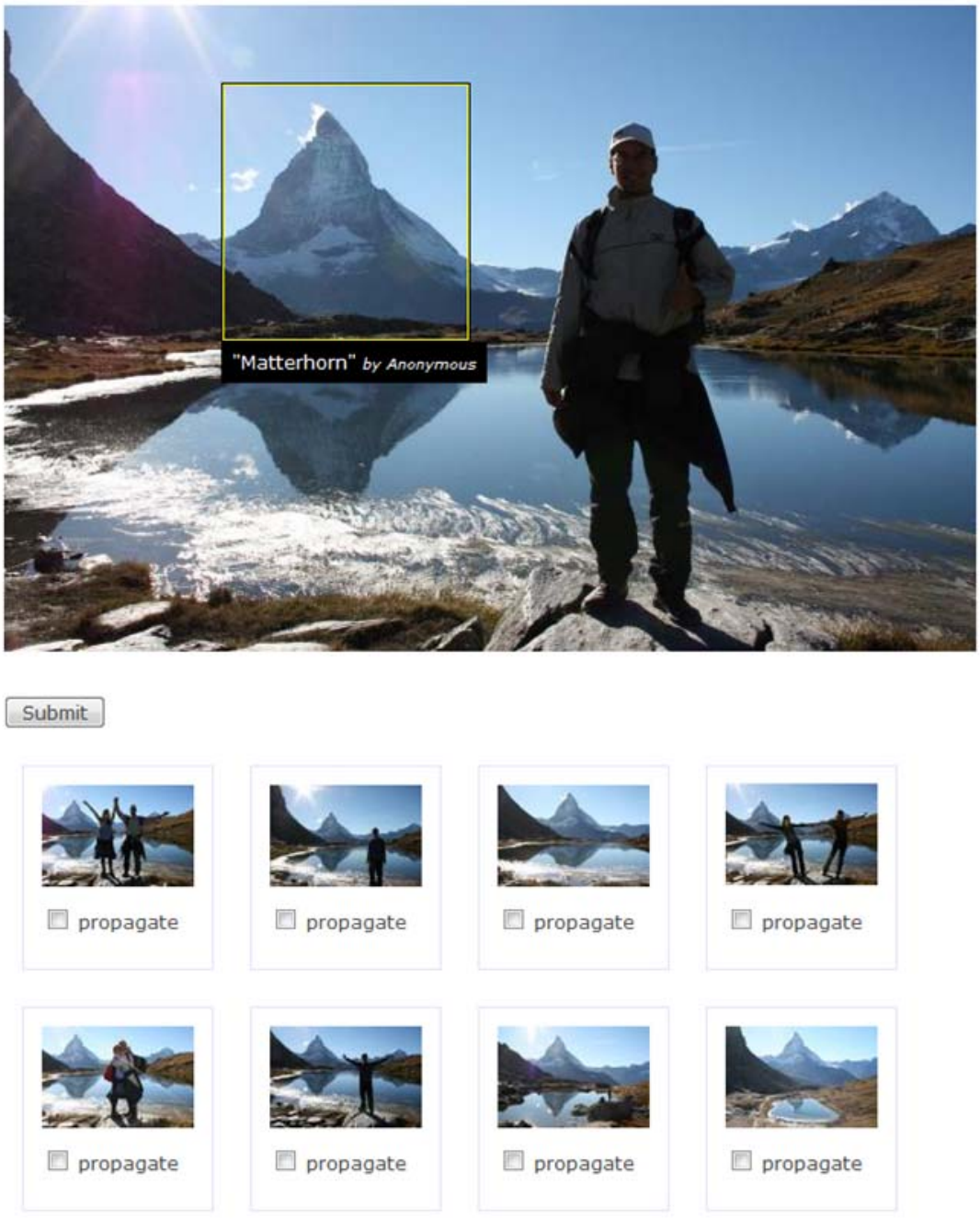


Figure 5.18 — Tagpropagation on Cheese paltform.

Figure 5.19 — Cheese, advanced image management platform.

The resulting dataset consists more than 1million images: 8 galleries, 125 albums. Summary of the galleries are shown in Table 5.2. Figure 5.20 shows some example images from the Flickr gallery. As it can be seen, images with a large variety of contents with different view points and distances, are uploaded to Cheese.

5.4.4 Statistics

Cheese was opened for public in July 2011. Till September 2011, 348 unique users has been visited Cheese from all over the world as sown in Figure 5.21. 5806 pages were visited during this period. Users spent 4m23s time on the site in average using different browsers and operating systems as it can be seen in Figure 5.22 and Figure 5.23. Statistic was measured by Google analitics *.

5.5 Chapter summary

Social networks are gaining popularity for sharing interests and information. Especially photo sharing and tagging is becoming increasingly popular. Among others, tags of people, locations, and objects provide efficient information for grouping or retrieving images. Since the manual annotation of these tags is quite time consuming automatic tag propagation based on visual similarity offers a very interesting solution.

In this chapter we have developed two applications for large scale object duplicate detection.

*http://www.google.com/analytics/

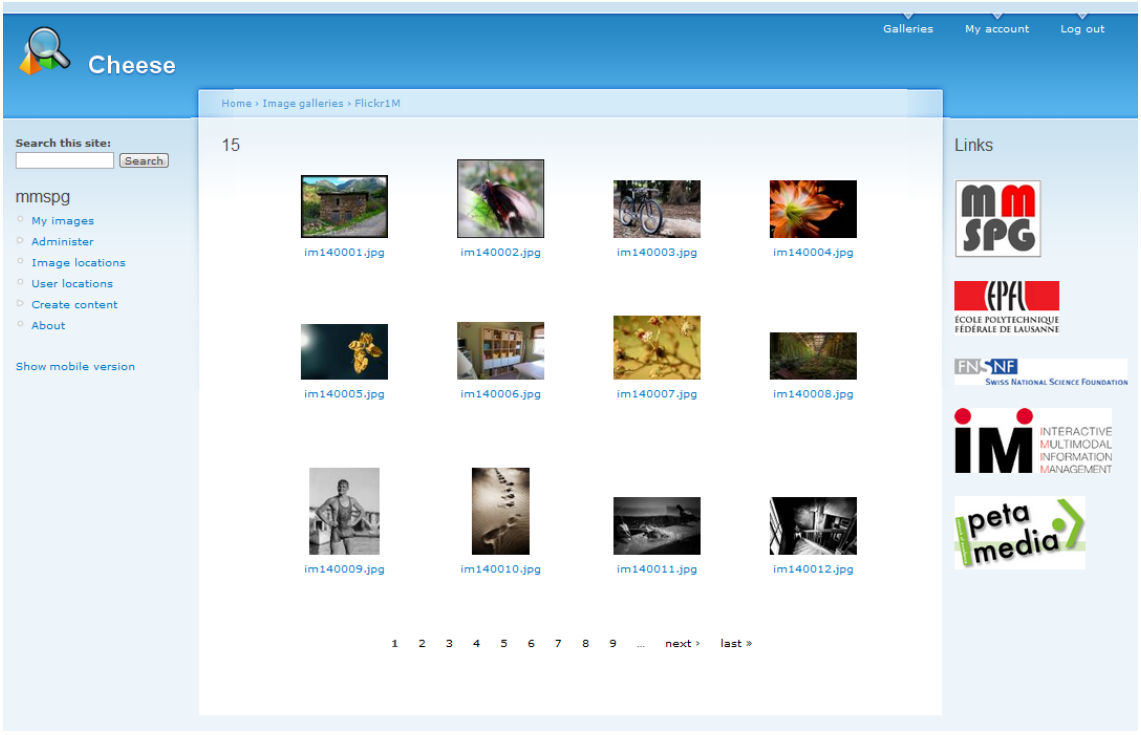


Figure 5.20 — Database samples from Cheese, advanced image management platform.

Table 5.2 — Summary of the classes and some example objects

Gallery	Albums	#images
S3MR 2011	This album uploaded by participants of the 2nd Summer School on Social Media Retrieval (S3MR)	1787
Test	Debugging gallery for testing new features	4
MMSPG	MultiMedia Signal Processing Groups's gallery for demonstration purpose	352
Mobile upload	Gallery for Cheese mobile application	415
ACM Multimedia 2010	Gallery for ACM Multimedia conference 2010	4
Flickr1M	Flickr 1million database collection	1013076
Trademarks	Trademarks gallery for evaluation purpose	80
World	Albums from different places over the world	300
8	125	1016018

Figure 5.21 — User visits statistic of Cheese all over the world.

	Operating System	None	Visits	Visits
1.	Windows		366	65.47%
2.	Macintosh		111	19.86%
3.	Linux		67	11.99%
4.	Android		10	1.79%
5.	iPhone		3	0.54%
6.	iPad		2	0.36%

Operating System contribution to total: Visits

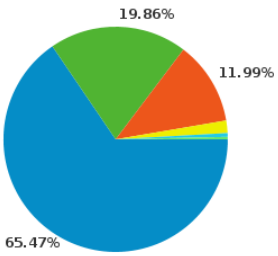


Figure 5.22 — Operating System statistics of Cheese platform.

	Browser	None	Visits	Visits
1.	Firefox		224	40.07%
2.	Chrome		184	32.92%
3.	Internet Explorer		78	13.95%
4.	Safari		56	10.02%
5.	Android Browser		9	1.61%
6.	Opera		5	0.89%
7.	Mozilla Compatible Agent		1	0.18%
8.	Opera Mini		1	0.18%
9.	SeaMonkey		1	0.18%

Browser contribution to total: Visits

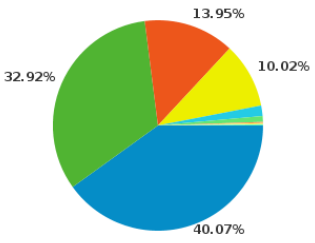


Figure 5.23 — Web browser statistics of Cheese platform.

- Two level object detection was proposed for efficient search in a large scale database. First, a fast image selection algorithm eliminates images which do not contain the query object, then an accurate object duplicate detection algorithm provide bounding boxes. The annotation can be performed through a tag recommendation process, in which the system recommends tags associated with the object in the images of the search results, or through tag propagation process, when the user enters his/her tag for the object and it is propagated to the images in the search results. It has been also shown that the detection works reliably for salient objects such as trademarks, books, newspapers, and gadgets.

- For landmarks and building, our method significantly outperforms the state of the art methods as it can be seen in Figure 3.14 (F-measure: Our method: 77%, Lowe's: 72%, Ransac: 65% and BoW: 51%). The reason is that our approach considers buildings as 3D objects which is huge advantage against the planar geometry checking methods. Another reason is the use of spatial information, which is very important with artificial object, like landmarks, because it contains several repeated similar features.

Future work can focus on new features which considers spatial information, but do not decrease the speed of search or increasing the size of the database. In our case one computer is capable to search among 10million images and objects. On of the solution can be automatic segmentation of objects in the images during the preprocessing phase.

Talk does not cook the rice.

Chinese proverb

6 Specific applications for mobile phone

In this Chapter, we show several mobile applications for object duplicate detection. First we discuss different frameworks for mobile visual search. Then we improve the accuracy of the object recognition by developing specific feature extraction or even specific detection algorithms which consider prior information about the target object. For example, general object duplicate detection may fail to detection chess figures, however considering context, like chess board position and hight of the chess figure, detection can be more accurate. Another example is a coin and banknote recognition algorithm on mobile considering specific features for coin, such as relative size between coins. We show that user interaction further improves image retrieval compared to pure content-based methods through a game, called Epitome. Therefore in this chapter we open several research fields for object duplicate detection by considering contextual information.

6.1 Introduction

Thanks to advances in digital acquisition, processing, and storage technologies, millions of images are captured every day and shared in online social services such as Facebook*, Flickr†, and Picasa ‡. Furthermore, images provide an interesting way to identify or to find desired objects and locations. Image based search and retrieval is becoming increasingly popular to annotate images in large databases and for their retrieval.

With around 60% worldwide penetration, mobile phones are by far the most popular electronic devices ever used. In addition to basic functionalities, modern mobile phones provide other features

*http://www.facebook.com/
†http://www.flickr.com/
‡http://picasa.google.com/

such as internet connection and embedded cameras. These features provide an intuitive human computer interface for web search on the go. Instead of a traditional text-based query which is quite inconvenient given the constraints of mobile phones, the query is simply formulated by capturing a photo of the object of interest. The search application will then use that photo to find similar instances of that object in a database, and provides users with associated information or services.

Mobile image search and retrieval has become increasingly popular and several commercial applications and services have been developed, including Kooaba, Google Goggles and Snaptell. *Kooaba*[*] is based on SURF features and it detects specific objects, such as posters, CDs, DVDs, books, and game covers. *Snaptell*[†] detects objects through local features and Accumulated Signed Gradient matching. Therefore Snaptell is robust to changing viewpoints and partial occlusions. *Goggles*[‡] is the most recent commercial application from Google. It can detect logos, book covers, artworks, places and wines using visual and GPS information. In case of wine recognition the algorithm recognize the label of the bottle. All of these commercial applications use their own database and do not allow users to create and annotate objects.

In this chapter we present new and innovative solutions for object recognition based on higher level features by considering i) knowledge on the environment by taking into account the time, location and sight direction, ii) knowledge upon the user and provided by the user, and iii) information on content (cf. Figure 6.1). We have identified a number of promising and interesting research problems which improve the performance, the understanding, and the usability of object recognition and identification systems.

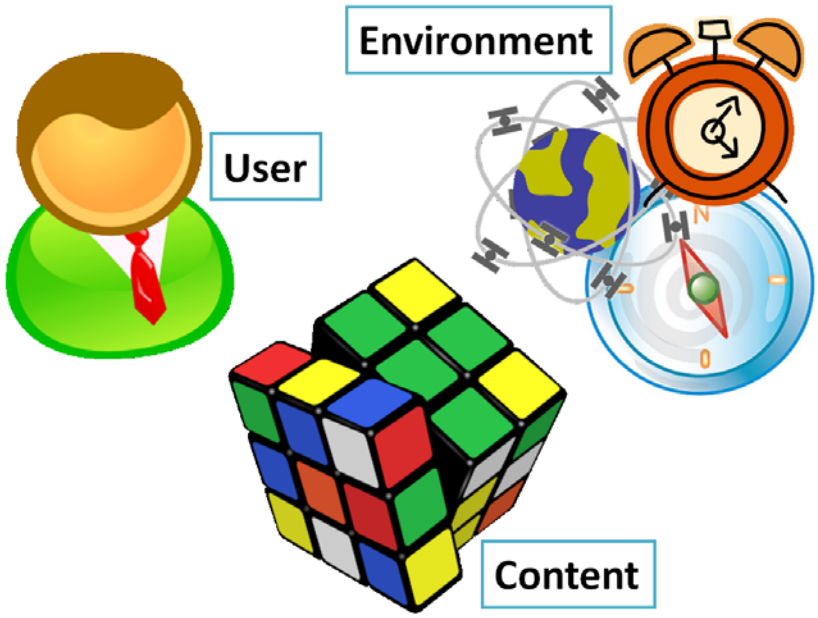


Figure 6.1 — Context of image contains environment, user and the actual content.

[*] http://www.kooaba.com/
[†] http://www.snaptell.com/
[‡] http://www.google.com/mobile/goggles/

General object recognition algorithms are efficient, but exhibit some limitations and cannot be as accurate when compared to specific object recognition systems. Therefore our goal was to extend our object replica detection paradigm toward more accurate specific object recognition and identification algorithms. Considering the higher level features and knowledge, we bring new and innovative solutions for efficient and accurate object replica detection. Our final objective was to build demonstrators, based on our system, and to evaluate their performance in real life situations.

In the following Section, we discuss general frameworks for mobile visual search, considering system architectures from mobile based system to server side application with thin interface on the mobile phone. Next content based application is presented for museum guide in Olympic Museum, Lausanne in Section 6.2. Considering contextual information, coin and banknote recognition follows, calculating the exchange value for different currency, by recognizing them with different methods as shown in Section 6.3. Finally, to demonstrate the importance of user-interaction, a game is proposed for photo album summarization and user results with automatic image processing algorithm is compared in Section 6.4. It demonstrates a simple game for photo album summarization where humans outperform the existing automatic computer vision algorithms. Therefore it shows that photo album summarization is complex and difficult processing task in the human brains, and that there are still big gap between computer and humans in image processing and retrieval. This game shows a way for solving a complex problem by separating or reshaping the problem for people without deep knowledge on the complex problem.

Further applications from content, context and user-interaction based object duplicate detection can be find in the Appendix B as content based: Visual Bookmark in Section B.1, context based: chess recognition in Section B.2 and user-interaction based object duplicate detection: flower recognition in Section B.3.

6.1.1 Frameworks

The comparison of the different architectures for mobile image search is conducted in the following scenario. When a user is interested in an object and wants more information about it, he/she takes a photo with the camera of his/her mobile phone and queries the application using this image. The application matches the object with a (distributed) database from different sources, such as Wikipedia*, Amazon† and eBay‡. If a match is found, the application returns associated information such as a detailed description, product price or other pointers.

Given this mobile image search scenario the following issues have to be considered:

- In comparison to stationary devices such as servers, mobile devices are limited in terms of computational power, available memory and autonomy. Therefore the amount of data which is stored and the complexity of the used algorithms have to be much lower.
- The mobility offered by these devices is mainly due to wireless transmission channels such as 2G, 3G and WiFi. While wireless transmission channels are widely available, the provided bandwidth in some locations could be limited, potentially leading to unacceptable latencies.

*http://www.wikipedia.org/
†http://www.amazon.com/
‡http://www.ebay.com/

- While the performance of 2D object recognition has improved considerably over the last decade, 3D object recognition is still not robust enough and requires improved algorithms for feature extraction and matching.

The limited performance and diversity of mobile devices makes a large-scale deployment of complex mobile applications rather difficult. Therefore, alternative approaches such as cloud computing [Weiss, 2007] can be used to resolve this problem. The basic idea it to provide "convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [NIST, 2010]. This centralized architecture of data, applications and computational power may be accessed and used by any mobile device with internet connection.

In many cloud applications, mobile devices are considered as thin clients in charge of data capture, rendering, and communication. While this centralized architecture maybe suitable for text, speech, and even audio-based applications, it is not efficient for image or video due to the large amount of data which has to be transmitted over a wireless channel with limited capacity. In order to reduce the information which has to be transmitted, recent work on mobile image search and retrieval (Kooaba*, Google Goggles†, Snaptell ‡) transfers search related computing such as feature extraction and matching from the server to the client. In a decentralized architecture the complete image search may be implemented on the mobile device.

Given an input image of the object of interest, retrieval algorithms usually consists of the following steps [Hong, 2009]: keypoint detection, feature extraction, feature compression, feature matching and topology verification. Considering a client-server architecture, the previous steps maybe distributed in different ways between the clients and servers. However in mobile object duplicate detection, mobile device is expected to include an image or video acquisition and preprocessing functionality. We compare the following alternative configurations for mobile image retrieval, as depicted in Fig. 6.2, covering a pure server side all the way to a pure client side configuration:

1. *Cloud-based search:* Full server side configuration, where just a thin client is used on the mobile phone. The captured image is directly transmitted to the server, where all the processing steps are performed. As a result the retrieved information is sent back to the client. Cloud application allows scalable selection of content, features and resources. This framework is suited for general application in company, where the application should be flexible for modification in different mobile platform.

2. *Server-based search:* In order to reduce the required bandwidth for the transmission, the feature extraction and compression are performed on the mobile phone [Chandrasekhar *et al.*, 2009; Girod, 2009]. The features are sent to the server, where the remaining steps are completed. The resulting information is sent back to the mobile. Most popular search

*http://www.kooaba.com/
†http://www.google.com/mobile/goggles/
‡http://www.snaptell.com/

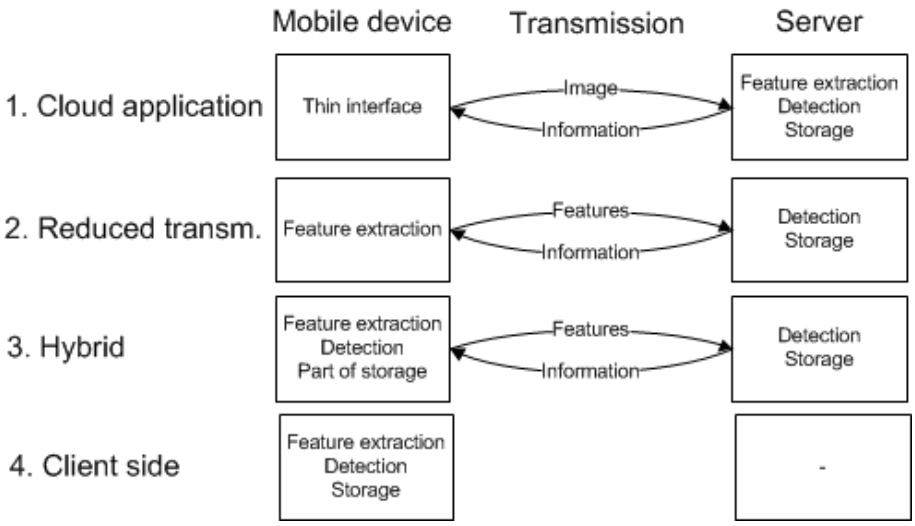


Figure 6.2 — Illustration of possible configurations for mobile image search with varying distribution of processing steps between mobile device and server.

engines are using this framework, such as Google Goggles. Usually the interface is presented as a web page for the user.

3. *Hybrid search:* In a hybrid configuration retrieval is divided in two steps [Hong, 2009]. On the mobile phone a first retrieval is performed on a reduced dataset with recent or frequent content stored locally. An extended retrieval is then performed on the server-side using the query image as well as eventual results of local query. Foursquare and GPS based applications are usually download small databases considering the actual user location for evaluation. This framework is suited for augmented reality for games.

4. *Client-based search:* The retrieval is performed on the mobile [Girod, 2009], using only locally stored datasets and less complex algorithms. This configuration is especially appealing for clients with very large storage capacity. Due to the limited memory and computation power of the device only reduced datasets and less complex algorithms may be used.

6.2 Museum guide

6.2.1 Introduction

Many museums still present their exhibits in a rather passive and non-engaging way. Even today in some museums the visitor has to search through a booklet in order to find descriptions about the objects on display. However, looking for information in this way is a quite tedious procedure. Moreover, the information found does not always meet the visitor's specific interests. One possibility of making exhibitions more attractive to the visitor is to improve the interaction between the visitor and the objects of interest by means of a mobile guide.

Using mobile phones as a platform for personal museum guide has several advantages over the traditional audio guide systems. The interaction of taking a snapshot is found more intuitive than finding an object's number and typing it into the device. In addition, from an economical point of view, either museum operators profit by significantly reducing maintenance and specific infrastructure costs or tourist operators can develop their own products, since the visitor can use his own mobile device.

In this Section Olympic Museum guide application is described.

6.2.2 Related Work

Mobile image search and retrieval has become increasingly popular among researchers in recent years. Some prototype applications have been developed in order to study different aspects of mobile image search. Besides these scientific prototypes, commercial applications and services have been developed, such as Kooaba*, Google Goggles† and Snaptell‡.

Several approaches have been proposed that allow visitors to interact via an automatic museum guide.

Kusunoki et al. [Kusunoki *et al.*, 2002] proposed a system for children that uses a sensing board, which can rapidly recognize type and location of multiple objects. It creates an immersing environment by giving audiovisual feedback to children. Other approaches include robots that guide users through museums [Burgard *et al.*, 1998; Thrun *et al.*, 2000]. However, such robots are difficult to adapt to different environments, and they are not appropriate for individual use. An Interactive Museum Guide [Bay *et al.*, 2006a] that is capable of recognizing objects in the Swiss National Museum in Zurich was proposed. In order to reduce the search space, Bluetooth emitters were installed on site. Objects are recognized with an approximated SIFT algorithm. Search space can be similarly reduced by using WiFi stations, or RFID tags. In our case, does not need to reduce the search space inside of the museum, due to the precise object duplicate detection and WiFi or Bluetooth can not give precise location, considering different obstacles and other moving persons. However, using RFID, can be a precise solution to replace the content based search, but it is cost more than the free content based search and it takes more time to set up these chips onto the objects. PhoneGuide [Föckler *et al.*, 2005] uses a simple and light-weight object recognition approach that is realized with single-layer perceptron neuronal networks. The whole computation for object recognition is carried out on the device. Boris Ruf et al. presented a PDA based museum guide system that enables to recognize paintings in art galleries [Ruf *et al.*, 2008]. The algorithms SIFT and SURF were integrated in a fully implemented prototype system and their performance was thoroughly evaluated under realistic conditions. The training data was extracted from the online archive Web Gallery of Art. In order to speed up the matching process for finding the corresponding sample in the feature database, an approximation to nearest neighbor search was investigated. The k-means based clustering approach was found to significantly improve the computational time.

*http://www.kooaba.com/
†http://www.google.com/mobile/goggles/
‡http://www.snaptell.com/

One of the challenges in mobile visual search is reliable and efficient identification of 3D as opposed to planar objects. Most state of the art techniques in visual search assume that a 2D projection of the 3D query object does not much affect its performance. Reliable 3D object duplicate detection has been developed for mobile phones as described in B.1.

6.2.3 Proposed application

The Olympic museum guide application an interactive guide that is able to automatically retrieve information about objects on display in the museum by applying image recognition.

The scenario of use of this application by a visitor of the museum is as follows: The visitor uses his/her mobile phone to capture an image of an exhibit in the museum. Then the mobile application sends the image to a server that applies visual object recognition by searching visual content in an image database. Each image in the database is associated with a webpage containing information such as audio, video, slideshow and text. When the matched object is found the server returns the corresponding url and the information related to the specific object is automatically shown to the user.

The architecture of the system follows the classical server-client approach. The client acquires and sends the data to the server that replies with the results. No additional computation such as feature extraction is executed on the client. This decision has been taken for several reasons. The CPU of the mobile client is much slower than that of the server and running the feature extraction on the mobile client could result in very long waiting times for the user. Also, as we will show later on, the recognition performance is good even at low resolution. Transmitting scaled down images of small data size is sufficient for successful object recognition.

A high-level description of the architecture of the system is shown in Figure 6.3.

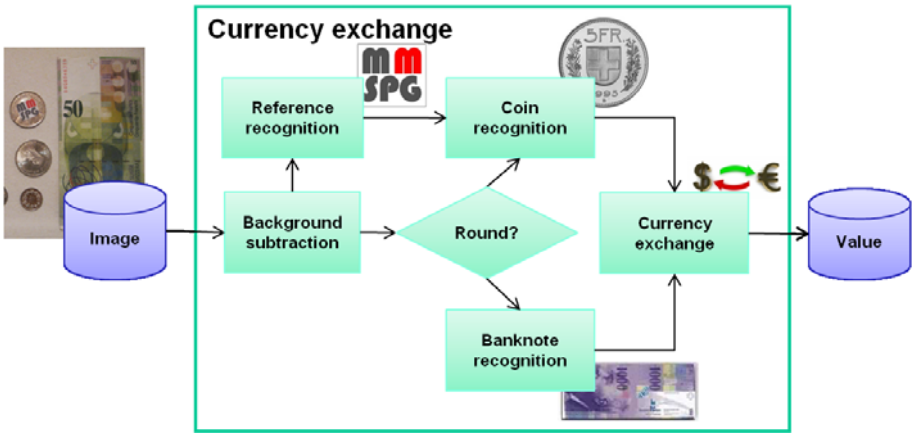


Figure 6.3 — The architecture of the Olympic Museum guide application.

Webpages are created on the server for 30 objects associating each item with the available audio, video, slideshow and text information, which is shown to the user once the object is recognized.

The duplicate object detection algorithm that was used is described in Section 3. However it is based on the SURF (Speeded Up Robust Features) algorithm for the feature extraction. This

algorithm provides robustness to scaling and rotation

The mobile client application was implemented in Android 2.3.3.

Figure 6.4 shows a flowchart of the system. When the application starts, it checks whether the internet connection is enabled. If not, it shows an alert message to the user and exits. Otherwise, it displays the welcome screen.

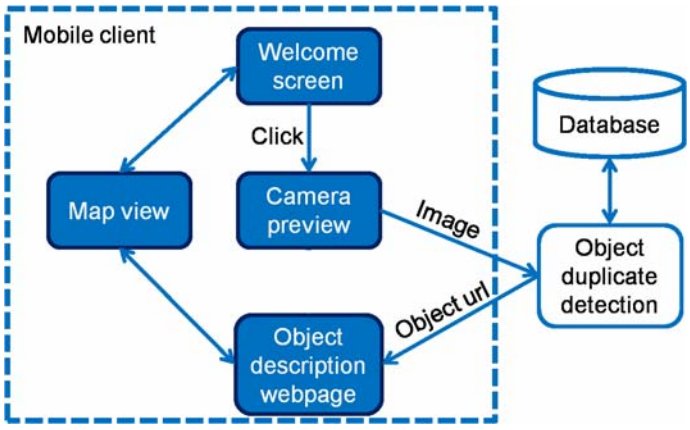


Figure 6.4 — Flowchart of the system.

In this screen a menu is provided, with the options to view the map of the museum or exit. After the user touches the screen the camera window opens and the user can take an image of an exhibit. After the image is captured, the application resizes the image width to 400 pixels while keeping the aspect ratio of the image. Then, the resized image is sent to the server. The application communicates with the server over the HTTP protocol. The server runs the object recognition algorithm and when the matching object is found it sends back to the application the url link associated with the matching object. Finally, the application loads the corresponding link that presents the webpage to the user. Attached to this view is a menu that gives the user the options to take another image, view the map of the museum or exit the application.

Figure 6.5 shows an example screenshot of the interface showing the webpage associated with a particular exhibit in the museum. It contains audio that starts playing automatically when the webpage is loaded, a slideshow of images of the athlete as well as text containing a short biography of his/her.

6.2.4 Evaluation

The application was tested on a real case scenario on selected artifacts in the museum. The system was evaluated by means of both objective and subjective evaluation. Database is collected with a help from Olympic Museum of Lausanne.

Database

Currently, the Olympic Museum of Lausanne uses a traditional audio guide (iPod device), which provides poor features, only audio information for the exhibits of the museum. They provided us

Figure 6.5 — Example screenshot showing the webpage provided for a sample exhibit of the museum.

with material for certain exhibits of the museum, including the audio information of their audio guide, additional images of the exposed objects as well as for athletes associated with them, videos, and text, for example biographies of athletes.

In order to create the training images database we took images for each of the 30 selected objects in the museum. The database of training images contains 270 images. For each object we took around 10 images in total, 5 images from 5 different angles and 5 more for the same angles but from a different distance from the object (Figure 6.6). The images were captured with a Samsung Nexus S mobile phone. The resolution of the captured images is 2560×1920 pixels. Figure 6.7 shows the set of training images with one image per object.

Figure 6.6 — Sample training images for a particular object of the museum.

6.2.5 Results and analysis

Objective evaluation

For the objective evaluation of the application we performed a set of measurements so as to compute the recognition rate and the response time by setting as parameters the resolution and the number

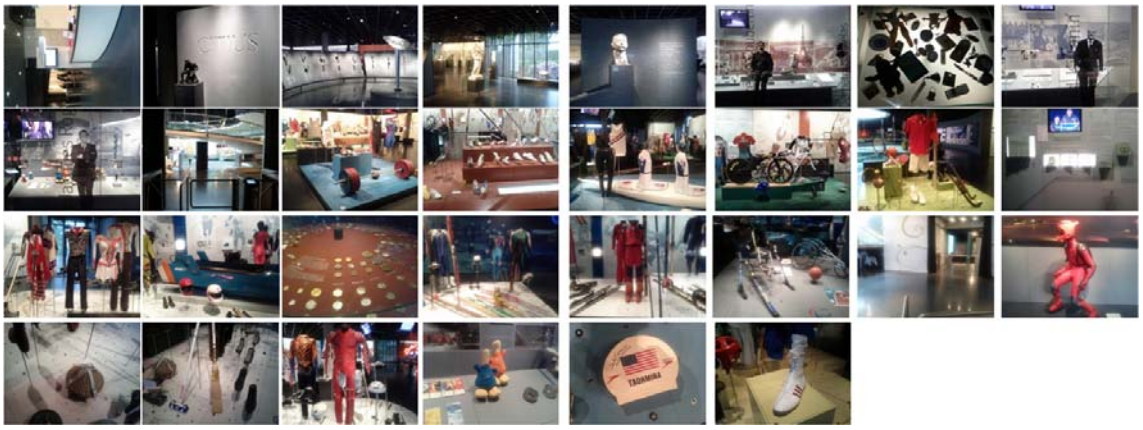
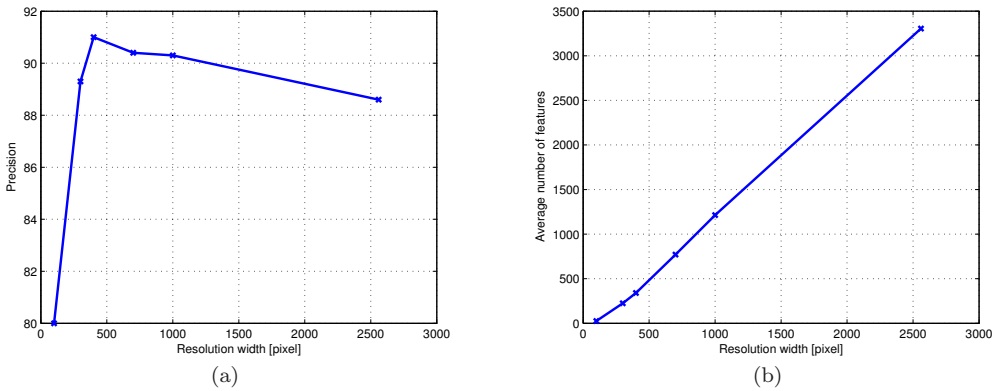


Figure 6.7 — Set of training images with one image per object.

of the training images. All the measurements were done with the training images of the collected database.

First, we measured the precision of the duplicate object detection for 7 different resolutions as shown in Figure 6.8 (a). For that we resized the original images to different resolutions. For each resolution, each image of the training images was tested against all the others using the object recognition algorithm proposed in Section 3 and we measured the percentage of images for which the correct matching object was found. Moreover, for the same set of resolutions we measured the average number of features per image, as shown in Figure 6.8 (b).



(a) (b)

Figure 6.8 — Precision of the object recognition (a) and average number of features per image for different resolutions (b) is shown.

We observe that reducing the resolution of the images from the original (2560 pixels) by resizing the image while keeping the aspect ratio, increases the precision, up to a point where it starts decreasing. As the resolution decreases from 2560 to 400 pixels width, the number of features decreases and the more important features are kept, which leads to higher precision. However,

when resolution drops under 400 pixels the number of features becomes insufficient for successful recognition and as a result the precision decreases. So, considering the precision aspect, among the resolutions that we considered, the 400 pixels is the most appropriate (91% precision). In the further evaluation, resolution: 400 pixels were used.

In addition, we investigated how the number of training images affects the recognition rate. For 24 of the 30 objects of our database we have 10 images per object captured by the ten different positions shown in the below Figure 6.9 below.

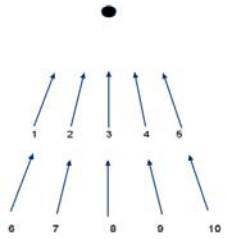


Figure 6.9 — The different positions from which we captured the training images.

For these 24 objects, for a given resolution, we measured the precision using as input four different sets of training images, as shown in Figure 6.10. The first set contained 2 images per object, those captured from positions 3 and 8 (directly in front of the object, considering 48 training images). The second set contained 6 images per object, those captured from positions 1, 3, 5, 6, 8 and 10 (144 images). The last set contained all the ten images of the object (240 images). The measurements were performed for a resolution of 400×300 pixels.

First of all we observe that using ten images per object for all the objects of the database provides precision 98.75%. In Figure 6.8 for resolution 400×300 pixels we have 91% precision. In those measurements we consider not only the 24 objects for which we have 10 images per object, but also the remaining 6 objects for which we have from 5 to 9 images per object. With 6 images per object the precision decreases to 94%, whereas with 2 images per object the precision drops to 77%, which is yet good precision considering that the recognition is done with only two front images of the object.

Finally, we did measurements in order to explore how the response times of the application are affected by different resolutions of images, as shown in Figure 6.11. Samsung S mobile phone was used with Android 2.3.3 operating system during the experiment. WiFi network was used. In case of real application, the museum can provide WiFi network along with the content. On the client side, we measured the delay from the moment that the application sends the image to the server until our application gets back a response with the corresponding url. We also measured the total delay from the time that the application sends the image to the server until the corresponding webpage is fully loaded. On the server side, we calculated the delay of the object duplicate detection algorithm.

From these measurements we computed the time required for the uploading of the image on

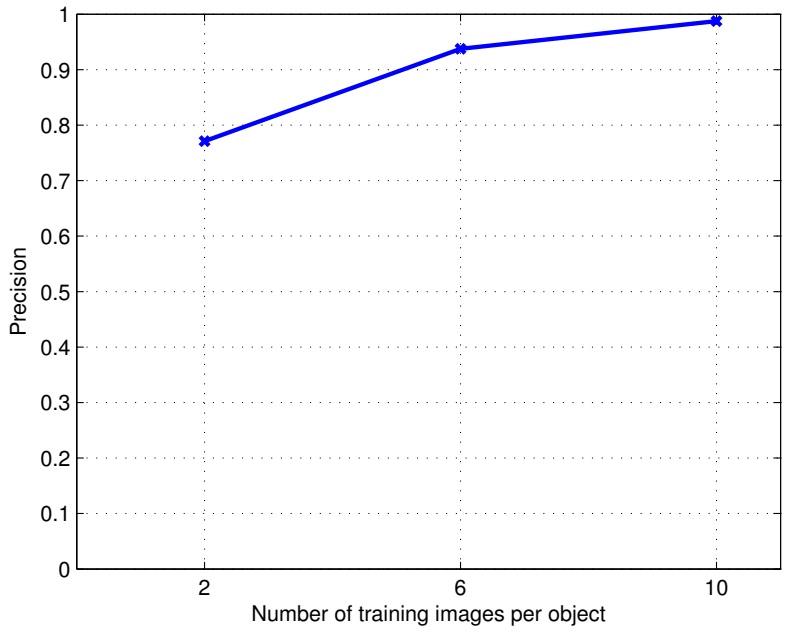


Figure 6.10 — Precision of the object detection for different number of training images per object.

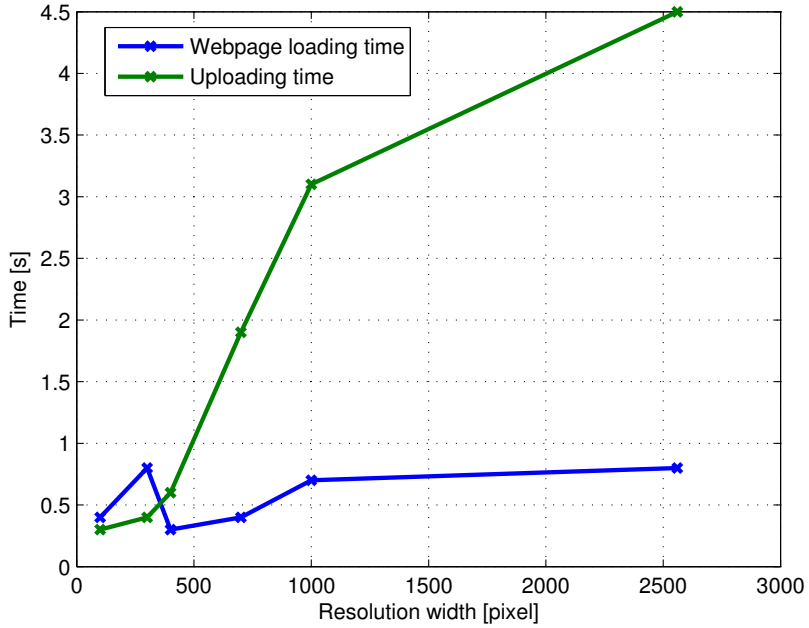


Figure 6.11 — Webpage and uploading times for different resolutions.

the server as a difference of the delay from the moment that the application sends the image to the server until our application gets back a response with the corresponding url minus the delay of the object recognition. We also computed the time required to fully load an object's webpage after the response with the corresponding url is provided from the server, as a difference of the time required to fully load the object's webpage after the image is sent to the server minus the time required in order to get the response from the server.

We observe that the webpage loading time is as expected nearly constant (around 0.6 seconds), thus independent of the size of the images. The uploading time remains practically invariant below the resolution of 400 pixels. However, after 400 pixels, there is a sharp increase until the resolution of 1000 pixels, after which it increases with a lower rate.

From the diagrams we have presented in the objective evaluation we can conclude that, among the considered resolutions, the 400×300 pixels resolution is the most appropriate to use, since it provides satisfactory results with respect to precision and response time.

Subjective evaluation

The application was evaluated in a real case scenario from visitors in the museum in order to investigate user experience.

Ten visitors participated in the evaluation. Their age varied between 20 and 40 years. Three of them had no previous Smartphone experience.

First, each user was given a description of the goal of the application and how it works. Then they were given a mobile phone with the application installed and they used the application at three selected exhibits of the museum. We were investigated in measuring subjectively the interface and not the object detection algorithm, therefore three objects were enough to get opinion on the mobile interface. Help was provided to them when they had questions. After that, each user filled a questionnaire that we designed so as to evaluate user experience.

Multidimensional scaling was used to determine dimensions used in the evaluation. The questionnaire contained 9 scaling questions (scale from 1 to 5) as well as two textboxes for comments and proposed improvements. The scaling questions concerned the following dimensions: the design of the interface, the content of the information provided, the usefulness of the application, as well as the satisfaction with the response time. Questions are the following:

1. How did you like the application?
2. How did you like the design?
3. How did you like the navigation?
4. How did you like the information provided for the exhibits?
5. Were you satisfied with the response time?
6. Did it work correctly?
7. Did you find it easy to use?

8. Did you find it useful?

9. Would you use it as your guide?

The charts presented in Figure 6.12 show the results of the subjective evaluation for four questions.

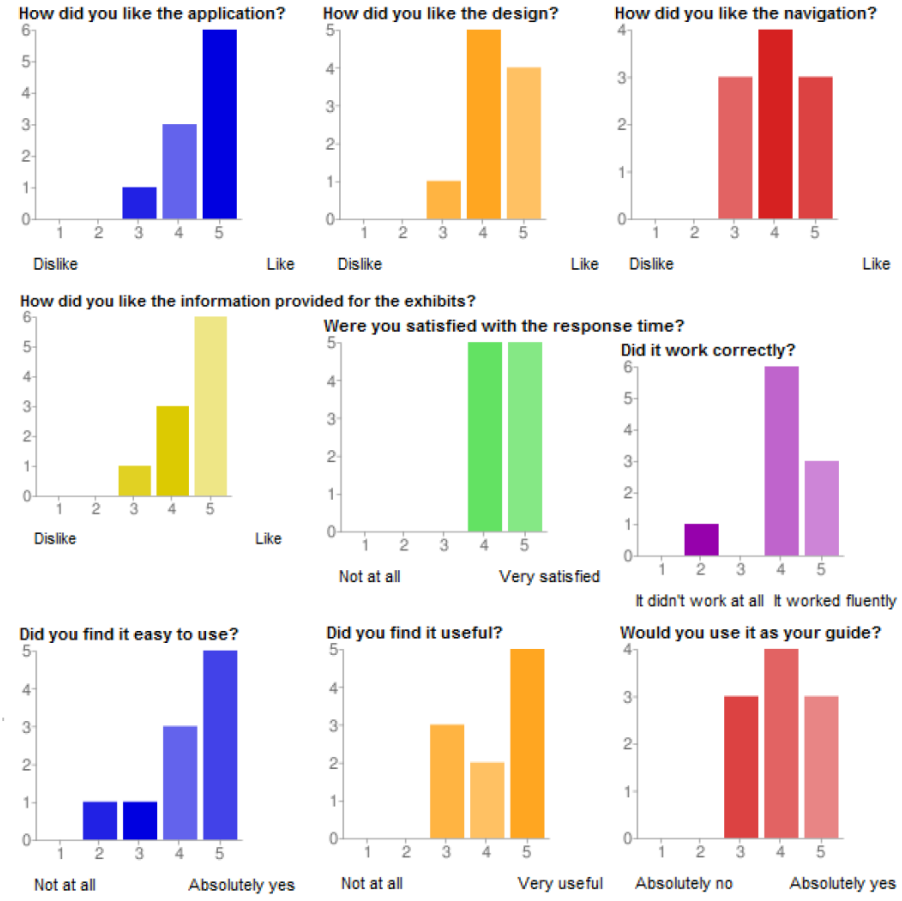


Figure 6.12 — Charts presenting the summary of responses.

The overall subjective ratings of the visitors for the Olympic Museum guide were considerably high (4.5 out of 5 on average) and the satisfaction with the response time is impressive (4.5 on average). However, the ratings concerning the question "Did you find the application easy to use?" varied more (4.2 on average). The lowest rating was given for the question "Would you use it as your guide?" (4 on average), but yet it is a satisfactory result considering that our application is a first implementation of a research product.

Overall, the user's attitude towards this type of mobile service was very positive. However it has to be noticed, that the experiment was conducted in the interaction between the researcher managing the test and the users, this may have biased the results.

The comments/improvements proposed by the participants are summarized in the following:

- One user proposed to make use of a rotating map according to the user's direction, so that the user can be easily oriented.
- Another user suggested to show the location of the user on the map.
- Another improvement proposed is the integration of interactive games.
- Two users would prefer the application to provide also interface in French and other languages.
- A useful improvement proposed is to build a trip in the museum according to the user's preferences and time.
- One user commented that the application may be too complicated for older people. This is true, but we mainly address people experience in using mobile phones.
- Finally, two users suggested the extension of the application for coverage of more exhibits and more information per exhibit.

6.2.6 Conclusion

In this Section an indoor guide, the Olympic Museum guide application was described as demonstration tool for object duplciate detection algorithm. The museum guide provides the user with audio-visual information concerning the exhibits. The application was evaluated on a real case scenario in the museum by visitors and the user's attitude towards this type of mobile service was very positive. However it has to be noticed, that the experiment was conducted in the interaction between the researcher managing the test and the users, this may have biased the results. It is impressive that 80% of the visitors rated the overall application with the highest rate. From the results of our objective evaluation we can conclude that using a client server approach, where the image recognition is done on the server side, and choosing a value for the resolution of the images close to 400×300 pixels as well as a sufficient number of training images per object (at least 10) leads to satisfactory and acceptable by the users response times, but also to sufficiently high recognition rate, which could allow the deployment of such an application to the museum.

6.3 Exchange calculation

6.3.1 Introduction

In this Section, mobile application is presented for currency exchange and counting, as shown in Figure 6.13. A specific object recognition algorithm can efficiently recognize banknotes and coins, and convert their values to other currencies using an external exchange rate API, which could be used on a trip, where the user has very few or no knowledge at all of the local currency. By just taking a picture of a certain amount of this foreign currency and using this application, he can automatically know how much it is worth in his own currency. Similar to the Chess mobile application, context based object recognition is applied, considering priory knowledge on coins and

banknotes. Regarding the application scenario, there is no problem with identification of multiple bank notes of same value, rotated or exclusion of non banknotes or non round objects. However round objects will be detected as coins if the size of these objects are similar to one of the coins.

Figure 6.13 — Mobile application for currency exchange and counting.

6.3.2 Related work

The challenge of recognizing money, be it coins or banknotes, finds many applications and has been the object of research many times. However, due to the intrinsic content of coins and banknotes, many methods do not deal with image recognition at all, or combine it with other techniques. For instance, the metallic properties of coins can be used in order to compute quantities such as its thickness, which cannot be achieved with 2D digital images.

Banknotes have often been the subjects of research for object recognition. In most currencies, banknotes are ideal for features extraction and recognition tasks, due to their high information and visual content. In [Lee and Kim, 2003], neural networks* (NN) are used in a context-based method

*A neural network is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. (Wikipedia)

specifically designed for Euro currency recognition. It takes into account the similar structure between various kinds of Euro banknotes to focus on a region of interest (located around the value of the banknote) in order to extract particular and distinctive interest points for each banknote.

The method proposed by [Choi *et al.*, 2006] describes a wavelet transform-based approach. Features are extracted from the high spatial frequency content of the banknotes. Even with a simple Euclidian distance-based classifier, this method has shown to achieve 99% recognition accuracy. In addition, it is not restricted to any currency and can be globally used with any kinds of banknotes.

Other methods, such as [Shan *et al.*, 2009], based on Hidden Markov Models* have shown to outperform related works among which NN-related approaches.

Just as for banknotes and due to their particular shape, many techniques have been proposed to detect coins. In [Chalechale, 2007], strong edges of the coins are extracted and their round shape is taken as an advantage through a spiral decomposition, based on circles of various sizes, in order to extract features that are scale-, translation- and rotation invariant from the coins. This method has shown to improve related approaches.

The method presented in [Fukumi and Omatu, 1993] builds a small neural network for coin recognition that automatically varies its architecture to fit a coin recognition environment. To do so, a generic algorithm varies the neural network architecture while its training is done through back-propagation. Other works, such as [Fuerst *et al.*, 2006], combine image processing-related techniques with optical approaches to recognize coins from more than 100 different countries. Criteria such as the diameter and the brightness of the coins are taken into account. The recognition step also considers the probability of appearance of each coin. This approach has shown excellent results but needs, as said above, the combination of several techniques that are not all image processing-related. Generally speaking, many proposed methods take advantage of the particular shape of coins to detect and recognize them through geometric considerations, when dealing with a sufficiently small set of possible coins.

The last remark needs to be fully considered, where the list of coins is small (Swiss currency has 7 sorts of coins) which have all different sizes. Difficulty in our project, compared to the state-of-the-art algorithm, that camera of mobile phone is used for detection considering real life situation. Concerning banknotes, those of Switzerland are no exceptions and possess a quite high visual content, which makes them candidates for features extraction and recognition approaches.

6.3.3 Proposed algorithm

System architecture of currency exchange and counting can be seen in Figure 6.14. Firstly, in our scenario, the user takes image from the money from above using monotone background. However perspective of the image (the image do not need to be taken from above) can be compensated by using the reference coin. In order to detect the objects, the background is subtracted by color segmentation. The objects boundaries are extracted, which allows to compute a *roundness* score for each object. If round enough, the object is matched against the coin database. If not, it is

*A hidden Markov model is a stochastic model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. (Wikipedia)

matched against the banknotes database. One of the approaches uses a reference, the MMSPG logo in this case, which explains the upper-left part of the flowchart. In the following, the details of these steps are described.

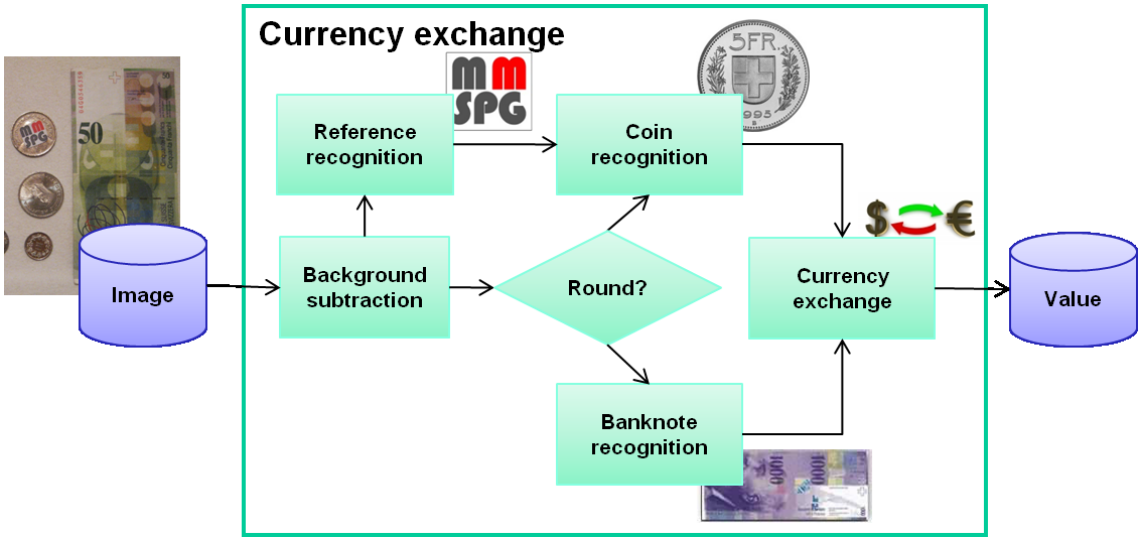


Figure 6.14 — Currency exchange and counting system architecture.

Background subtraction

In order to detect the objects in the input image, the color of the background is subtracted. Background's average color is extracted in very small locations on the borders of the image. Each color channel is then segmented in function of this average color, producing an RGB image. The resulting image in then converted to grayscale. In order to make it binary, a threshold is computed using MATLAB Otsu's method [Sezgin and Sankur, 2004], which chooses its threshold in order to minimize the intraclass variance among foreground and background pixels. Finally, potential holes inside objects are filled as explained and small blobs are removed using morphological opening.

Reference extraction

One of the examined algorithm uses a reference object for coin recognition. Reference object is in our case is a pre measured MMSPG logo, however it can be for example any bank card or other object where the size is standardized or known. In order to extract it in the binary image obtained after background subtraction, SURF feature extraction is performed on the original image, making it localizable among all detected objects by simple matching method proposed in [Bay *et al.*, 2006b]. Therefore, its dimensions in pixels can now be found and used as a reference between real-world metrics and pixels.

Objects classification

Among all objects extracted after background subtraction, some of them are coins, others are banknotes, and maybe some of them are just false detections due to noise or imperfect background subtraction. In order to classify each of them in one of those three classes, a roundness metric will be used. It depends on the area A and the perimeter P of the object, which can be computed as explained in the following.

In order to calculate the area and the perimeter of each object in a binary image, the first step consists in labeling them. This operation is known as blob coloring. The image is first scanned from the upper-left to bottom-right corner. In the end, each object is labeled and can be processed independently.

On the other hand, to compute the perimeter of an object in a binary image, the border pixels need to be extracted. Now that all objects are labeled, their border pixels can easily be extracted by direct 4-connectivity comparison. If at least one of the neighbors is black, the object is on the border. The perimeter "P" can then be estimated as the sum of all the pixel-wise Euclidian distances among the border pixels. On the other hand, the area "A" of the object is simply computed as the sum of all pixels belonging to the object.

Roundness metric, therefore can be calculated from the perimeter and the area.

$$Roundness(A, P) = \frac{4\pi A}{P^2} \tag{6.1}$$

It can be shown mathematically that such a metric can reach a maximal value of 1 only if the object is perfectly round. The rounder the object, the closer to 1 is teh Roundness value.

Therefore, it is now possible to classify the detected objects as round or not round. However, false detections will mostly be not round and will therefore be matched for banknote recognition, increasing needlessly the computation time. In the approach that uses a reference, it is possible to speed up the algorithm by classifying automatically small not-round objects as false detections by comparing their size with that of a banknote.

Coin recognition: Using a reference

The first approach to recognize coins uses a reference object. This object can be anything, since the only purpose of it is to obtain a relation between real-world metrics and pixels. It can for instance be a specific object provided to the user when he buys the application or any universal object such as credit cards, which have fixed known dimensions. In the case of this project, the MMSPG logo is used.

The size $S_{ref,mm}$ in real-world metrics (called mm) of the reference object being known and its size $S_{ref,px}$ in pixels being computed as explained in Section 6.3.3. It is therefore possible to determine a constant conversion factor in $pixels/mm$. Consequently, since the sizes $S_{obj_i,mm}$ of coins, with $i \in [1;7]$ corresponding to the 7 existing coins in the Swiss currency, in the original currency (Swiss francs in this case) are known in real-world metrics, their corresponding size in pixels can now also be known and compared to that of the tested object. Hence, if $S_{obj,px}$ is the size of the tested object in pixels, the best score is reached with the minimum difference of the

mm-based ratio versus the *pixels*-based ratio.

$$Score = \underset{i}{MIN}\left[abs\left(\frac{S_{ref,px}}{S_{obj,px}} - \frac{S_{ref,mm}}{S_{obj_i,mm}}\right)\right] \tag{6.2}$$

Objects tested for coin recognition have been tested for roundness. However, it might happen that a false detection was considered as round enough to be a potential coin. Therefore, the computed score is tested against a maximal threshold to make sure that the object is a coin.

Coin recognition: Using SURF and RANSAC

Using a reference can be annoying for the user. Therefore, another method, SURF with RANSAC, is used in order to recognize the coins. The use of RANSAC with SURF is done to ensure spatial coherence between the matched interest points. The score computation is similar as for banknotes, which is explained hereunder. Details on those two algorithms are provided in Section 2.

Banknote recognition: Using SURF and RANSAC

As for coin recognition, banknotes are recognized using SURF and RANSAC. In order to match the tested banknote with one of the database, three scores are considered.

After RANSAC is performed on the matched pairs of interest regions, a bounding box is built around the object. Swiss banknotes having a lot of similarities, many common keypoints are found between them. Hence, the use of RANSAC is necessary but not sufficient to ensure a perfect spatial coherency. If a bounding box is found, it is filled and convolved by the banknote's mask. If the result is high enough in comparison with the original area of the banknote, then the tested banknote is automatically recognized and the two other evaluation scores are not even tested. However, if the score is lower than the aforementioned threshold or simply if the bounding box has not even been able to be computed, the second score is considered.

The second score uses a RANSAC-like spatial coherency computation, since perfect coherency can seldom be achieved, as it has been explained above. It takes advantage of the flatness of the banknotes and therefore it is even more severe than RANSAC. For each possible triplet of matched pairs, a triangle between their locations in the reference image as well as in the tested image is considered. If the ratio between their sides is close enough, a counter is incremented. After all possible triplets have been considered, the score is the value reached by the counter. If this score is high enough, the tested banknote is automatically recognized and the last evaluation score is not even tested.

If both previous scores have failed, the last score is considered. It simply consists in the number of matched pairs found between the reference banknote and the tested object. The banknote with the highest score is chosen.

Banknote recognition: Using color histograms

A second method has been tested for banknote recognition since Swiss banknotes have quite different colors. The locations of peaks in each color channel histogram are found. The score is based on the distance between those locations and those of the reference banknotes, as shown in

Figure 6.15. If the distances between the tested maximums and those of a reference banknote in the database are small enough, the banknote is recognized.

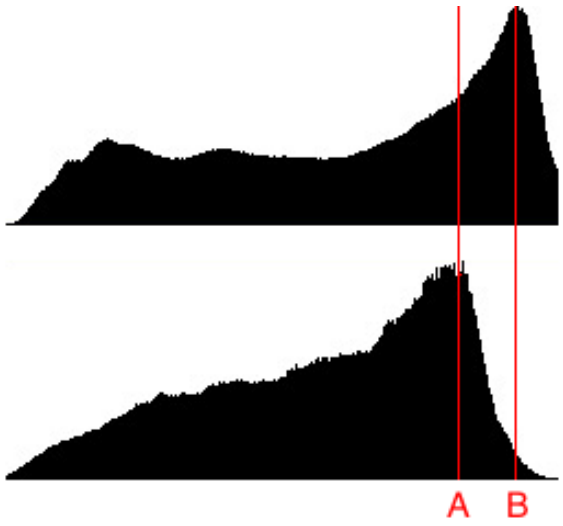


Figure 6.15 — Distance between peaks in color histograms.

6.3.4 Database

The database contains 73 images, captured by a mobile camera. Some examples are shown in Figure 6.16. Some assumptions have been made based on those images:

- There is no overlap between objects.
- The background is uniform.
- The banknotes are laid flat.
- The borders are free from objects.

Swiss francs banknotes contain a common small text part which produces logically many interest points when using SURF. In order to avoid meaningless matching (since all banknotes contain that same small text), the letters have been blurred in the reference images. For the same reason, the dates appearing on the Swiss coins have also been blurred, since they represent the year of manufacture of one specific coin and therefore are completely irrelevant with the value of the coin.

6.3.5 Results and analysis

For all images, the objects detection process, such as the extraction of the masks of the objects through background subtraction and image refinement, has shown a 100% detection rate for all objects of all images. The results obtained with this application are excellent. The objects detection has shown to be perfect, additionally the objects recognition's performance is extremely accurate.

Figure 6.16 — Examples of images of the database.

Figure 6.17 show the results for banknotes recognition with the two tested methods, SURF and color histograms. SURF method with spatial validation is outperforms color based method as the previous method worked perfectly in our database. The banknotes recognition using SURF has simply produced no error. The high visual content of these objects provides many interest points and is therefore perfectly suited for features extraction techniques. On the other hand, using color histograms has shown less accurate but promising results. The main challenge with color-based techniques resides in the control of the lighting conditions. With various illuminations, color histograms can get narrower/wider and be shifted, which makes the matching process rather complicated.

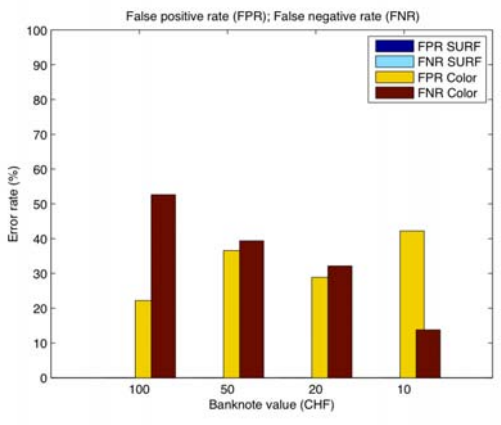


Figure 6.17 — Comparison between both methods for banknotes recognition by FPR and FNR value.

Figure 6.18 show the results for coins recognition with the two tested methods, using a reference and SURF. Algorithm, using diameter information from the coins, performs almost perfect. The coins recognition using a reference for radius comparison has proved to be a simple and very robust way of recognizing coins. The larger coins (5 CHF, 2 CHF, 1 CHF, 0.20 CHF) have shown a 100% recognition rate while the smaller coins (0.50 CHF, 0.10 CHF, 0.05 CHF), whose sizes are very close (18, 19 and 17 mm respectively), have only met 4 missed assignments all-together. On the other hand, using SURF to recognize coins has shown to be much less effective. Two main reasons can explain those results. The first one is clearly the similarity between the coins. Indeed, the heads sides of almost all coins are simply identical (Figure 6.19 (a)), which makes the recognition process completely random in-between those coins. Additionally, the tails sides are also extremely similar since they all show a Laurel wreath pattern, as illustrated in Figure 6.19 (b). The second main reason explaining this low performance comes from the size of the objects, which are relatively small and therefore cannot provide many interest points for the SURF detector.

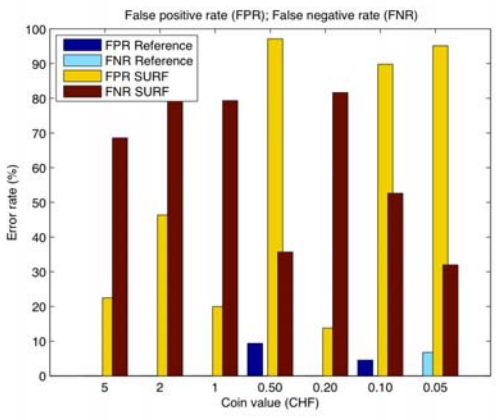


Figure 6.18 — Comparison between both methods for coin recognition by FPR and FNR value.

Figure 6.20 shows the global results achieved for the Currency exchange and counting mobile application when using SURF and RANSAC method for banknotes recognition and diameter for coins recognition.

Money value [CHF]	100	50	20	10	5	2	1	0.50	0.20	0.10	0.05
Occurrences	40	35	35	36	36	43	61	53	50	52	59
Missed recogn.	0	0	0	0	0	0	0	0	0	0	4
False recogn.	0	0	0	0	0	0	0	3	0	1	0
Recognition rate [%]	100	100	100	100	100	100	100	94.3	100	98.1	93.2
Error value [CHF]	0	0	0	0	0	0	0	0.029	0	0.002	0.003

Table 6.1 — Quality of service. Average error shown in CHF.

The Table 6.1 shows the number occurrences in our database, missed and false detections that happened for each object (banknote or coin). Global recognition rate RR is then computed for each of them with respect to the number of occurrences of this object in the images of the database. Additionally, the error value for each object is considered and computed as $(1-RR)\cdot ObjectValue$. Last but not least, the total error expressed in CHF. Over the 73 images of the database, the average

Figure 6.19 — Similarities between Swiss coins. Heads (a) and tails (b) are shown.

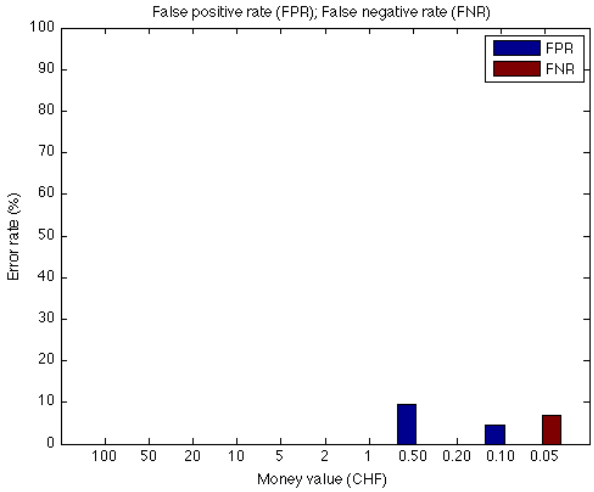


Figure 6.20 — Global method evaluation when using a reference for coins recognition and SURF for banknotes recognition by FPR and FNR values

error is therefore 0.02 CHF with a variance of 0.008 CHF^2.

All in all, from the point of view of the user, the quality of service has shown to be extremely reliable, with an average error of only 0.02 CHF per image.

6.3.6 Conclusion

The goal of this research was the implementation a mobile-intended application for currency exchange and counting. It has been designed on context-based object duplicate detection fundaments, that is: knowing the *context* provides *a priori* knowledge on the *content* of the targets to be detected and recognized. For instance, an use of context-based knowledge has been done in this application by taking into account the relative hight information of the coins.

The currency exchange and counting application has shown to be 100% effective for banknotes recognition and has proved to be almost perfect in recognizing coins by geometric considerations. Additionally, from the point of view of the user, this application has shown to be very reliable, since 95% of the tested images, which contained up to 7 objects (banknotes and coins) in average, have been recognized without any error. Above all, since all errors happen with small value coins (whose sizes are more similar, therefore more complex to differentiate), the average error is only of 0.02 CHF per image, with a variance of 0.008 CHF, which is extremely small.

To improve the accuracy several images or video can evaluated.

6.4 Epitome

6.4.1 Introduction

Rapid growth of digital photography in recent years has increased the size of personal photo collections. People use their digital cameras or mobile phones equipped with cameras to take photos. Beside storing them on computer hard drives, people also share their digital photos with friends, family and colleagues through social networks. Facebook*, Flickr† and Picasa‡ are examples of such photo sharing web sites. Some people also print their photos on post cards, calendars or photo books, often to give them as presents or to create physical souvenirs.

There is a saying: "A picture is worth a thousand words." Therefore, people like to use their photos to tell their own stories of some important events in their lives. One's wedding, birth of a baby, vacation, birthday party or even a long lasting period - from the date of one's birth till celebration of 18th birthday, are only a few examples of such events. One of the reasons why people share photos is to ask their friends to comment and tag photos.

Users usually organize their photos in albums (collections) based on places, events or people. By sharing these albums with others, they want to tell their own stories of some important events in their life, such as birthday party, vacation, wedding, or birth of a baby. It can be very time-consuming to go through all photos in one album, and therefore summarization is an effective way to help getting a quick overview of a set of photos. Album summarization can be defined as

*http://www.facebook.com
†http://www.flickr.com
‡http://picasa.google.com

selecting a set of photos from a larger collection which best represents the visual information of the entire collection. Selected photos can be used to create a collage of a given album, a cover for an album, or to be included in a photo book.

Beside spending a lot of time sharing and consuming content in online social networks, people also use online applications, especially social games. Players pour huge amounts of time and effort into games. For example, the recent survey [Mike Snider, 2011] revealed that most players (95%) play social games several times a week, with 64% playing daily. The average game session lasts more than half an hour (that is how long 61% play), while 10% may play more than 3 hours at a time. Work by von Ahn *et al.* [von Ahn and Dabbish, 2004] showed the tremendous power that networks of people possess to solve problems while playing social games. Therefore, the time and effort in playing a game can be utilized to address some issues in image processing community, i.e. users entertain themselves while playing an enjoyable and funny game, with the added side-effect that they are doing useful work in the process, for example, summarizing one's photo album. This was one of the motivations to develop a novel approach for photo album summarization through gaming.

Current state-of-the-art techniques are based on automatic summarization which considers time separated events, spatial information using GPS coordinates and content-based image similarities. Naaman *et al.* [Naaman *et al.*, 2004] developed a system which does automatic organization of digital photographs considering the geographic location of photo or event based description extracted from user tags. Combination of spatial, temporal and content-based similarity is then used for photo collection clustering. This clustering can be used for photo navigation and search for different categories, such as elevation, season, time of the day, location, weather status, temperature and time zone. Once photos are clustered, different page layouts should be considered. Geigel and Loui [Geigel and Loui, 2003] emphasized aesthetic side of a page layout for image collections. They used a genetic algorithm to optimize aspects such as balance and symmetry for a good placement of images in the personalized album pages. An automatic summarization has its limitations. There is a gap between what people think the summary should look like and what we get with an automatic summarization.

Ames and Naaman [Ames and Naaman, 2007] showed that providing incentives to the user in form of entertainment or rewards, e.g. games, can motivate them to tag photos in online and mobile environments. Gaming also provides a new way of motivating people to make the subjective data acquisition interesting and enjoyable. The most famous examples of these kinds of games are the ESP Game and Peekaboom, developed for collecting information about image content. In *ESP Game* [von Ahn and Dabbish, 2004], two players, who are not allowed to communicate with each other, are asked to enter a textual label which describes a shown image. The aim of each user is to enter the same word as his/her partner in the shortest possible time. In *Peekaboom* game [von Ahn *et al.*, 2006], one player is given a word related to the shown image, and the aim is to communicate that word to the other player by revealing portions of the image, while the second player sees an empty black space in the beginning. Our social game can collect research data and, at the same time, it provides a collage or a cover photo of the user's photo album, while, at the same time, the user enjoys playing a game. In this way, both users and research community can benefit.

In this Section, we analyze an approach for photo album summarization through a novel social mobile game "Epitome" [Vajda *et al.*, 2011b]. The main idea of our approach is to show a reduced set of photos from a Facebook album, ask users to play the game and then integrate results of all users in order to produce a summarization for the whole album. Moreover, we compare results obtained by this game with an automatic image selection, making use of visual and time features, which make us demonstrate a simple task as photo album summarization where humans outperform the existing automatic computer vision algorithms. Therefore it shows that photo album summarization is complex and difficult processing task in the human brains, and that there are still big gap between computer and humans in image processing and retrieval. This game shows a way for solving a complex problem by separating or reshaping the problem for people without deep knowledge on the complex problem.

6.4.2 Epitome game

The goal of this application is to provide an intuitive and enjoyable user interface as a Facebook application, which creates and annotates photo collages for Facebook photo albums. Therefore, the game "Epitome" is created, which can provide its potential users with many pleasant hours while playing it, and enjoying photos. At the same time, it determines the most representative photos of a user's photo album and provides useful research data [Vajda *et al.*, 2011a,c].

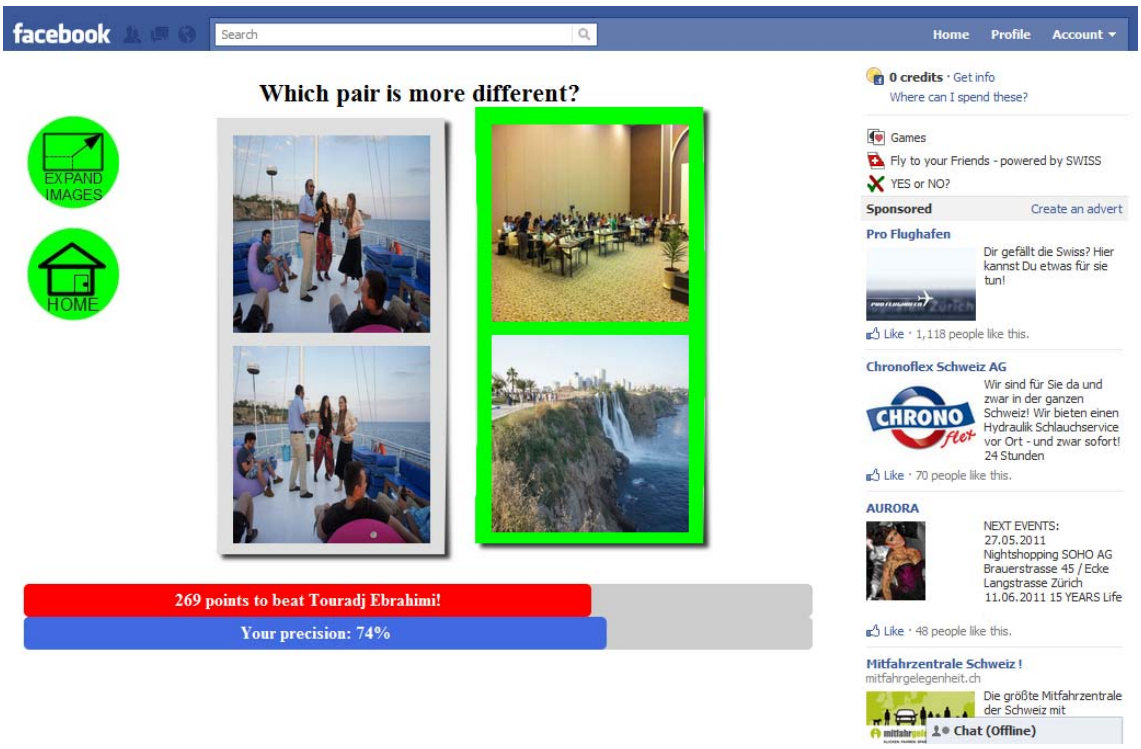


Figure 6.21 — Screenshot from Epitome game.

The scenario of the game is as follows. A Facebook user, in this chapter denoted as a player,

installs the game in his/her Facebook applications page and allows access to his/her photo gallery, as shown in Figure 6.21. Then, the player can select between two games. In the first game, called "Select the Best!", two random photos are shown to the player from one of his/her friends' photo albums chosen randomly and he/she has to choose the better photo. If the player chooses the photo which is the most frequently selected by his/her friends, then player's score increases. The second game is called "Split it!". In this game, two pairs of consecutive photos are shown, where the player should select the photo pair which is more different. The time stamp is extracted from EXIF tags associated to each photo, which corresponds to the time order by which photos were uploaded to Facebook. The results of "Select the Best!" and "Split it!" games are combined to form a score and if a user reaches a certain score level, then the photos for the collage of the user's photo album are shown to the owner. Therefore, the player can get a feedback from all other players, regarding his/her Facebook photo albums. The game has appealing look using different visual and audio effects based on desktop and mobile platform, as shown in Figure 6.21.

In order to perform summarization using players' inputs, the application calculates three different values: $Importance$, $Segmentation$ and $UserScore$.

Importance value is determined in the "Select the Best!" game for each photo album separately. The goal of this game is to select the most representative photos of the particular Facebook album. Two randomly chosen photos are shown to the user and he/she selects the better one in his/her opinion. A feature vector $Selected_n$, $n \in [1 \dots N]$, is calculated for each player, n among N players, as follows:

$$Selected_n[i] = \delta_{i,s} \tag{6.3}$$

$$Appeared_n[i] = \delta_{i,j} + \delta_{i,s} \tag{6.4}$$

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{cases} \tag{6.5}$$

where $i, j, s \in [1 \dots M]$, M is the size of a particular Facebook album, j, s are indices of the two photos shown to the player and s is index of the selected photo. i is the index of any image. The vector *Appeared* of dimension M stores the frequency of all photos that appear in the game. At the end, we perform normalization on vector by element-wise division:

$$Importance[i] = \frac{\sum_n Selected_n[i]}{\sum_n Appeared_n[i]} \tag{6.6}$$

which is an M-dimensional vector showing the distribution of the most representative photos within one Facebook album.

Segmentation vector is calculated in "Split it!" game for each photo album separately in an analogous way as explained for *Importance* value. It shows the frequency with which each photo in one album is selected as a starting photo in a new segment.

Finally, vectors $Importance$ and $Segmentation$ are used to automatically select $L = 5$ most representative photos within one Facebook photo album. At first, the particular album is segmented into L most probable segments by determining $L - 1$ maximum values from the vector

Segmentation. For each of these segments, a photo with the highest score in the vector *Importance* is chosen. These L photos represent a collage of the album, which is shown to the owner of that album, if he/she reaches a certain level of $UserScore$.

UserScore value is defined to motivate players to play this game frequently. For example, in the "Select the Best!" game, the player increases his/her own $UserScore$ if he/she selects the photo which has the highest *Importance* value among two photos. The same approach is used in "Split it!" game, where the player increases his/her $UserScore$ if he/she select separation place where *Segmentation* value is the highest among two separation places. Initial $UserScore$ is set to 0. $UserScore$ values for all players are sorted to show ranking of players in "Epitome" game.

6.4.3 Automatic photo album summarization

Automatic photo album summarization is performed considering different visual and temporal features, which were described in Section 2.4. After extracting these features, the album is segmented into five parts by calculating the four highest Euclidean distances of the consecutive photos' features. For each image in the particular segment, we calculate the sum of the Euclidean distances between that feature of the photo and the rest of the image features in the segment. The image with the lowest sum is then selected as the most representative photo in that segment. Different features can be used for segmentation and to select the most representative photo in the segments. Therefore, we calculated the performance of 20 different feature pairs among the "Bag of Words" method based on SURF features, "Histogram of Oriented Gradients", "HSV Color histogram" and "Tiny" features, as described here.

Tiny feature is used as benchmark representing scaled 32×32 grayscale tiny images. The dimensions of the features are around 1000.

Time stamp is extracted from EXIF for further analysis.

Color histogram descriptor is extracted from photos in HSV domain. Color descriptors often fail in image retrieval in different light conditions, however in our case consecutive photos were compared in an album, therefore the light conditions are similar for each photo.

Bag of Words model in computer vision was derived from BoW model in natural language processing (NLP) [Fei-Fei and Perona, 2005]. Similar method in computer vision documents represents images or objects, and visual clusters of local features are considered as a word. In our case, SURF features were used as local features [Bay *et al.*, 2006b]. Thus, BoW is a vector which represents the histogram of visual features. Therefore, this method does not consider spatial information or order of visual features. 1000 feature clusters were calculated by hierarchical k-means algorithm. Then each feature of an image is represented by 1000 normalized value, depending on how many local features are containing each class of feature. By pair wise comparison, the distance of the query and training features is represent the scoring value.

Histogram of Oriented Gradients [Dalal and Triggs, 2005] is similar to SIFT and SURF region descriptors. It calculates the histogram of gradients in the region around the one keypoint. It is evaluated on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. Using gradient information for feature description is very robust to different illumination conditions.

6.4.4 Evaluation

The summarization performance of the "Epitome" is evaluated with respect to the ground truth given by humans.

Collecting ground truth data

Ground truth was constructed by asking different people for their subjective opinion about photos and then tested our algorithm against the ground truth data. We recruited 63 participants, among whom 61% were males and 39% were females, aged $18-65$, with different backgrounds and cultural differences. In the collection of the ground truth data, participants were shown 20 photos which belong to the same dataset (collection or album). The task of the participants was to select the 5 most representative photos of the whole album, while looking at all photos of that album.

For simplicity of the explanation on how this approach was evaluated, let us consider only one dataset with M photos. First, a ground truth data is collected. Every user n among N users is asked to select the 5 most representative photos. After his/her participation in collecting the ground truth data, the corresponding feature vector $Selected_n$, $n \in [1, N]$, is formed as follows:

$$Selected_n[i] = \sum_{k \in [1 \ldots 5]} \delta_{i,s_k} \tag{6.7}$$

$$\forall k, l \in [1 \ldots 5], s_k \neq s_l \tag{6.8}$$

where $s_k \in [1, M]$ are the five indexes of the photos, which were chosen as the representative photos. The selected indexes are distinctive.

Feature vectors of the users $u, v \in [1, N]$, are then compared to each other and the score of their matching $S_{u,v}$ is calculated as:

$$S_{u,v} = Selected_u \cdot Selected_v^{\mathrm{T}} \tag{6.9}$$

In other words, the higher the number of identical photos that are chosen by two users, the better will be the score of the match between them. Note that the maximum score of the match is 5. Finally, to each user i, $i \in [1, N]$, a value $Score_i$ is assigned as:

$$Score_i = \sum_{j=1}^{N} S_{i,j} \tag{6.10}$$

The maximum value in the vector $Score_i$ shows the best performing participant who has the highest number of selected photos which are matched with all other users. The maximum possible value of the score is $5XN$, which in our case becomes 315. These results are considered as the ground truth data and compared with the results obtained from the games in order to prove the concept of the approach. All computations are repeated in a similar way for all datasets.

Experiments

To collect the ground truth data and to evaluate the designed photo selection tool (social game), we conducted two experiments. Since there are different criteria upon which a human user would rate digital photos, we first constructed a ground truth by asking different people for their subjective opinion about photos and then tested our algorithm against the ground truth data. We recruited 63 participants, among whom 61% were males and 39% were females, aged $18 - 65$, with different backgrounds and cultural differences.

In the collection of the ground truth data, participants were shown 20 photos which belong to the same dataset (collection or album). The task of the participants was to select the 5 most representative photos of the whole album, while looking at all photos of that album.

Then, participants were asked to play two games "Select the Best!" and "Split it!" with a dataset from Section 6.4.4. The results obtained from these games are used to assess the performance of our approach by comparing them with the ground truth and results from automatic visual analysis.

Furthermore, we performed automatic photo album summarization considering different visual and temporal features. At first "Bag of Words" method based on SURF features, "Histogram of Oriented Gradients", "HSV Color histogram" and "Tiny" features are extracted. Where "Tiny" feature is used as benchmark representing scaled $32X32$ grayscale tiny images. The dimensions of the features are around 1000. Moreover creation time stamp is extracted from EXIF for further analysis. We segment the album into 5 parts by extracting the four highest Euclidean distances of the consecutive photos' features. For each image in the particular segment, we calculate the sum of the Euclidean distances between that feature of the photo and the rest of the image features in the segment. The image with the lowest sum is then selected as the most representative photo in that segment. Different features can be used for segmentation and to select the most representative photo in the segments. Therefore we calculated the performance of 20 different feature pairs.

Dataset

The dataset used in this experiment is the official dataset from "HP Challenge 2010: High Impact Visual Communication" at the "Multimedia Grand Challenge 2010" [Multimedia, 2010]. The dataset of photos used was fixed for all participants and therefore it can be used for further comparisons. Some example photos are shown in Figure 6.22. It consists of 6 datasets, each with 20 photos. These datasets cover photos that are usually taken during a vacation, describing a variety of topics: photos depicting different landmarks and famous sightseeing places, photos with parents and kids, and photos of cars, flowers and sea animals. Figure 6.4.5 provides example photos of the datasets.

6.4.5 Results and analysis

Automatic photo album summarization was performed considering different visual and temporal features, using "Bag of Words" method based on SURF features, "Histogram of Oriented Gradients", "HSV Color histogram" and "Tiny" features as described in Section 6.4.3.

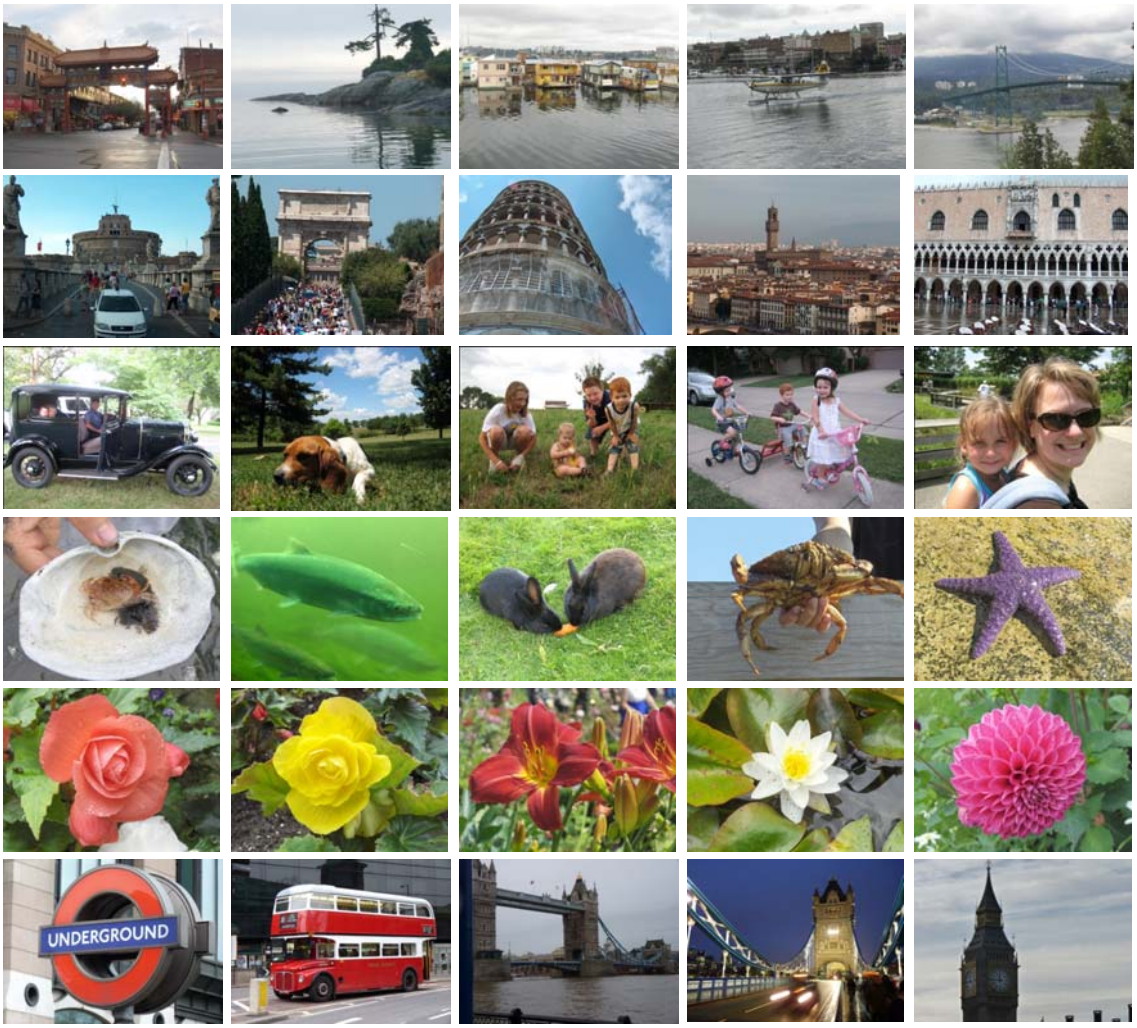


Figure 6.22 — Some example photos for each of 6 datasets. Photos in each row belong to different datasets. The datasets cover a large variety of objects and scenes usually taken during a vacation.

We calculated the performance of these 20 different feature pairs, separately for segmentation and choosing the representative images as shown in Figure 6.23. The result shows that the best performance is achieved by the pair of "Color histogram" for album segmentation and best photo selection in the segment. This highlights the robustness of the features obtained with the "Color histogram" method used with the given database.

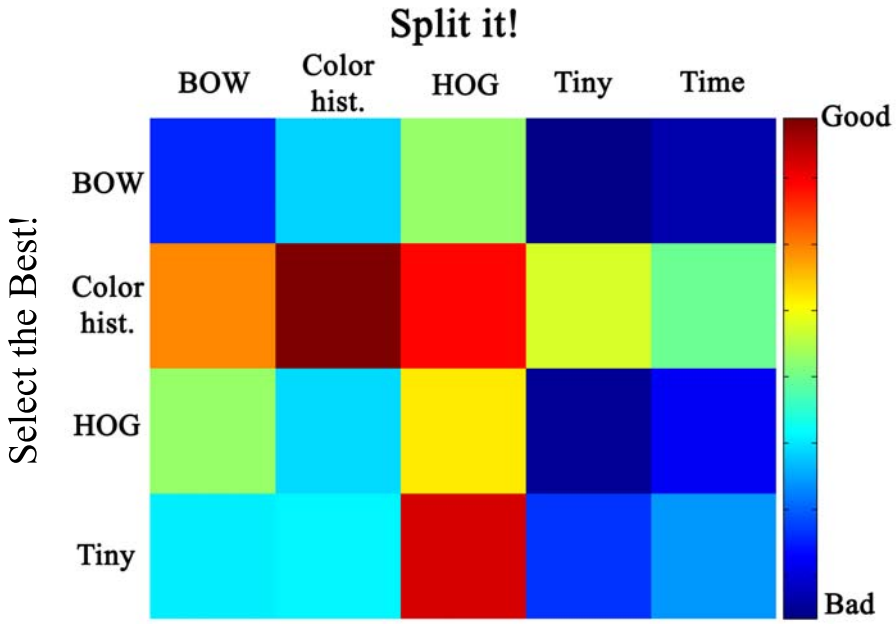


Figure 6.23 — Comparison between different visual feature. The best result is achieved with "color histogram" feature for "Split it!" and also for "Select the best!" task. Dark red color indicates the best and blue color indicates the worst performing algorithm. For example, using "time" feature for Segmenting the database (Split it!) and using "BOW" feature for selecting the most representative images (Select the Best!), gives poor results on evaluation.

Figure 6.24 shows the distribution of the participants' scores, including the choice of the proposed method and the automatic visual analysis. All scores are sorted in a descending order. These results look promising. As we can see, the scores of the proposed method have a small relative distance from the best ground truth scores achieved in our experiments. In average, this approach achieves 80% of the best score for each dataset, which proves the concept of the game. It also outperform the automatic visual analysis, which can achieve score of 64%. For datasets 3 and 5, this value is even higher, i.e. about 95%. The most representative photos for one of the datasets selected by the proposed method are shown in Figure 6.25.

Advantages and disadvantages

In summary, the "Epitome" game has the following advantages:

1. Performance of the game-based album summarization is better than using only computer vision approaches, which was shown in [Vajda *et al.*, 2011b].

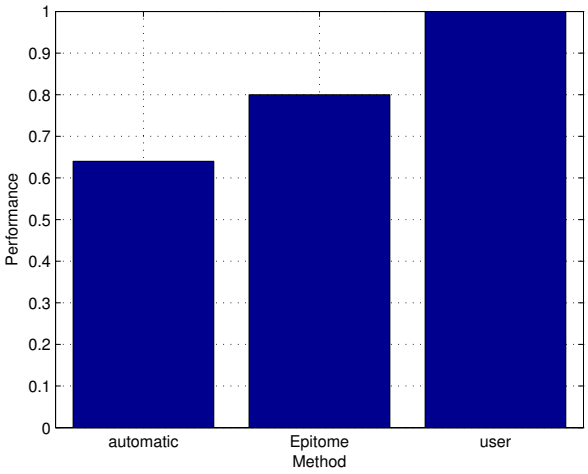


Figure 6.24 — The results of the proposed method, automatic visual analysis and ground truth data by users survey. The results are promising and prove the concept of the approach.

Figure 6.25 — Photos from the dataset 3. The most representative photos selected by the proposed method are marked with green bounding box, while the red bounding box denotes photos selected by making use of color histogram.

2. People like to watch their friends' photos through this game, which also encourages social interaction between people.

3. The game itself is interesting and people can have fun through the game.

However, a disadvantage is the processing time for generating fine album summarization, as shown in [Vajda *et al.*, 2011b].

6.4.6 Conclusion

In this research, we analyzed a social mobile game for an album summarization on Facebook. The proof of concept of these games was demonstrated and validated through a set of experiments on several photo collections. The results of our experiments show that our summarization game achieves 80% of the best score of different participants and significantly outperforms automatic visual summarization methods, which achieved 64%. However it should be mentioned that summarization by Epitome takes much more time than it takes for automatic album summarization methods.

As a future study, we will include in our approach more sophisticated visual analysis and make the game more attractive for users.

6.5 Chapter summary

In this chapter, we proposed four mobile applications for object duplicate detection to demonstrate that user interaction, environment/context improve the detection accuracy compared to the pure content-based algorithm.

First we have analyzed and described a content-based web navigation application for mobile phones, which application has a great capacity for gaming, education and personal usage.

As further improvement, a mobile museum guide application is developed for Olympic Museum, based on object duplicate detection. The application was evaluated on a real case scenario in the museum by visitors and the user's attitude towards this type of mobile service was very positive.

Mobile application for chess recognition has been designed on context-based object duplicate detection fundament, which is: knowing the context provides a priori knowledge on the content of the targets to be detected and recognized by taking into account the rules of chess. The proposed method for chessboard detection and localization has shown to be very robust, with a 98% accuracy. Chess figures recognition based on heights comparisons through a perspective transformation, is promising.

Mobile-intended application for currency exchange and counting is developed based on coin and banknote recognition using content and contextual information by taking into account the relative hight information of the coins. This application has shown to be 100% effective for banknotes recognition and has proved to be almost perfect in recognizing coins by geometric considerations. Additionally, from the point of view of the user, this application has shown to be very reliable since 95% of the tested images have been recognized without any error. Above all, since all errors

happen with small value coins, the average error is only of 0.02 CHF per image, with a variance of 0.008 CHF.

Finally, we proposed a social mobile game for an album summarization on Facebook. The results of our experiments show that our summarization game achieves 80% of the best score of different participants and significantly outperforms automatic visual summarization methods, which achieved 64%. Moreover, we compared results obtained by this game with an automatic image selection, making use of visual and time features, which show that this simple task are solved by complex processing in the human brains, and there are still big gap between computer and humans in image processing and retrieval. We are contributed for closing this gap, however visual search and retrieval needs still need more research.

I hear and I forget. I see and I remember. I do and I understand.

Attributed to the Greek philosopher Confucius (*cerca* 551 B.C. — 479 B.C.)

7 General Conclusions

7.1 Summary of the achievements

In this dissertation, a novel graph-based approach is proposed and analyzed for 3D object duplicate detection in still images and videos [Vajda *et al.*, 2009b]. This approach combines the efficiency with the accuracy, by making an attempt towards 3D modeling, while keeping the efficiency of 2D processing. A graph model is used to represent the 3D spatial information of the object based on the features extracted from the training images so that we can avoid explicitly making a complex 3D object model. Therefore, improved performance is achieved in comparison to existing methods in terms of robustness and computational complexity. Another advantage of our method is that it requires only a small number of training images in order to build a robust model for the target object. Usually, several images from different views of an object are needed to create its 3D model. However in our approach, only a few common features are necessary to link spatial graphs from different views; therefore fewer training images are needed for the model creation. The method is evaluated through a comprehensive set of experiments, in which an in-depth analysis of its advantages and limitations is performed and optimal algorithm parameters are derived from the analysis. A comparison with the state-of-the-art best-performing methods shows its significant performance improvement, because unlike our method, they consider a 3D object as 2D. Furthermore, we analyzed how synthetic training images can be created through an affine transformation in order to decrease the number of captured training images.

Our proposed graph-based approach is extended and analyzed for 3D object duplicate detection in video. Objects are detected in video content iteratively in order to compensate for 3D view variations, illumination changes and partial occlusions. Given a query image with the object of interest, the proposed system retrieves key frames with duplicates of that object. Due to invariance of the object duplicate detection approach to minor appearance changes, the retrieved frames

usually contain also variations from the object of interest. Therefore, the retrieved objects are considered as iterative queries to retrieve object duplicates with larger variations. For example, given the frontal view of a car as the initial query, the iterative query mechanism can retrieve the back side of the car if intermediate views of the car are available in the video clip. Considering video, the recall value of the object duplicate detection in video is improved by a factor of 2 using the iterative detection procedure in comparison to the non-iterative object duplicate detection algorithm, while the precision value is kept around 90%.

Two levels object detection was proposed for efficient search in a large scale database. First, a fast image selection algorithm eliminates images which do not contain the query object, and then accurate object duplicate detection algorithm provides bounding boxes. The annotation can be performed through a tag recommendation process, in which the system recommends tags associated with the object in the images of the search results, or through tag propagation process, when the user enters his/her tag for the object and it is propagated to the images in the search results. It has been also shown that the detection works reliably for salient objects such as trademarks, books, newspapers, and gadgets.

We developed and analyzed mobile applications for object duplicate detection to demonstrate that user interaction, context improves the detection accuracy compare the pure content based algorithm.

First we have analyzed and described a content based web navigation application for mobile phones. Instead of a traditional text-based query which is quite inconvenient given the constraints of mobile phones, the query is simply formulated by capturing a photo of the object of interest. The search application then use that photo to find similar instances of that object in a database, and provides users with associated information, such as tags, descriptions, or links to web pages. This application has a great capacity for gaming, education and personal usage. Museum guide application was developed as demonstration tool for object duplciate detection algorithm on mobile platform. The museum guide provides the user with audio-visual information concerning the exhibits. The application was evaluated on a real case scenario in the museum by visitors and the user's attitude towards this type of mobile service was very positive.

The following two application demonstrate that object duplicate detection algorithm, using context based features and prior knowledge, can be more efficient and accurate. Mobile application for chess recognition has been designed on context-based object duplicate detection fundament, which is: knowing the context provides a priori knowledge on the content of the targets to be detected and recognized by taking into account the rules of chess. The proposed method for chessboard detection and localization has shown to be very robust, with a 98% accuracy. Chess figures recognition based on heights comparisons through a perspective transformation, is promising. Mobile-intended application for currency exchange and counting is developed based on coin and banknote recognition using content and contextual information by taking into account the relative size information of the coins. This application has shown to be 100% effective for banknotes recognition and has proved to be almost perfect in recognizing coins by geometric considerations. Additionally, from the point of view of the user, this application has shown to be very reliable since 95% of the tested images have been recognized without any error. Above all, since all errors happen with small value coins, the average error is only of 0.02 CHF per image, with a variance

of 0.008 CHF.

User-interaction help can improve the object recognition algorithm significantly as shown in the following two applications. In flower recognition application, segmentation step has been proven to be crucial and a marker-controlled watershed algorithm has been used. Moreover, several specific features were implemented for flower recognition, namely contour, texture, color based features. Results shows that the best feature is HSV color histogram with 69% precision. However weighted linear combination of features results with 80% precision. A social mobile game is proposed for an album summarization on Facebook. Our social game collects research data and, at the same time, it provides a collage or a cover photo of the user's photo album, while, at the same time, the user enjoys playing the game. As a benchmark comparison to this game, we performed automatic visual analysis considering several state-of-the-art features. The results of our experiments show that our summarization game achieves 80% of the best score of different participants and significantly outperforms automatic visual summarization methods, which achieved 64%. Moreover, we compared results obtained by this game with an automatic image selection, making use of visual and time features. It shows that this task are solved by complex processing in the human brains, and there are still big gap between computer and humans in image processing and retrieval. We are contributed for closing this gap, however visual retrieval still need more research.

7.2 Perspectives

Prior work on object detection shows the success of local visual features for visual search in large-scale database. However it may fail in detecting shiny objects, such as a car, a book cover, or it may fail on small objects like mouse, pen, license plate, etc. In the future work, we would like to investigate detection of such difficult objects. Our proposed future work is based on three pillars: *context information, multimodality* and *data acquisition.* The use of mobile phones in object duplicate detection can take better advantage of user interaction and contextual information, such as mobile phone's camera focal length, time, position, view direction, etc. Multimodality is a critical property to reduce the uncertainty and the search space in object detection by taking advantage of shape, text, and barcode recognition. Moreover considering different data acquisition techniques such as stereo image and video can capture richer visual information, which can lead to more robust object detection. This approach will be a step forward in the direction of building the future mobile augmented search engines on large scale databases. Examples of applications include information augmentation to get more information about specific objects, markerless augmented reality, and video surveillance.

Future work on object duplicate detection can consider and inspire from visual processing methods of the human brain. Human visual processing is very robust and efficient, using motion parallax cues, considering multi level features and environmental information for 3D object duplicate detection. Research can be carry out on bioinspired feature extraction and detection on single, stereoscopic images and video considering depth cues.

We believe that these challenging problems can produce a high-quality research by answering the following scientific questions:

1. **What type of objects can be detected accurately?** Certainly there are objects, which can be detected more accurately than others. Future work can explore which objects can be detected with what kind of object detection algorithms, considering different modalities, contextual information, and different data acquisition techniques.

2. **Is it feasible to implement real-time visual augmented reality application based on large-scale databases?** It is not easy to create very efficient and very accurate systems to search objects in large databases. However using combination of tracking algorithms along with object duplicate detection makes it feasible. Future work can explore the feasibility of this idea, and evaluate its performance.

3. **How is it possible to detect objects, which are not feasible to detect by using local features?** Future work could use contextual information, multimodality and data acquisition techniques to overcome the limitations of usage of conventional captured images based local features for object duplicate detection.

4. **Which contextual information can improve the detection accuracy?** Object duplicate detection algorithms can consider contextual information from the object, such as GPS location, mobile orientation, temperature, camera focal length, object position, etc. Future work can explore the impact of each of the above mentioned contextual information in the overall performance of the detection. These modalities are then combined and used for more accurate detection algorithms.

5. **How can multimodality improve the detection accuracy?** Object duplicate detection algorithms can use several modalities from the object, beyond conventional image features, such as text, faces, shapes, etc. These modalities can be combined and used for more accurate detection algorithms.

6. **How can advanced data acquisition improve the detection algorithm?** Alternative data can be used to improve the performance of the object duplicate detection. Future work can explore these opportunities, such as using several images from an object or video frames to improve the detection or to increase the usability of the application. Moreover stereo-images, depth images, or video can be captured with the future or current mobile phones, such 3D mobile phones.

7. **How to combine/fuse these methods for general visual search algorithm?** Several modalities, context based features and data acquisition types could be studied during detection of different objects. Different methods of data fusion will be applied to combine the information and to verify if this additional information can improve the performance of the system.

A mathematician is a device for turning coffee into theorems.

A matematikus egy gép csupán, amely az elfogyasztott kávémennyiséget elméletekké alakítja.

Paul Erdos (1913 — 1996)

Evaluation methodologies

A

The goal of evaluation is to assess the performance of a method to make it comparable to other methods [Fawcett, 2006]. Performance of the object duplicate detection can be evaluated as a typical detection problem where the set of predicted objects is compared against a set of ground truth objects.

A.1 Confusion matrix

The detection task can be evaluated using correspondences between a set of predicted objects which are represented by their bounding boxes, and a set of ground truth objects. A pair-wise comparison of ground truth (gt) and predicted ($pred$) objects is performed in order to see if they are the same or not. If the ratio between the overlapping area and the overall area exceeds a certain threshold, in our case it is 50%, it is considered as a match as shown in Figure A.1.

$$d_{ij} = \frac{area(gt_i \cap pred_j)}{area(gt_i \cup pred_j)} \tag{A.1}$$

where i and j are index of objects.

The results and the ground truths are used to obtain the following values:

1. *True positives (TP):* The number of correct predictions that an observation is positive.
2. *True negatives (TN):* The number of correct predictions that an observation is negative.
3. *False positives (FP):* The number of false predictions that an observation is positive. Also referred to as type I error in statistics.

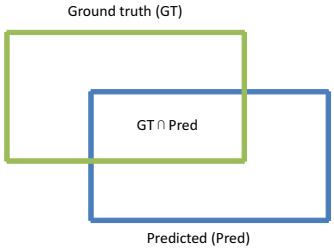


Figure A.1 — Overlapped area calculation from predicted and ground truth bounding box.

4. *False negatives (FN):* The number of false predictions that an observation is negative. Also referred to as type II error in statistics.

Table A.1 — Confusion matrix

		Ground truth	
		Positive	Negative
Prediction	Positive	TP	FP
	Negative	FN	TN

This confusion matrix (Table A.1) serves as a basis on which Receiver operating characteristic (ROC), Precision-recall (PR) curves can be derived as shown in Figure A.2.

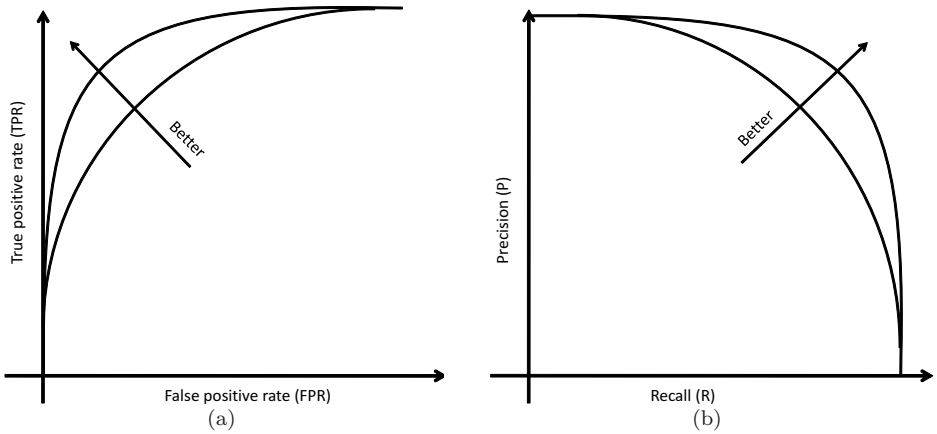


Figure A.2 — Receiver operating characteristic (ROC) curve and Precision-recall (PR) curve schame.

A.2 Receiver operating characteristic (ROC) curve

The receiver operating characteristic (ROC) curve represents the true positive rate (TPR) versus the false positive rate (FPR), with:

$$TPR = \frac{TP}{TP + FN} \tag{A.2}$$

$$FPR = \frac{FP}{FP + TN} \tag{A.3}$$

A.3 Precision-recall (PR) curve

The precision recall (PR) curve plots the precision (P) versus the recall (R) with:

$$P = \frac{TP}{TP + FP} \tag{A.4}$$

$$R = \frac{TP}{TP + FN} \tag{A.5}$$

This curve does not consider TN which is not uniquely defined for detection problems, because true negative detection could be any bounding box on the image which are different from the ground truth.

A.4 F-measure

In order to determine a single performance number for object detection, the F-measure is calculated as the harmonic mean of P and R values, given by:

$$F = \frac{2PR}{P + R} \tag{A.6}$$

which considers P and R equally weighted. The more general definition of the F-measure considers the weighted values of P and R, and measures the effectiveness of the object duplicate detection algorithm with respect to a user who gives β times the importance to P when compared to R:

$$F_\beta = \frac{(1 + \beta^2) \cdot PR}{\beta^2 \cdot P + R} \tag{A.7}$$

Further specific mobile applications

B

B.1 Visual bookmark

B.1.1 Introduction

In this Section, we present a prototype application for object based web search and navigation on a mobile phone. Instead of a traditional text-based query which is quite inconvenient given the constraints of mobile phones, the query is simply formulated by capturing a photo of the object of interest. The search application will then use that photo to find similar instances of that object in a database, and provides users with associated information, such as tags, descriptions, or links to web pages.

B.1.2 Related Work

Most of the mobile applications for content based image search and retrieval rely on visual models which are created from one or more training images and matched against a set of test images to find image or object duplicates.

Beside commercial applications mentioned in Section 6.1, a few prototype applications have also been developed in order to study different aspects of mobile image search and retrieval.

In Yeh *et al.* [2005], contour based matching is applied for object duplicate detection in mobile phones. Contour matching is treated as a graph matching problem, and uses the Earth Mover's Distance (EMD) as a metric of similarity, which approximates the minimum cost that is necessary to transform one weighted point set into another. For this application well segmented contours are needed, therefore they ask user to take two pictures from the same position, one with the object and second with removing the target object. In this case they can extract sufficient object contours. For our algorithm a simple image from the object is enough even with high variety of

view point changes, due to the usage of local features.

Real time object search is analyzed in [Chen *et al.*, 2009], where detected object is tracked and object localization is performed every second within a video stream. Image compression is applied to reduce the necessary bandwidth, and local SURF with geometric consistency check is applied for localization on the server side. However this algorithm just localizes an object and does not search them in large databases.

In [Tsai *et al.*, 2009], feature extraction is performed on mobile phone and feature compression applied, to reduce the amount of data to be sent to a server. The location information is converted into a location histogram, and a context-based arithmetic coding with location refinement method is then proposed to code the histogram.

B.1.3 Proposed application

The goal of this application is to provide an intuitive interactive query interface for mobile information retrieval based on photos captured by a mobile phone. After recognizing the object of interest, the user automatically gets access to information or services associated with it.

The application can be used in various scenarios depending on whether the information or service is related to the object or not. In the former case, a user may be interested in an object. The object is captured with the mobile phone camera and a short summary about the object along with links to information or services from public databases is received. For example, one could capture a movie poster to get information about the actors, expert reviews or purchase of a ticket; image from EPFL logo can show the university home page or a university forum, etc. In the latter case, a user may want to use objects as shortcuts to frequently used information or services. Therefore, he/she creates a personal database of objects with not necessarily related information or services and accesses them by capturing the corresponding object. For example, one could capture the refrigerator to access and online grocery shop or capture the radio or television to receive today's program guide; showing a person's ID card to this application will automatically add this person as a Facebook friend.

Each object has several descriptions and this information is shared by users, therefore this application can be used as social games too. Virtual treasure hunting is one of these games. Similar to the popular facebook application, Treasure Isle by Zynga [Zynga, 2009b], you are collecting items by hunting treasure. However in our case the application is running in real life and "digging" is done by taking pictures from objects. Another game for generating data is proposed, based on ESP Game [von Ahn and Dabbish, 2004], users can create a photo from an object and describe it with an URL or simple word. If the object has already described in the same way, then both user get scores.

The application (Figure B.1) consists of two parts, the *mobile side* which deals with the user interface and the *server side* which performs the object duplicate detection, and other necessary processing. These two parts communicate over a TCP/IP network. The mobile side sends a query and the picture of the object to the server using an XML data structure. On the server side, the object duplicate detection is applied and the result of query is sent back to the mobile and presented in the user interface.

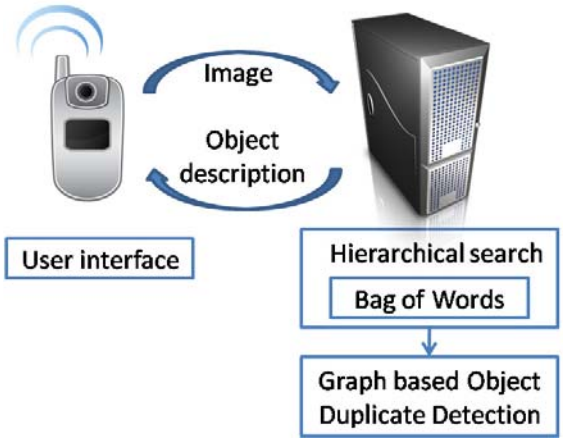


Figure B.1 — System architecture of visual bookmark application.

The application works in two different phases. In the *training phase*, the user takes one or more photos of an object and links it to some description or service. The information is sent to the server, where the photos are analyzed and an object model is created. In the *detection phase*, a user takes a photo of an object which is matched against the objects in the database and receives the information or services associated to the recognized object.

Mobile side

The mobile side of the application contains the user interface and the communication part with the server. The user interface contains a main screen with buttons for object training, object detection and clustering (Figure B.2, 1st mobile screen) . A working dialog is displayed during the detection and the training (Figure B.2, 2nd mobile screen), which could take up to a few seconds. Additional description query dialog is shown when a user trains an object (Figure B.2, 5th mobile screen), and a notification is displayed when the task has completed successfully (Figure B.2, 6th mobile screen). For object detection, the results are descriptions of the detected object, which can contain URL links as shown in the third screen shot (Figure B.2, 3rd mobile screen). Clicking on a simple keyword, the application shows the google search result and clicking on a URL will access the corresponding web page (Figure B.2, 4th mobile screen). If the user selects clustering, a command is send to the server, to recalculate the feature vocabulary, described in B.1.3.

Server side

The server side of the application performs the object training and detection based on the photos provided by the mobile side. Since the database may contain a large number of objects, the object detection algorithm needs to be scalable in terms of reliability and complexity. For an improved efficiency, the object training and detection are divided into two different steps. Images, which contain the original object, are selected by the algorithm using image similarity method and then a more complex object duplicate detection algorithm is applied on these images.

Figure B.2 — Screen shots from the mobile application in reading order: main, detection, results of detection, resulted web page, object description in training and notification screen shot.

In the *training phase*, the system extracts local features and learns an object model from the photo taken by the user. The description of the object is attached to the model and both are stored in a database. Sparse local features are used to resolve the object localization problem efficiently. First, regions of interest are extracted using the Hessian affine detector [Mikolajczyk and Schmid, 2002] and each of these regions are described using SIFT features [Lowe, 2004]. These features are robust to arbitrary changes in viewpoints. K-means clustering is applied to the features of the whole database, to create a feature vocabulary. This process is very slow, therefore the clustering algorithm is applied offline. In this case all the object models are recreated as well. For a fast image matching, Bag Of Words model [Fei-Fei and Perona, 2005] is created for each of the objects based on the local features and creating a histogram over the used features vocabulary. For a more precise object duplicate detection a spatial graph model of the object which considers the scale, orientation, position and neighborhood of features, is built as shown in Section 3.3.

In *detection phase*, the system recognizes a captured object based on the learnt object models of the whole database. As a result, the scores and bounding boxes of the objects are calculated and the associated descriptions are sent to the mobile application. During the image matching step, several candidate images are selected according to their similarity to the previously trained Bag of Words model [Fei-Fei and Perona, 2005] which is shown to be robust to arbitrary changes. Since it compares histograms of local features using hierarchical clustering, it is very efficient, fast and requires only a small amount of memory. In the second step, the candidate images are compared to the query object using a graph based object duplicate detection method, which provides more accurate results, as shown in Section 3.3. This algorithm has been shown to be more robust and accurate than the "Bag of Words" model, as it also considers the spatial arrangement of local features. However, because it uses local features for matching, it is slower than the "Bag of Words" method.

B.1.4 Results and analysis

The goal of the conducted experiments is to assess the performance of the proposed object detection method which serves as the backbone of the mobile information retrieval application.

For the evaluation, the object database discussed in Section 3.4.1 was used. It consists of 850 images of individual objects with a large variety of view points and sizes. It contains 15 object classes and each class contain at least three different object instances.

We have compared the graph based object duplicate detection algorithm, proposed in Section 3.3 against the Bag of Words (BoW) algorithm [Fei-Fei and Perona, 2005] and the object duplicate detection algorithm proposed in [Lowe, 2004]. While all of these methods rely on SIFT features, they differ in the way the spatial arrangement of the local features is considered. The BoW model does not consider any spatial information and combines the local features into a histogram. Both the model in [Lowe, 2004] and the graph based model in Section 3.3, consider spatial information by applying a General Hough transform. However, the latter improves the reliability of the feature correspondences due to the graph matching.

As it is shown in Section 3.5.2, the BoW method performs worst, as it does not consider any spatial information. The method in [Lowe, 2004] shows improved performance by considering this

information. The method in Section 3.3 provides the best performance among them thanks to the graph matching. However "Bag of Words" method is more efficient computationally, therefore with these results we validate that our algorithm, which contains two separated tasks, is good for accuracy and computation efficiency.

BoW method is a global feature, therefore nearest neighbor search is computationally very efficient using this algorithm. BoW method is used for pre-selection in object duplicate detection and the method in [Vajda *et al.*, 2009b] for robust and accurate object validation.

B.1.5 Conclusion

Image and video retrieval systems are becoming increasingly popular and important in many applications. Smart mobiles are the new target of this growing area. In this paper, we have analyzed and described a new web navigation application for mobile phones.

Different techniques to speed up the process of indexing and retrieval are presented in this paper, such as two levels of object duplicate detection. In the first level, a fast image matching technique is used to select a subset of the images from the dataset which are likely to contain the object of interest. In the second level a more complex object duplicate detection technique is applied to improve the accuracy. Further improvement can consider GPS location or recognition of text to improve the detection and speed of the algorithm. Also mobile based feature extraction can speed up the communication time with the server.

This application has a great capacity for gaming, education and personal usage. For example, one could capture a cover page of newspaper to access it online or capture the radio or television to receive today's programme guide.

B.2 Chess recognition

B.2.1 Introduction

In 20th century, many people have been fascinated by the idea of constructing a chess-playing machine. In 1951 the first program was developed on paper, capable of playing a full game of chess, making use of simplified rules because of the limited available computing power. In 1957 the first fully functional chess program was developed by Alex Bernstein. The big turning point was 1997 when IBM's Deep Blue defeated chess master Garry Kasparov. From this year, computer is officially better in playing chess games than humans.

The goal of this mobile application is to provide a tool able to advise a chess player upon the next move to make, as shown in Figure B.3. The user provides two pictures. The first one shows the chessboard and the figures at the beginning of the game. It will be used as a reference in the figures recognition method. The second picture shows the chessboard at a random state during the game. Both cases can have different points of view, however the closest field should be A1.

Object duplicate detection can find several applications such as multimedia objects automatic tagging, surveillance and security videos or image-based database search. While most applications are probably just *content-based* applications, some others, such as those presented in this Section,

Figure B.3 — Detecting chess board and recognizing chess figures.

are also *context-based* applications. Content-based recognition techniques use the actual content (not tags or keywords, but the actual visual content) to recognize something, such as an image in a database. However, in most applications, there is few (or even no) knowledge of the *context*. In other words, most content-based applications do not necessarily have any information on the number of instances of a given class of objects that they are trying to detect, nor on their possible locations, sizes or orientations, nor even on their presence (or not) in the image, as opposed to context-based object duplicate detection applications which base their methods on one or several of this a priori knowledge.

In this section, an applications using context-based object duplicate detection will be presented. Nowadays, the market of computer vision-based applications and business is booming, particularly in the gaming field. The application proposed in this section deal with real-life situations. Therefore and to ensure a better portability, this application is mobile phone-intended.

B.2.2 Related work

Due to their very particular structure, chessboards are relatively easy objects to detect in an image. For instance, they are often used for the calibration of stereo cameras. However, detecting and locating a populated chessboard, i.e. a chessboard with chess figures on it, the task gets harder. It has however been the object of several previous works. On the other hand, the task of recognizing chess figures is even much more challenging and has therefore very seldom been tackled.

In [Neufeld and Hall, 2010], the chessboard detection process uses the Hough transform and a perspective transform to undistort the image. The detected lines are then used to build a set of potential quads. Each quad is tested to match a chessboard pattern. The quad with the highest probability (hopefully the whole chessboard) is chosen. The testing procedure is performed by using figure-shaped masks in order to keep just the squares, which are individually tested for chessboard pattern by averaging their global intensity value. Though the chessboard detection has shown promising results, the challenge of detecting the figures (or recognizing them) has not been tackled.

The approach proposed in [Tam *et al.*, 2008] is also based on line detection. To discard the bad lines and verify the alignment, the diagonal information is considered by ignoring the intersections that do not fall onto a likely chessboard diagonal. Experimental results have shown to be good for the detection of a populated chessboard with the proposed technique. The authors have compared it with Harris corner detection-based technique, which proved to be significantly less good. However, chess figures have again not been dealt with.

Some autonomous chess playing robots were built in the last decade. The main advantage of these setup that robot can analyze the whole game with every movements and follow the states of the figures, which simplifies the figure recognition task [Groen *et al.*, 1992]. However in our application we decided to not record the whole game with mobile phone, because it is maybe unconformable for the user. Therefore an advanced figure recognition algorithm is necessary.

As explained above, the detection of a populated chessboard has already been the object of several works. The best results seem to have been achieved with line detection-based methods rather than with corner detection-based methods, since the occlusion of corners by chess figures

make the extraction of the grid really challenging.

On the other hand, the challenge of detecting and, above all, recognizing chess figure using single image is open new researches in chess application.

B.2.3 Proposed algorithm

System architecture of the proposed algorithm can be found in Figure B.4.

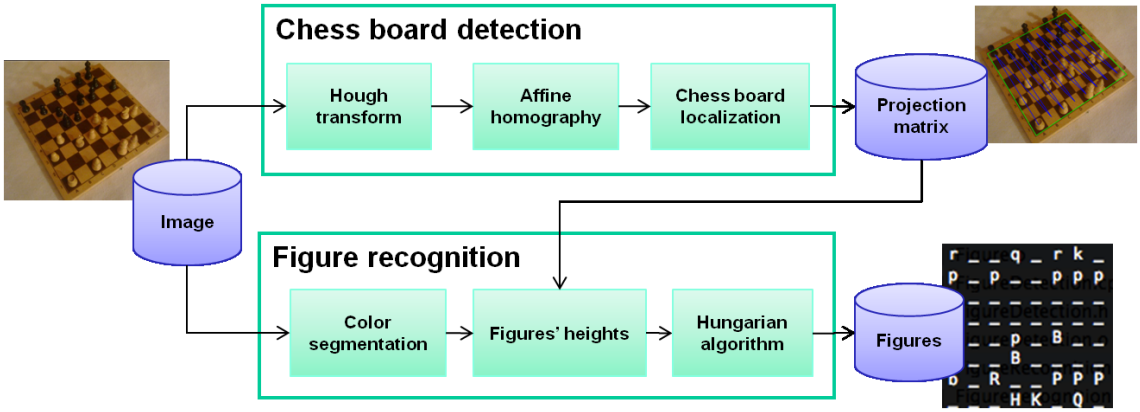


Figure B.4 — Proposed algorithm flowchart for detecting chess board and recognizing chess figures.

First chess board detection and localization is applied on the input image given from a chess board, resulting in projection matrix which represents the 3D position of the board. Evaluating this information with the input, image figure recognition is applied. Using height information of the chess figures and position of the chessboard can lead to better results than a general object recognition of chess figures. More explanation is discussed in the following Sections.

B.2.4 Chessboard detection and localization

Hough transformation

The first step in detecting the chessboard is the detection of the lines in the input image. To do so, the image is converted to gray-scale and *Canny edge detection* [Canny, 1986] is performed on it. Then *Hough transform* is used as shown in Section 2.7.2 for line detection as shown in blue color in Figure B.5 (a).

Due to the presence of figures on the chessboard, some lines may be missed, while some false lines may be detected. In order to discard those wrong lines, the two *principal directions* are computed by projected histograms. All the lines that are not along one of the two principal directions will be discarded. The second step in the lines discarding process is to eliminate double lines, that lines are practically superposed on others but that are actually the same line. To do so, only one line is kept when dealing with one of those cases:

- Both lines are along the same direction and are crossing in the image.

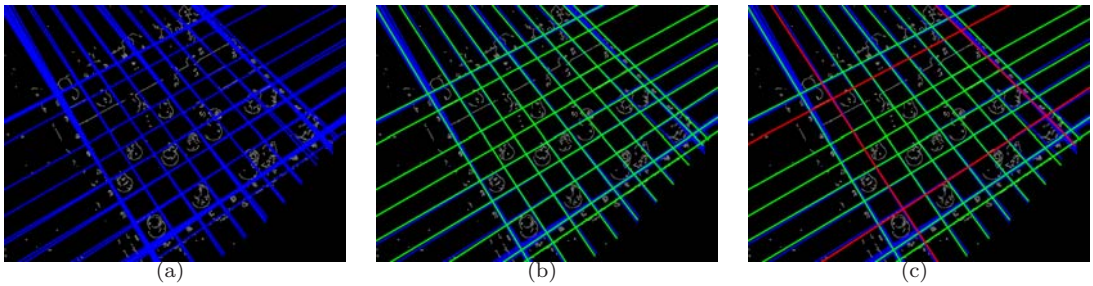
- Both lines are along the same direction and are very close.

At the end of this process, only real and legitimate lines will hopefully remain, as illustrated in Figure B.5 (b) with green lines.

Affine homography

Once the lines have been detected and bad lines have been discarded, a square part of the chess board (quad) is built from them. Then the quad is mapped on the screen plane.

To undistort the quad and map it onto the screen plane, an homography is computed taking as object points the four corners of the quad as described in Section 2.7.3. Since the homography matrix coefficients are estimated with those coordinates, the larger the quad, the better it is. There is quite a high probability that at least one of those lines actually belongs to the borders of the chessboard and not to the outermost line of it. Therefore, the second outermost lines are considered to build the quad. Similar way, four lines are detected as border of the quad as shown in Figure B.5 (c) with red color.



(a) (b) (c)

Figure B.5 — Steps are shown to obtain the inner quad. Detecting Hough lines (a), discarding bad lines (b), getting inner quad (c) are shown.

The four corners of the quad are then used as object points and mapped onto a square image of a known size through homography.

Chess board localization

In order to determine, how many lines and columns the quad is contains, all possible chessboard pattern are created, as illustrated in Figure B.6, and convolved with the image, resulting scores for each pattern. The pattern that produces the highest score is chosen the one that is similar to the quad's pattern. The process is of course repeated with the negative version of the binary image, since its pattern can simply be inverted.

Knowing the number of lines and columns of the quad gives information of all the possible locations of the chessboard in the original image. If the quad has the size of one square, there would be 64 possible locations. The larger the quad, the smaller the number of possible locations of the chessboard as shown in Figure B.7 (a).

Canny edge detection is applied on the affine-transformed input image [Canny, 1986]. The edge image is slightly dilated, using a disk-shaped structuring element with a 5 pixels diameter, in

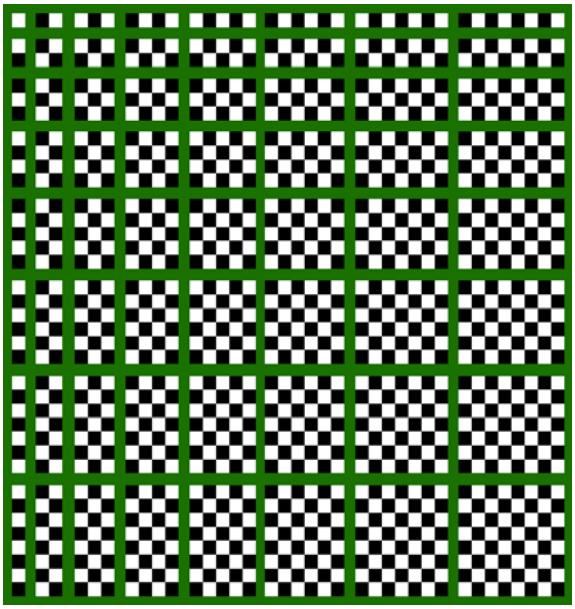
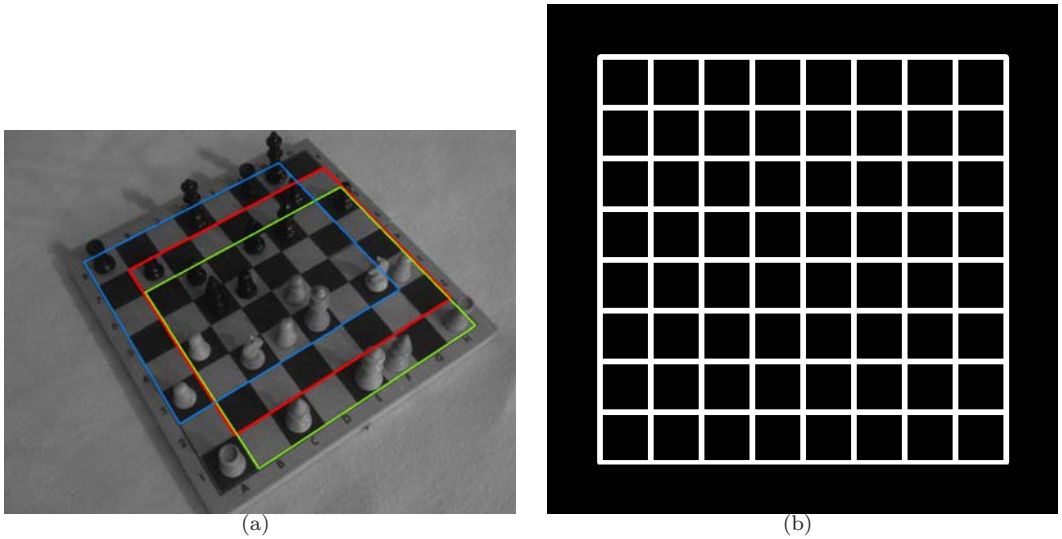


Figure B.6 — All possible chess patterns of the quad is shown and separated by green lines.



(a) (b)

Figure B.7 — Location of the quad on the chessboard. Three (of the six) possible locations of the quad (a), synthetic chessboard (b) are shown.

order to make sure that it matches at least partially the real alignments. In parallel, a synthetic chessboard edge image is generated, as illustrated in Figure B.7 (b). Convolution is performed with all the possible locations of the chessboards. The highest score is achieved when the position of the synthetic chessboard is accurately matched with the real chessboard. By mapping it back to its distorted view, the location of the chessboard in the image plane is now known.

B.2.5 Figures recognition

Color segmentation

The detection of the chess figures is performed through *color segmentation*. The HSV and YUV color spaces has been considered, but both have shown to be inadequate. However, the RGB color space has proved to be quite ideal for the segmentation of the black figures. Simple color distance is calculated by channel and thresholded.

The detection of the figures through color segmentation is not perfect. Shady or bright parts of the figures are missing, while some parasite pixels are also detected. In order to refine those images, morphological opening is applied then holes in the figures are filled and small blobs are removed.

In order to *localize the figures*, a synthetic image of a 8×8 repeated disk pattern is created as shown in Figure B.8 (a). The refined images obtained above are then mapped onto the screen plane and multiplied with this repeated disk pattern, resulting in Figure B.8 (b). High figures such as kings and queens can massively occlude another square; However, they will seldom occlude more than 60% of the zone corresponding to the base of a figure in the occluded square. Therefore, the blob removal algorithm used, with the discriminating criterion of the area of the blobs. Those that are smaller than 75% of the original area of a disk are discarded, as shown in Figure B.8 (c). As a result, most occlusion problems can be solved. This process being performed for black and white figures separately, the color of the figures that populate the chessboard is also known.

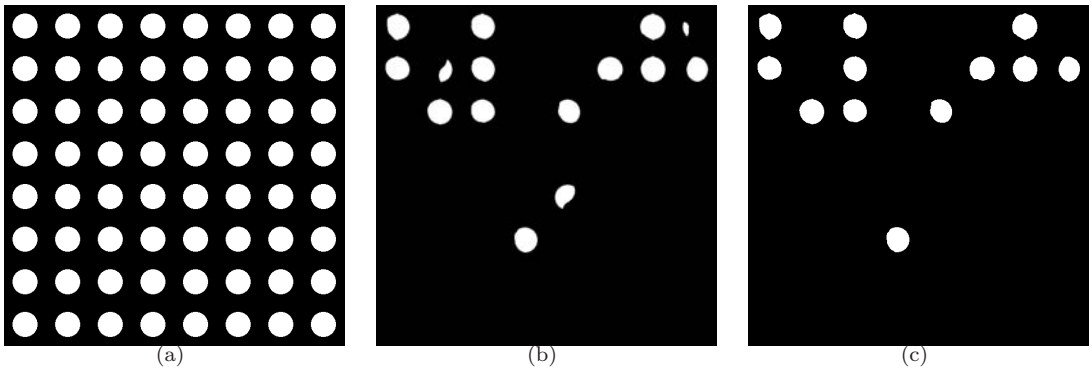
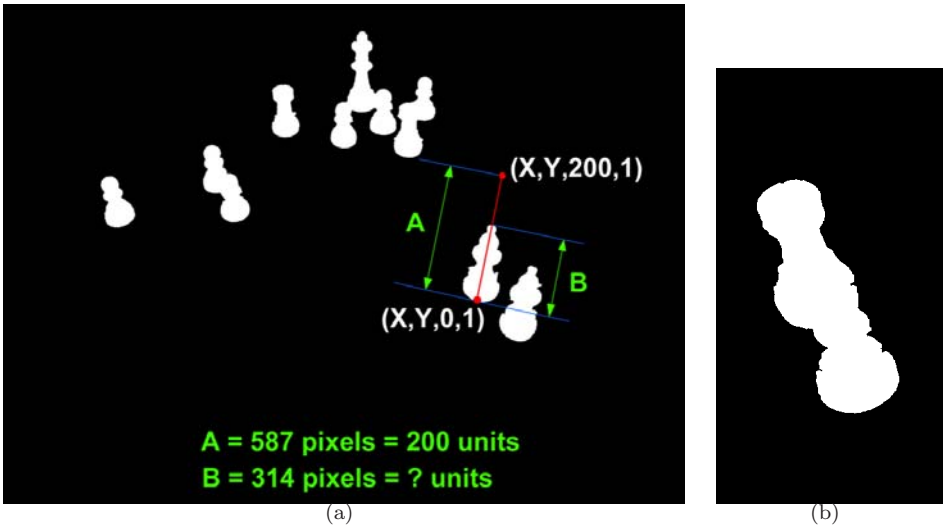


Figure B.8 — Figures localization. Repeated disk pattern (a), Multiplication with the black figures (b), Removal of small blobs (c) are shown.

Figures' heights

Chess figures recognition is one of the most challenging task of chess recognition. The approach considered here is context-based, since chess figures are known to have different heights (king, queen, bishop, knight, rook, pawn, from highest to smallest). The distortion added by the perspective view of the chessboard makes them even more meaningless. However this point can be coped with by computing the projection matrix as seen in Section 2.7.3. With this matrix, it is possible to map any real-world point, onto the image plane. Therefore, the technique proposed here is to consider a high point located above all detected and localized figures in the 3D real-world coordinate system. The dimensions are expressed in real-world-related units (unified units), which was calculated by taking a calibration image from the fully populated chessboard. These points are located on the lines that are orthogonal to the chessboard and that go through the bases of the figures as illustrated in Figure B.9 (a). Then the top of each figures are computed.

Hence, using the projection matrix, the heights of figures located far or close to the camera and pointing towards different directions because of perspective distortion can all be measured and compared in real-world-related units. In the reference image, which shows the chessboard state at the beginning of the game, the location of the pieces are known. Therefore, the reference height is calculated for each type of figure by simply selecting the squares where the figures are located.



(a) (b)

Figure B.9 — The challenge of figures height computation. Height computation method (a), figure occlusion (b) are shown.

Occlusion is the major issue when computing the height. It might produce false recognitions. For instance, the pawn located in front of the rook in Figure B.9 (b) might produce a height comparable to that of a king. However some extreme cases can be dealt with. For instance, if the computed height is larger by at least 20% of the reference height of a king (which is the highest chess figure), we can assume that there is an occlusion problem with the current figure. Unfortunately, there is no way to know what kind of figure it is. Therefore, one simply decides that

it is pawn, which is the most frequent figure on a chessboard. By doing this, one avoids not only a false recognition of the current figure, but also the suboptimal choice that would be made later when one deals with the king. Indeed, if the king is detected as in Figure B.9 (b), it cannot be located elsewhere since there is at most one king on the chessboard. This is precisely the challenge of the best combination computation, as detailed hereunder.

Hungarian algorithm

Now that the heights of the figures are known for both the reference image and the tested image, one cannot simply assign the closest reference height to each tested figure. Indeed, the context needs to be considered. The maximum number of each type of figures are known, for instance there are at most two bishops for each color. To take this context into account, a *multiple hypothesis* approach is considered. For each figure:

- Compute the difference of height with all 6 references.
- Sort them from smallest to biggest.
- Assign a vector of increasing cost which is quadratically proportional to the differences of height with each reference figure.

To minimize the global cost of pair matching problem, *Hungarian algorithm* is used as known that it gives optimal solution in cubic computation complexity ($O(n^3)$ where n is the number of figures)[Frank, 2004].

The positions, colors and types of all figures are saved in an output text file as seen in Figure B.10 and can be used with an online program such as GNU Chess* to compute the best next move and advice the user.

```
CHESS GAME HELPER
=================

K/k = King, Q/q = Queen, R/r = Rook, B/b = Bishop, H/h = Horse, P/p = Pawn.
White = capital letters. Black = small letters.

h _ _ q _ h k _
p _ _ _ _ _ _ _
_ _ r _ _ _ Q _
_ _ _ _ _ p _ _
_ _ _ r _ _ _ _
_ _ _ B _ _ _ _
b _ R _ _ P P P
_ _ _ B _ _ K _
```

Figure B.10 — Example of output.

*http://www.gnu.org/software/chess.

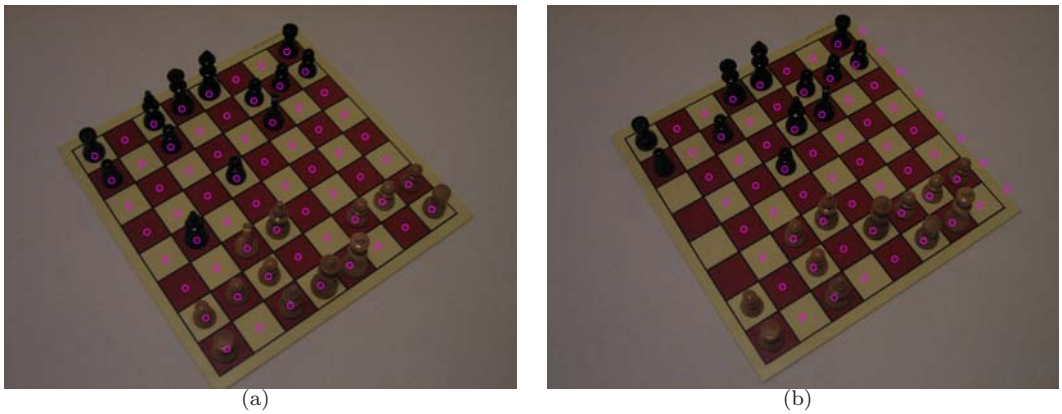
B.2.6 Database

The database created from a chessboard during the replayed chess match against Judit Polgár Hungarian chess grandmaster and Garry Kasparov world chess champion. In this game Judit defeated Kasparov, this game was historic as not only the first time in chess history a female player beat the world's best player in competitive play, it was the first time in any sport that the No. 1 ranked male player has lost to the No. 1 ranked female player. The database, contains 43 images as shown some of them in Figure B.11. The following assumptions have been made based on those images:

- There is no or little overlap between figures.
- The chessboard can be seen entirely.
- The $A1$ square is the closest to the user, which defines the chessboard's orientation.

B.2.7 Results and analysis

The entire chessboard detection and localization process can be considered as a whole and evaluated by simply considering the tested images where the chessboard was correctly detected and localized and those where it was not. As for comparison, the OpenCV library provides a method called *cvFindChessboardCorners*, which is usually used in order to calibrate a camera using a chessboard. It is based on the Harris corner detector. To successfully extract the chessboard, all the corners of the chessboard need to be detected. Therefore, such a method is not designed for populated chessboards detection, where there is almost a null probability for all corners to be visible at the same time because of the occlusion produced by the figures. For the our proposed method, one localization was missed, where the chessboard was detected but its location was shifted because of the failure of pattern matching, as shown in Figure B.11 (b). Therefore the detection rate for the proposed algorithm is 98%, meanwhile the detection rate of the OpenCV function is 0%.

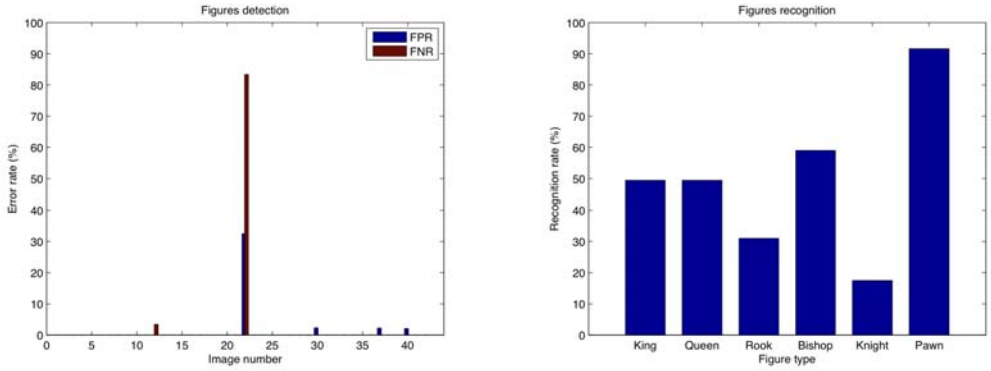


(a) (b)

Figure B.11 — Illustrated chessboard localization. Good chessboard localization (a) and missed chessboard localization (b) are shown.

Figure B.12 (a) shows the performance of the figures detection process in terms of false positive rate (FPR), false negative rate (FNR) discussed in Appendix A. The error rate of Image 22 is unusually high compared to the other cases. The reason for this result is that chess board detection algorithm failed in this image.

Figure B.12 (b) shows the recognition rate of the global figures recognition for each type of figure. It shows the number of times a figure was well-recognized with respect to the number of occurrences of this figure in the images. Pawns can be detected easily, due its size, however detection of knights and rocks are mixed by the algorithm, thanks to their similar height.



(a) (b)

Figure B.12 — FPR and FNR of figures detection are shown for each image in (a). Type-wise figures recognition rate is shown in (b).

The global figures detection and recognition process can be evaluated in a square-wise manner. That is, one evaluates the number of chessboard squares for which the content has been successfully recognized. In other words, a match, as illustrated in Figure B.13 (a), happens when an empty square is recognized as empty or when a populated square is recognized as populated and its figure is recognized correctly. A mismatch, as illustrated in Figure B.13 (b), appears when an empty square is recognized as populated or when a populated square is recognized as empty or when a populated square is recognized as populated but its figure is recognized incorrectly. Finally, the overall square-wise results for average recognition rate is 80%.

B.2.8 Conclusion

Mobile application for chess recognition has been designed on context-based object duplicate detection fundament, that is: knowing the *context* provides *a priori* knowledge on the *content* of the targets to be detected and recognized. For instance, a massive use of context-based knowledge has been done in this application, when computing the most probable set of possible chess figures, by taking into account the rules of chess.

Results are very promising. The task of detecting a populated chessboard has quite often been the object of research. However, the challenge of detecting and, above all, recognizing chess pieces has seldom been tackled, due to its higher complexity.

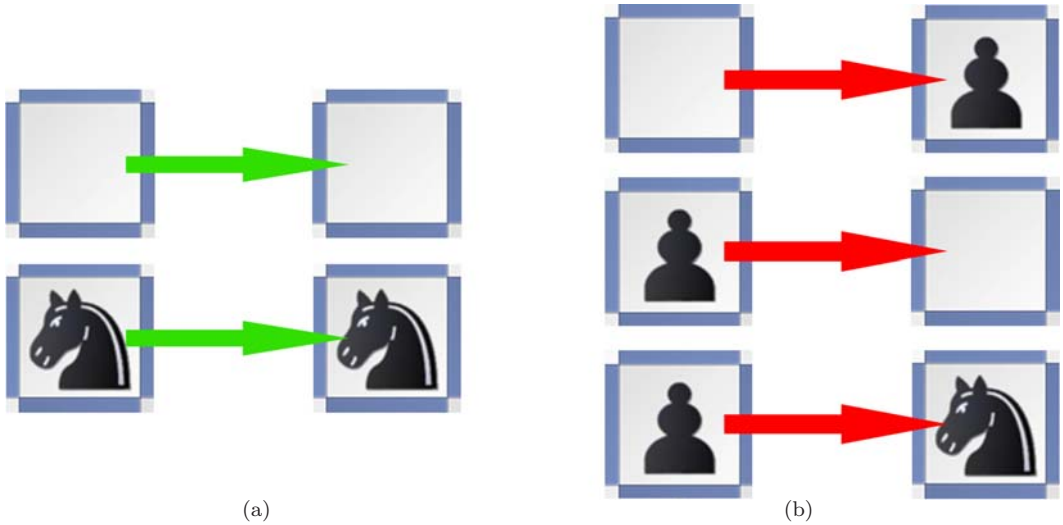


Figure B.13 — Square-wise matches and mismatches. Possible matches (a) and possible mismatches (b) are shown.

The proposed method for chessboard detection and localization has shown to be very robust, with a 98% accuracy. Also, the repeated disk-pattern matching approach used for figures detection has proven to be extremely efficient and robust against occlusion.

The main challenge of this whole project was clearly the chess figures recognition part. The proposed approach, based on heights comparisons through a perspective transformation, is both promising and yet limited. The angle of view from which the pictures are taken influences the perspective distortion but also the occlusion between chess figures. Nevertheless, the results obtained through this approach are really good, when considering the complexity of the task.

B.3 Flower recognition

B.3.1 Introduction

In this Section, mobile application is presented for flower recognition. The scenario is the following; Someone hiking in the Swiss mountains who finds a beautiful flower. He/she would like to know more about that flower. What's its name? Is it rare? Is it protected? etc. By simply taking a picture from the flower with a mobile phone, he or she could get all these information through an automatic flower recognition application.

In this section, the elaboration of such application has been aimed. The recognition of flowers from photographs implies several steps, starting with the localization of the flower in the image, followed by identifying and extracting the specific characteristics of this flower, and finally finding the best match for this flower in the database. The solution proposed in this section includes the following elements; specific research on plants and on existing method, a segmentation algorithm based on user's inputs, implementation of several visual features suitable for flowers differentiation. A demonstration tool was implemented in Samsung S mobile phone using Android 2.3.3 operating

system. Other applications could include educational purposes and nature preservation programs.

B.3.2 Related work

Several research conducted in flower recognition and general feature extraction.

It is suggested to use domain knowledge of flower colors to index the images, in [Das *et al.*, 1999]. In this context, they developed an iterative segmentation algorithm to isolate the flower from the background. They used only color names and their relative proportions within the flower region as features which is not sufficient for full recognition. Saitoh and Kaneko [Saitoh and Kaneko, 2000] have proposed a method that uses two input images, one of the flower and one of the leaf. In order to do so, the user should place a black cloth behind the flower. The background separation uses k-means clustering method on color space. They considered color and shape information for both the flower and the leaf. Interactive methods such as CAVIAR (Computed Assisted InterActive Recognition) [Nagy and Zou, 2002; Zou and Nagy, 2004] have been developed to exploit the human perceptibility. A rose-curve is generated for the test flower and the top three candidates are proposed to the user who can then either select the right flower or modify the rose-curve to get a new set of propositions. This is done iteratively until the right flower is found. In [Saitoh *et al.*, 2004], an automatic method is developed under the assumption that the flower is focused while the background is unfocused. It then uses a Normalized Cost (NC) [Saitoh *et al.*, 2003] method, which needs a manual entry point on the boundary. They overcame this drawback by implementing an automatic method that minimizes the NC among a set of smartly chosen entry points. The resulting segmentation is then shown to the user who can, in case of failure, add a new entry point. Four shape features (number of petals, central moment, roundness and width/height ratio) and six color features taken from HS color space (x and y coordinates of the largest/second largest color cell and ratio of the largest/second largest color cell) were used yielding a recognition rate of 90%.

A region-based color image retrieval using geometric properties is presented in [Hsieh and Fan, 2000]. Briefly, they used a region-growing technique to form color regions, then spatial relational graph and Fourier description coefficients are computed for each region. Brandt et al. [Brandt *et al.*, 2000] proposed a technique, applied on non-segmented objects, using edge histograms and Fourier-transform-based features computed for an edge image in Cartesian and polar coordinate planes. Based on dominant color in the foreground image, the method proposed in [Krishnan *et al.*, 2007] used a color look-up table to divide the color space into smaller categories then Euclidean distance was used for matching. Three features, color, texture and shape information are combined together in [Hiremath and Pujari, 2007] by first partitioning the image into non overlapping tiles of equal size. Then color moments and moments on Gabor filter responses of these tiles locally describe the color and the texture respectively. Most similar highest priority principle is applied for matching.

Some work has been done in the domain of leaves recognition, Im et al. [Im *et al.*, 1998] used a hierarchical polygon approximation representation to recognize the Acer family variety. Zhenjiang et al. [Zhenjiang *et al.*, 2006] used size, shape and color of petal and leaf like many others but added an object-oriented pattern recognition (OOPR) approach which mathematically deals with

how to use all different features rationally in the recognition scheme.

Finally a mobile-based flower recognition system has been implemented [Kim *et al.*, 2009]. It uses Difference Image Entropy (DIE) and contour features of the flower. The user has to draw the exact contour of the flower himself and then the two images, the drawn and the original image, are sent to the server where the DIE is processed.

B.3.3 Proposed algorithm

The flower recognition scenario is the following; The user takes a picture from a flower and he/she indicates the flower region using the touch screen of the mobile phone. This information is sent to the server. Segmentation is done on the server which results in a binary mask of the flower. Specific features are extracted from the image of the flower, using the mask and then it is compared to the database. Finally, information from the detected flower is provided to the user on the mobile phone. The system architecture is illustrated in Figure B.14.

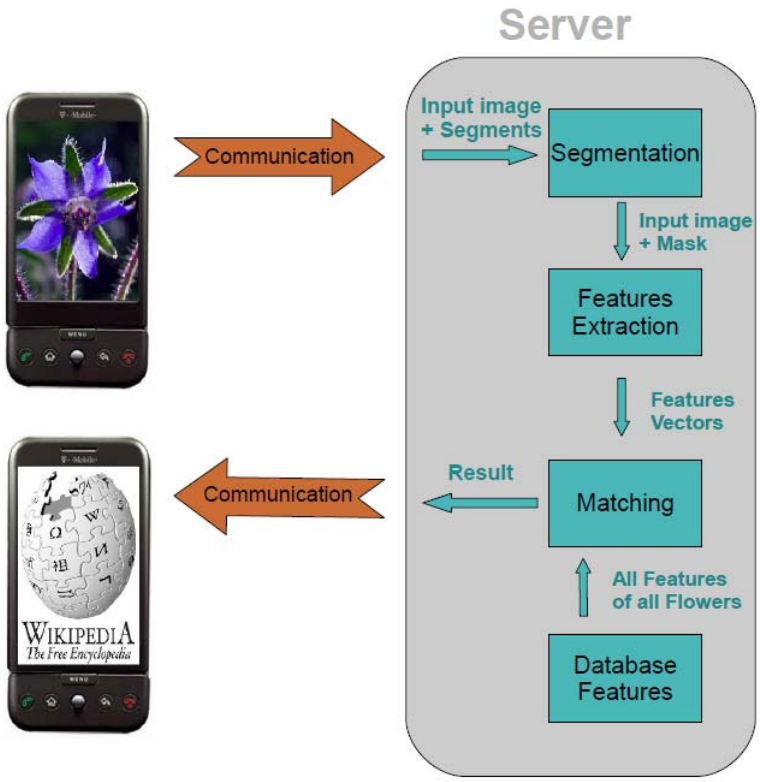
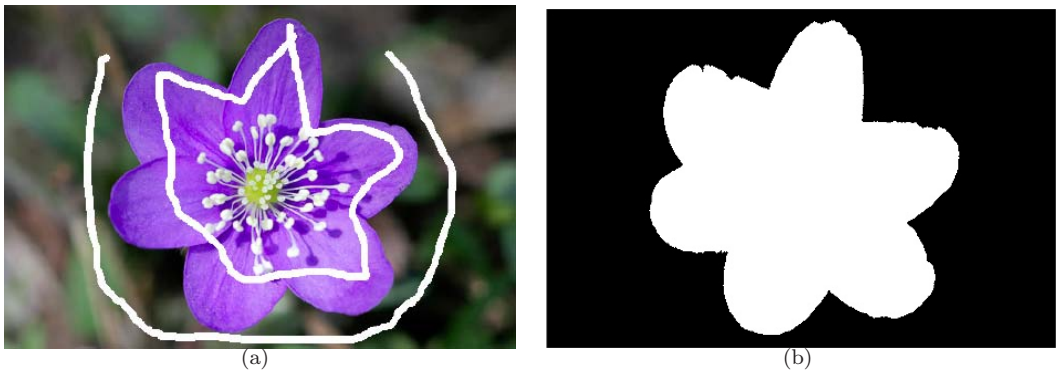


Figure B.14 — System architecture of the proposed flower recognition algorithm.

Segmentation

The main purpose of segmentation is to determine regions in the image that belong together, in this application only two regions are needed; one defining the flower and another corresponding to the background. It is important, since the background can mislead the flower recognition algorithm.

Interactive watershed algorithm [Najman and Schmitt, 1994] is applied for flower segmentation. This algorithm is based on the gradient of the image which is viewed as a topographic relief where high gradient values correspond to mountains while low values correspond to valleys. The algorithm segment the image based on the attitude of the topographic relief. This algorithm sometimes yield in over-segmentation, due to the presence of too many local minima. A variation of this algorithm called marker-controlled watershed takes over this drawback by allowing the user to choose the locations of the water sources. These locations are called markers and they basically say which regions have to belong together in the final result. This functionality is very useful as it can be specified which part of the image is of interest and which is not. This markers facility can be adopted in an iterative way i.e. the result of segmentation can be shown to the user and if the result does not satisfy him, he can add new markers, then the result is improved by the algorithm. In our case the markers are provided by the user using the touch screen of the mobile phone as shown in Figure B.15.



(a) (b)

Figure B.15 — Segmentation by user interaction.

Features

This section lists all the 18 features extracted for the recognition. The extracted features have been grouped by different categories: colors, contour of the flower, texture and finally specific features extracted from petals. This organization is illustrated in Figure B.16.

In our research we used FELib, Feature Extraction Library [Zhu, 2008], which provides tools to extract features for Color Histogram, Color Moment, Edge Histogram, Gabor wavelet transform, Local Binary Pattern and Gist features. More details about these features can be found in Section 2.4.

There are many different ways of extracting the color information from an image, some of the basic ones are color histograms and color moments. They both can be evaluated from different color spaces, so the main ones are listed below.

- **RGB**: color space comes from an additive model in which the three primitive colors red/green/blue are added together to reproduce the all range of colors. Similar in a way to the HVS it is widely used and present in all CRT monitors.

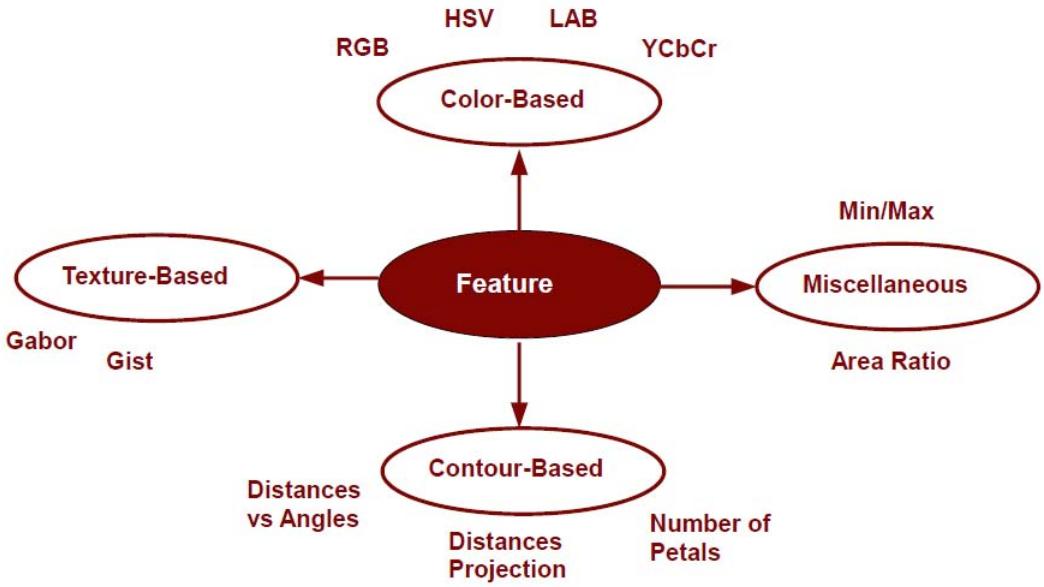


Figure B.16 — Features which are extracted for flower recognition.

- **HSV**: stands for Hue, Saturation, Value, it is a cylindrical-coordinate representation and it is known for its intuitiveness.
- **LAB**: Lab is a color-opponent space, L is for lightness and a and b for the color-opponent dimensions, based on non linearly compressed CIE XYZ color space coordinates. Because of its opponent process that better match how the information from the cones are processed by the HVS, this color-space is known to be perceptually uniform.
- **YCrCb**: Luma, blue difference, red difference. YCrCb is not an absolute color space, it is more a way of encoding RGB information. It is more efficient than RGB for storage and communication.

The feature vector for color histogram is extracted, whose length equals 768 (3·256, 256 byte for each color channel). The option of inputting a mask for the histogram calculation is offered and has plainly been used in order to discard the background. Different color spaces were tested namely; RGB, HSV, LAB and YCrCb.

For extraction of color moments, the image is partitioned into 3x3 grids and for each grid three values are extracted: the color mean, the color variance and the color skewness. This finally yields in a 81-dimensional grid color moment vector [Zhu, 2008].

The most straightforward contour based feature is to go through the contour for all angles for a given step size and the distances from the contour point and the center of gravity are then computed as shown in Figure B.17 (a). This vector is called distance vector from now. Determination of the start angle of this feature is defined by the maximum distance of the contour. From this distances

vector the maximum and the minimum distances is detected from the center of the flower, then by taking their ratio the min-max feature is obtained. Histogram of distances is produce the projection feature as shown in Figure B.17 (b).

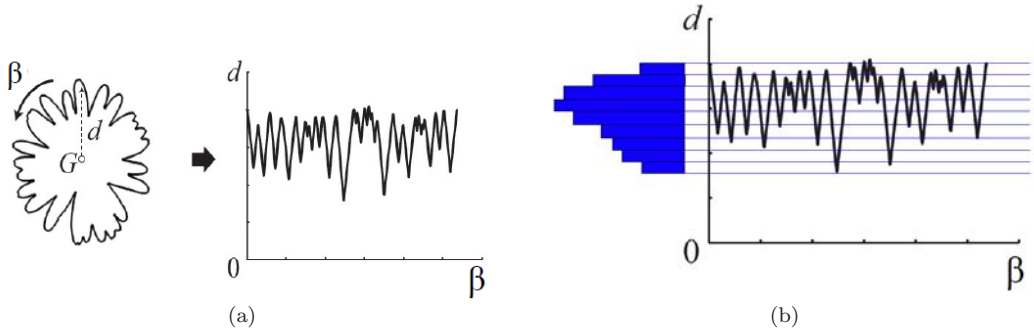


Figure B.17 — Distance histogram feature for a flower (a) and projection feature are shown.

The area ratio feature is extracted from the mask of the flower, which is the normalized area of the mask by the square of the maximum of the distance vector. The number of petals are calculated from the distance vector. The distances are thresholded by the mean of them and then the number of petals is defined by half of the number of times this mean distance is crossed.

In order to take into account the texture of the flower two features taken from the FELib [Zhu, 2008], Gabor wavelets and Gist.

B.3.4 Results

The Visual Geometry Group of Oxford University has been working on flower recognition and has therefore created a dataset consisting of 102 flower categories and more than 8000 images. The flower mainly come from United Kingdom [Nilsback and Zisserman, 2008]. The second database can be found on the website [CRSF, 2011]. It gathers only flowers growing within Switzerland. Last source of images was created manually by Samsung S mobile phone. This demarche really brings an asset to the database as the testings with the phone will employ such self-taken pictures.

Finally by combining the three sources exposed in the previous paragraph, the final database contains 110 images. All images are segmented manually from the background. The repartition is roughly the following 60% from the Oxford dataset, 30% from the CRSF website and 10% of manual shots. Within each of the 29 categories there are between 2 and 6 flowers. A good variety of colors and shapes was obtained as it can be observed in Figure B.18.

For evaluation cross-validation method is used. Features are matched by Euclidean distance and the flower with the closest feature is selected as result. Further feature fusion method is evaluated by linear weighting scheme.

Evaluation results can be seen in Figure B.19. Results shows that background segmentation improve significantly the results of flower recognition using color histogram features. The most important features are the color, however other combination of the features can lead even better

Figure B.18 — Examples from the database of flower recognition algorithm.

results as shown with a weighted linear combination feature. The best result is at 69% using HSV color histogram.

Combination of features are designed by weighted combination of the similarity score. The weights were selected by exhausted search. The obtained best weights are 1, 1.1, 1.5, 1.2 for respectively to HSV Color Histogram, Moments, Distance vector and HSV-based color histogram with background. Other features were not increase significantly the result of the fusion. A final precision of 80% has then been achieved.

B.3.5 Conclusion

The flower recognition scenario is the following; The user takes a picture from a flower and he/she indicates the flower region using the touch screen of the mobile phone. This information is sent to the server. Segmentation is done on the server which results in a binary mask of the flower. Specific features are extracted from the image of the flower, using the mask and then it is compared to the database. Finally, information from the detected flower is provided to the user on the mobile phone.

Segmentation step has been proven to be crucial and a marker-controlled watershed algorithm has been used. 18 different features were implemented specific to flower recognition, namely contour, texture, color based features. Final results shows that the best feature is HSV color histogram with 69% precision. However weighted linear combination of features results with 80% precision.

In this Section a mobile based flower recognition algorithm demonstrated the importance of user-interaction and feature fusion method.

In future work, benefit can be considered from additional information available in a phone like when or where the picture was taken. Also additional leaves analysis can improve the recognition rate of the algorithm.

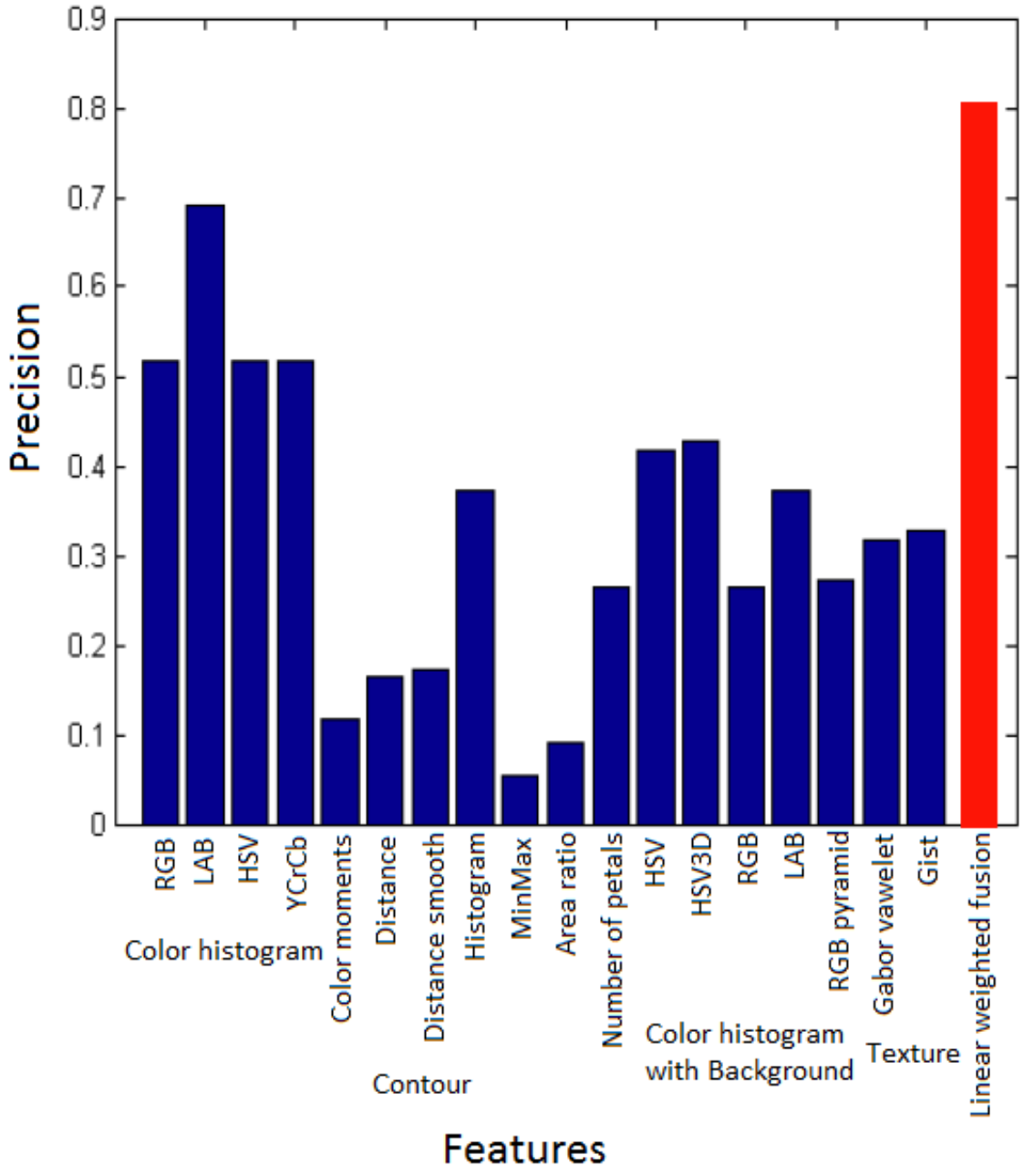


Figure B.19 — Results form each feature separately and with weighted linear combination are shown.

Victory is ever there where union of hearts is.

Publius Syrus (1st century BC)

Bibliog raphy

M. Ames, M. Naaman (2007). Why we tag: Motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2007)*, pp. 971–980.

D. Arthur, S. Vassilvitskii (2006). How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG '06, pp. 144–153, ACM, New York, NY, USA.

D. H. Ballard (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* **13**(2):111–122.

H. Bay, B. Fasel, L. V. Gool (2006a). Gool. interactive museum guide: Fast and robust recognition of museum objects. In *In Proc. Int. Workshop on Mobile Vision*.

H. Bay, T. Tuytelaars, L. Van Gool (2006b). Surf: Speeded-up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, pp. 404–417.

J. S. Beis, D. G. Lowe (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *In Proc. IEEE Conf. Comp. Vision Patt. Recog*, pp. 1000–1006.

S. Belongie, J. Malik, J. Puzicha (2002). Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(4):509 –522.

J. L. Bentley (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM* **18**:509–517.

J. Y. Bouguet (2002). *Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm*. Tech. rep., Intel Corporation Microprocessor Research Labs.

S. Brandt, J. Laaksonen, E. Oja (2000). Statistical shape features in content-based image retrieval.

W. Burgard *et al.* (1998). The interactive museum tour-guide robot. In *Proc. of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*.

M. Calonder, V. Lepetit, C. Strecha, P. Fua (2010). BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*.

J. Canny (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6):679–698.

D. Carboni, S. Sanna, P. Zanarini (2006). GeoPix: image retrieval on the geo web, from camera click to mouse click. In *Proceedings of the 8th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile HCI 2006)*, pp. 169–172.

A. Chalechale (2007). Coin recognition using image abstraction and spiral decomposition. In *9th International Symposium on Signal Processing and Its Applications*, pp. 1–4.

V. Chandrasekhar *et al.* (2009). CHoG: Compressed histogram of gradients A low bit-rate feature descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 2504–2511.

D. M. Chen *et al.* (2009). Streaming mobile augmented reality on mobile phones. In *ISMAR '09: Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pp. 181–182, IEEE Computer Society, Washington, DC, USA.

E. Choi, J. Lee, J. Yoon (2006). Feature extraction for bank note classification using wavelet transform. In *Pattern Recognition, 18th International Conference on Pattern Recognition*, vol. 2, pp. 934–937.

C. Cotsaces, N. Nikolaidis, I. Pitas (2006). Video shot detection and condensed representation: A review. *IEEE Signal Processing Magazine* **23**(2):28–37.

D. Crandall, L. Backstrom, D. Huttenlocher, J. Kleinberg (2009). Mapping the world's photos. In *Proceedings of the 18th International Conference on World Wide Web (WWW 2009)*, pp. 761–770.

CRSF (2011). CRSF - Centre du Reseau Suisse de Floristique. Available at: `http://www.crsf.ch/`.

DailyMotion (2010). DailyMotion Statistics. Available at: `http://www.heroesforhire.info/press/press_release_los_angeles_june_9.htm`.

N. Dalal, B. Triggs (2005). Histograms of oriented gradients for human detection. In *In CVPR*, pp. 886–893.

M. Das, R. Manmatha, E. M. Riseman (1999). Indexing flower patent images using domain knowledge. *IEEE Intelligent Systems* **14**:24–33.

T. N. W. David H. Hubel (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology* **28**:229–289.

M. Douze *et al.* (2009). Evaluation of gist descriptors for web-scale image search. In *International Conference on Image and Video Retrieval*, ACM.

C. Enroth-Cugell, J. G. Robson (1966). The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of physiology* **187**(3):517–552.

Facebook (2010). Facebook Statistics. Available at: `http://www.facebook.com/press/info.php?statistics`.

T. Fawcett (2006). An introduction to ROC analysis. *Pattern Recognition Letters* **27**(8):861–874.

L. Fei-Fei, P. Perona (2005). A bayesian hierarchical model for learning natural scene categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, vol. 2, pp. 524–531 vol. 2, IEEE Computer Society, Washington, DC, USA.

L. Fejes Tóth (1972). *Lagerungen in der Ebene auf der Kugel und im Raum*. Springer-Verlang, Berlin, 2nd edn.

C. Fellbaum (ed.) (1998). *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA; London.

P. Felzenszwalb, D. Mcallester, D. Ramanan (2008). A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

M. A. Fischler, R. C. Bolles (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6):381–395.

P. Föckler *et al.* (2005). PhoneGuide: museum guidance supported by on-device object recognition on mobile phones. In *MUM '05: Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, pp. 3–10, ACM, New York, NY, USA.

A. Frank (2004). *On Kuhns Hungarian method - a tribute from Hungary*. Tech. rep., Egerváry Research Group on Combinatorial Optimization.

W. T. Freeman, E. H. Adelson (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**:891–906.

H. Fuchs, Z. M. Kedem, B. F. Naylor (1980). On visible surface generation by a priori tree structures. In *Computer Graphics*, pp. 124–133.

M. Fuerst *et al.* (2006). Intelligent high-speed, high-variant automation of universal coin sorting for charity organizations. In *IEEE International Conference on Robotics and Automation*, pp. 303–308.

M. Fukumi, S. Omatu (1993). Designing a neural network for coin recognition by a genetic algorithm. In *Proceedings of International Joint Conference on Neural Networks*, vol. 3, pp. 2109–2112.

S. Gammeter, L. Bossard, T. Quack, L. Van Gool (2009). I know what you did last summer: Object level auto-annotation of holiday snaps. In *Proceedings of the 20th International Conference on Computer Vision (ICCV 2009)*.

J. Geigel, A. Loui (2003). Using genetic algorithms for album page layouts. *IEEE Multimedia* **10**(4):16–27.

B. Girod (2009). Mobile visual search. In *Workshop on Mobile Visual Search*.

M. A. Goodale, A. D. Milner (1992). Separate visual pathways for perception and action. *Trends in neurosciences* **15**(1):20–25.

L. V. Gool *et al.* (2009). Mining from large image sets. In *ACM Int. Conference on Image and Video Retrieval*.

C. Grana, G. Tardini, R. Cucchiara (2005). MPEG-7 compliant shot detection in sport videos. In *Proc. Int. Symposium on Multimedia*, pp. 395–402.

R. L. Gregory (1997). Knowledge in Perception and Illusion. *Royal Society of London Philosophical Transactions Series B* **352**:1121–1127.

F. Groen, G. den Boer, A. van Inge, R. Stam (1992). A chess-playing robot: lab course in robot sensor integration. *IEEE Instrumentation and Measurement Society* **41**(6):911–914.

R. H. Hardin, N. J. Sloane, W. D. Smith (1997). Spherical coverings. Retrieved from `http://www.sphopt.com/math/question/covering.html`.

C. Harris, M. Stephens (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151.

R. I. Hartley, A. Zisserman (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edn.

J. Hays, A. A. Efros (2008). IM2GPS: estimating geographic information from a single image. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8.

P. S. Hiremath, J. Pujari (2007). Content based image retrieval using color, texture and shape features. In *Proceedings of the 15th International Conference on Advanced Computing and Communications*, pp. 780–784, IEEE Computer Society, Washington, DC, USA.

L. Hollenstein, R. Purves (2010). Exploring place through user-generated content: using Flickr to describe city cores. *Journal of Spatial Information Science* pp. 1–29.

J. Hong (2009). Computer vision for the masses. In *Workshop on Mobile Visual Search*.

I.-S. Hsieh, K.-C. Fan (2000). Color image retrieval using shape and spatial properties. *Pattern Recognition, International Conference on* **1**:5023.

C.-R. Huang, H.-P. Lee, C.-S. Chen (2008). Shot change detection via local keypoint matching. *IEEE Transactions on Multimedia* **10**(6):1097–1108.

C. Im, H. Nishida, T. L. Kunii (1998). Recognizing plant species by leaf shapes-a case study of the acer family. In *Proceedings of the 14th International Conference on Pattern Recognition-Volume 2 - Volume 2*, ICPR '98, pp. 1171–, IEEE Computer Society, Washington, DC, USA.

P. Indyk, R. Motwani (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, ACM, New York, NY, USA.

International Press Telecommunications Council (2009). *IPTC Photo Metadata Standard, IPTC Core 1.1 and IPTC Extension 1.1.* Tech. rep., International Press Telecommunications Council.

I. Ivanov *et al.* (2010a). Geotag Propagation in Social Networks Based on User Trust Model. *Multimedia Tools and Applications (Springer), Special Issue on Social Mining and Search* .

I. Ivanov *et al.* (2010b). Object-based Tag Propagation for Semi-automatic Annotation of Images. In *Proceedings of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval*, pp. 497–506.

H. Jégou, M. Douze, C. Schmid (2011). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **33**(1):117–128. To appear.

T. Kadir, A. Zisserman, M. Brady (2004). An affine invariant salient region detector.

Y. Ke, R. Sukthankar (2004). PCA-SIFT: a more distinctive representation for local image descriptors. In *CVPR*, vol. 2, pp. 506–513.

L. S. Kennedy, M. Naaman (2008). Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th International Conference on World Wide Web (WWW 2008)*, pp. 297–306.

J.-H. Kim, R.-G. Huang, S.-H. Jin, K.-S. Hong (2009). Mobile-based flower recognition system. In *Proceedings of the 2009 Third International Symposium on Intelligent Information Technology Application - Volume 03*, IITA '09, pp. 580–583, IEEE Computer Society, Washington, DC, USA.

N. Krishnan, M. S. Banu, C. C. Christiyana (2007). Content based image retrieval using dominant color identification based on foreground objects. *Computational Intelligence and Multimedia Applications, International Conference on* **3**:190–194.

F. Kusunoki, M. Sugimoto, H. Hashizume (2002). Toward an interactive museum guide system with sensing and wireless network technologies. *Wireless and Mobile Technologies in Education, IEEE International Workshop on* **0**:99.

S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back (1997). Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks* **8**:98–113.

S. Lazebnik, C. Schmid, J. Ponce (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2169 – 2178.

Y. Lecun, L. Bottou, Y. Bengio, P. Haffner (1998). Gradient-Based Learning Applied to Document Recognition. In *Proceedings of the IEEE*, vol. 86, pp. 2278–2324.

J.-K. Lee, I.-H. Kim (2003). New recognition algorithm for various kinds of euro banknotes. In *Industrial Electronics Society, 2003. IECON '03. The 29th Annual Conference of the IEEE*, vol. 3, pp. 2266–2270.

R. Lienhart (1999). Comparison of automatic shot boundary detection algorithms. In *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases*, pp. 290–301.

S. Lindstaedt *et al.* (2008). Recommending tags for pictures based on text, visual content and user context. In *Proceedings of the 3rd International Conference on Internet and Web Applications and Services (ICIW 2008)*, pp. 506–511.

N. K. Logothetis, J. Pauls, T. Poggio (1995). Shape representation in the inferior temporal cortex of monkeys. *Current Biology* **5**(5):552–563.

D. G. Lowe (1999). Object Recognition from Local Scale-Invariant Features. *Computer Vision, IEEE International Conference on* **2**:1150–1157 vol.2.

D. G. Lowe (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2):91–110.

J. B. MacQueen (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam, J. Neyman (eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press.

B. Manjunath, J.-R. Ohm, V. Vasudevan, A. Yamada (2001). Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on* **11**(6):703 –715.

A. Manzanera (2010). Local jet based similarity for nl-means filtering. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pp. 2668–2671, IEEE Computer Society, Washington, DC, USA.

J. Matas, O. Chum, U. Martin, T. Pajdla (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference*, vol. 1, pp. 384–393, London.

U. t. Mike Snider (2011). Social games: news and survey findings. Available at: `http://content.usatoday.com/communities/gamehunters/post/2010/02/social-games-news-and-survey-findings/1`.

K. Mikolajczyk, C. Schmid (2002). An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision (ECCV 2002)*, pp. 128–142, Springer Verlag.

K. Mikolajczyk *et al.* (2005). A comparison of affine region detectors. *International Journal of Computer Vision* **65**(1–2):43–72.

J.-M. Morel, G. Yu (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.* **2**(2):438–469.

A. Multimedia (2010). HP Challenge 2010 Dataset: High Impact Visual Communication. Available at: `http://comminfo.rutgers.edu/conferences/mmchallenge/2010/02/10/hp-challenge-2010`.

M. Naaman, Y. J. Song, A. Paepcke, H. Garcia-Molina (2004). Automatic organization for digital photographs with geographic coordinates. In *Proceedings of the International Conference on Digital Libraries*, pp. 53–62.

G. Nagy, J. Zou (2002). Interactive visual pattern recognition. In *Proc. International Conference Pattern Recognition*, pp. 478–481.

L. Najman, M. Schmitt (1994). Watershed of a continuous function. *Signal Process.* **38**:99–112.

A. Neubeck, L. V. Gool (2006). Efficient non-maximum suppression. *International Conference on Pattern Recognition* pp. 850–855.

J. Neufeld, T. Hall (2010). Probabilistic location of a populated chessboard using computer vision. In *Proceedings of the 53rd International Midwest Symposium on Circuits and Systems (MWSCAS)*.

M. Neuhaus, H. Bunke (2006). Edit distance based kernel functions for structural pattern classification. *Pattern Recognition* **39**:1852–1863.

M.-E. Nilsback, A. Zisserman (2008). Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

NIST (2010). Cloud computing definition, national insitute of standards and technology. Available at: `http://csrc.nist.gov/groups/SNS/cloud-computing/index.html`.

D. Nister, H. Stewenius (2006). Robust scalable recognition with a vocabulary tree. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pp. 2161–2168.

T. Ojala, M. Pietikainen, D. Harwood (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, pp. 582 –585 vol.1.

D. K. Park, Y. S. Jeon, C. S. Won (2000). Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM workshops on Multimedia*, MULTIMEDIA '00, pp. 51–54, ACM, New York, NY, USA.

J. Philbin *et al.* (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8.

Pingdom (2011). Internet 2010 in numbers. Available at: `http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers`.

T. Quack, B. Leibe, L. Van Gool (2008). World-scale mining of objects and events from community photo collections. In *Proceedings of the IEEE International Conference on Content-based Image and Video Retrieval (CIVR 2008)*, pp. 47–56.

M. Riesenhuber, T. Poggio (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience* **2**:1019–1025.

E. Rosten, T. Drummond (2006). Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, vol. 1, pp. 430–443.

F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce (2004). Segmenting, modeling, and matching video clips containing multiple moving objects. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pp. 914–921.

B. Ruf, E. Kokiopoulou, M. Detyniecki (2008). Mobile museum guide based on fast SIFT recognition. In *6th International Workshop on Adaptive Multimedia Retrieval*, Lecture Notes in Computer Science, Springer.

B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision* **77**(1-3):157–173.

T. Saitoh, K. Aoki, T. Kaneko (2003). Automatic extraction of object region from photographs. In *Proceedings of the 13th Scandinavian conference on Image analysis*, SCIA'03, pp. 1130–1137, Springer-Verlag, Berlin, Heidelberg.

T. Saitoh, K. Aoki, T. Kaneko (2004). Automatic recognition of blooming flowers. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01*, ICPR '04, pp. 27–30, IEEE Computer Society, Washington, DC, USA.

T. Saitoh, T. Kaneko (2000). Automatic recognition of wild flowers. In *International Conference on Pattern Recognition*, pp. 2–2507.

C. Schmid, A. Zisserman (1998). The geometry and matching of curves in multiple views. In H. Burkhardt, B. Neumann (eds.), *Computer Vision ECCV'98*, vol. 1406 of *Lecture Notes in Computer Science*, pp. 394–409, Springer Berlin.

T. Serre, T. Poggio (2010). A neuromorphic approach to computer vision. *Commun. ACM* **53**:54–61.

T. Serre, M. Riesenhuber (2004). *Realistic Modeling of Simple and Complex Cell Tuning in the HMAX Model, and Implications for Invariant Object Recognition in Cortex*. Tech. rep., Massachusetts Institute of Technology.

T. Serre, L. Wolf, T. Poggio (2005a). Object recognition with features inspired by visual cortex. In *In CVPR*, pp. 994–1000.

T. Serre *et al.* (2005b). A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. In *AI Memo*.

M. Sezgin, B. Sankur (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* **13**(1):146–168.

G. Shan, L. Peng, L. Jiafeng, T. Xianglong (2009). The design of hmm-based banknote recognition system. In *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, vol. 4, pp. 106–110.

E. Shechtman, M. Irani (2007). Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*.

B. Sigurbjörnsson, R. van Zwol (2008). Flickr tag recommendation based on collective knowledge. In *Proceeding of the 17th International Conference on World Wide Web (WWW 2008)*, pp. 327–336.

P. Y. Simard, D. Steinkraus, J. C. Platt (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, pp. 958–, IEEE Computer Society, Washington, DC, USA.

J. Sivic, F. Schaffalitzky, A. Zisserman (2006). Object level grouping for video shots. *International Journal of Computer Vision* **67**(2):189–210.

J. Sivic, A. Zisserman (2003). Video Google: A text retrieval approach to object matching in videos. In *Proc. Inter. Conf. on Computer Vision*, pp. 1470–1477.

J. Sivic, A. Zisserman (2006). Video Google: Efficient visual search of videos. In J. Ponce, M. Hebert, C. Schmid, A. Zisserman (eds.), *Toward Category-Level Object Recognition*, vol. 4170 of *LNCS*, pp. 127–144, Springer.

A. W. M. Smeulders *et al.* (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(12):1349–1380.

S. M. Smith, J. M. Brady (1997). Susan - a new approach to low level image processing. *Int. J. Comput. Vision* **23**:45–78.

R. J. Sternberg (2002). *Cognitive Psychology*. Wadsworth Publishing.

M. Stricker, M. Orengo (1995). Similarity of color images. In W. Niblack & R. C. Jain (ed.), *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 2420 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 381–392.

M. J. Swain, D. H. Ballard (1991). Color indexing. *International Journal of Computer Vision* **7**:11–32.

K. Tam, J. Lay, D. Levy (2008). Automatic grid segmentation of populated chessboard taken at a lower angle view. *Digital Image Computing: Techniques and Applications* **0**:294–299.

Technical Standardization Committee on AV & IT Storage Systems and Equipment (2002). *Exchangeable image file format for digital still cameras: Exif Version 2.2*. Tech. Rep. JEITA CP-3451, Technical Standardization Committee on AV & IT Storage Systems and Equipment.

S. Thrun *et al.* (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research* **19**:972–999.

S. S. Tsai *et al.* (2009). Location coding for mobile image retrieval. In *Mobimedia '09: Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, pp. 1–7, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium.

T. Tuytelaars, L. V. Gool (2000). Wide baseline stereo matching based on local, affinely invariant regions. In *In Proc. BMVC*, pp. 412–425.

T. Tuytelaars, K. Mikolajczyk (2007). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision* **3**(3):177–280.

T. Tuytelaars, L. Van Gool (1999). Content-based image retrieval based on local affinely invariant regions. In D. Huijsmans, A. Smeulders (eds.), *Visual Information and Information Systems*, vol. 1614 of *Lecture Notes in Computer Science*, pp. 656–656, Springer Berlin, Heidelberg.

T. Tuytelaars, L. Van Gool (2004). Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision* **59**:61–85.

O. Tuzel, F. Porikli, P. Meer (2006). Region covariance: A fast descriptor for detection and classification. In *In Proc. 9th European Conf. on Computer Vision*, pp. 589–600.

P. Vajda, F. Dufaux, T. H. Minh, T. Ebrahimi (2009a). Graph-based approach for 3D object duplicate detection. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2009)*, pp. 254–257.

P. Vajda, L. Goldmann, T. Ebrahimi (2009b). Analysis of the limits of graph-based object duplicate detection. In *Proceedings of the International Symposium on Multimedia*, pp. 600–605.

P. Vajda, I. Ivanov, L. Goldmann, T. Ebrahimi (2011a). Let Epitome summarize your photo collection! In *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo*.

P. Vajda, I. Ivanov, L. Goldmann, T. Ebrahimi (2011b). Omnidirectional object duplicate detection. In *Proceedings of the 14th Digital Signal Processing Workshop*.

P. Vajda, I. Ivanov, L. Goldmann, T. Ebrahimi (2011c). Social game Epitome vesus automatic visual analysis. In *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo*.

P. Vajda *et al.* (2010a). 3D Object Duplicate Detection for Video Retrieval. In *Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services*.

P. Vajda *et al.* (2010b). Propagation of geotags based on object duplicate detection. In *Proceedings of SPIE*, vol. 7798.

P. Vajda *et al.* (2010c). Robust Duplicate Detection of 2D and 3D Objects. *International Journal of Multimedia Data Engineering and Management* Invited paper.

E. Valle, D. Picard, M. Cord (2009). Geometric consistency checking for local-descriptor based document retrieval. In *Proceedings of the 9th ACM symposium on Document engineering*, DocEng '09, pp. 135–138, ACM, New York, NY, USA.

L. von Ahn, L. Dabbish (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*, pp. 319–326.

L. von Ahn, R. Liu, M. Blum (2006). Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2006)*, pp. 55–64.

X. Wang, T. X. Han, S. Yan (2009). An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 32 –39.

T. Warshall (1962). A theorem on boolean matrices. *Journal of the ACM* **9**:11–12.

A. Weiss (2007). Computing in the clouds. *COMPUTING* **16**.

Wikipedia (2010). Wikipedia - Flickr. Available at: `http://en.wikipedia.org/wiki/Flickr`.

L. Wiskott (2001). How does our visual system achieve shift and size invariance?

W. Wolf (1996). Key frame selection by motion analysis. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 1228–1231.

W. R. Woodward, R. S. Cohen (1991). *World views and scientific discipline formation.* Kluwer Academic Publishers, Dordrecht ; Boston :.

T. Yeh, K. Grauman, K. Tollmar, T. Darrell (2005). A picture is worth a thousand keywords: image-based object search on a mobile platform. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pp. 2025–2028, ACM, New York, NY, USA.

Youtube (2010). YouTube Fact Sheet. Available at: `http://www.youtube.com/t/fact_sheet`.

D. Zhang (2006). A generative-discriminative hybrid method for multi-view object detection. In *In Proc. of CVPR*, p. 2006.

H. Zhang (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition* **30**(4):643–658.

J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision* **73**(2):213–238.

Y. Zheng *et al.* (2009). Tour the World: Building a web-scale landmark recognition engine. In *Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 1085–1092.

M. Zhenjiang, M. H. Gandelin, Y. Baozong (2006). An oopr-based rose variety recognition system. *Eng. Appl. Artif. Intell.* **19**:79–101.

J. Zhu (2008). Feature Extraction Library - FELib. Available at: `http://www.vision.ee.ethz.ch/~zhuji/felib.html`.

J. Zhu, S. C. Hoi, M. R. Lyu, S. Yan (2008). Near-duplicate keyframe retrieval by nonrigid image matching. In *Proceeding of the 16th ACM international conference on Multimedia*, MM '08, pp. 41–50, ACM, New York, NY, USA.

J. Zou, G. Nagy (2004). Evaluation of model-based interactive flower recognition. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02*, ICPR '04, pp. 311–314, IEEE Computer Society, Washington, DC, USA.

M. Özuysal, M. Calonder, V. Lepetit, P. Fua (2009). Fast keypoint recognition using random ferns. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Zynga (2009a). Affine covariant features. Available at: `http://www.robots.ox.ac.uk/~vgg/research/affine/`.

Zynga (2009b). Treasuer isle facebook application by zynga. Available at: `http://apps.facebook.com/treasureisle/`.

If I feel unhappy, I do mathematics to become happy. If I am happy, I do mathematics to keep happy.

Ha rossz kedvem van, matematizálok, hogy jó kedvem legyen. Ha jó kedvem van, matematizálok, hogy megmaradjon a jó kedvem.

Alfred Renyi (1921 — 1970)

Curriculum Vitæ

Name:	Peter VAJDA
Citizenship:	Hungarian, EU
Birthdate:	October 12th, 1982
Birthplace:	Kaposvár, Hungary
Marital status:	single

Contact information

Address:	César Roux 9
	CH-1005 Lausanne
Phone:	+41 76 221 3982
Email:	`vajdap@gmail.com`
Web page:	`http://people.epfl.ch/peter.vajda`

Objective

My current **research interest** concerns **multimedia applications, artificial intelligence, algorithms** and **theory**. My professional experience and educational background have allowed me to develop extensive competencies in the areas of **multimedia signal processing, mathematical analysis, graph theory, program development** and **project management**. I am an open-minded and creative person with **international experience** and my **language skills** suggest assignments in a multinational environment.

Professional experience

- ***09/2007 – present***: **Swiss Federal Institute of Technology (EPFL)**, Lausanne, Switzerland, *Research Assistant*
  - **Research** in the area of image and video retrieval for object recognition.
  - **Teaching experience:** Media security, Image and video processing, Programming environment, Compiler constructions.

  - European Network of Excellence: Petamedia, K-Space, Visnet II
  - **Program development** and research
    * **Cheese**: Visual search webpage.
    * **Epitome**: Facebook and mobile game for photo album.
    * **Mobile application** for object recognition: flower, museum guide, tourist guide.
  - Wrote Scientific Grant application for Swiss National Foundation: **CHF 203k**.
  - **MPEG** and **JPEG** Standardization groups: Contribution and conference organization.

- ***01/2008 - 06/2008***: **Telecontrol**, Switzerland, *Analyzer, Developer*
  - **IPTV** channels, teletext pages, Electronic Program Guide are detected in real time, while user watched the TV, by evaluating only channel data of digital Set Top Box.

- ***09/2007 - 09/2008***: **NFI user management software**, Hungary, *Developer*
  - Project and user management software for communication and project decomposition.
  - Used tools and languages: Java, RMI, SQL.

- ***09/2008 - present***: **Polyprog association**, Switzerland, *Committee member*
  - The mission is to promote the interests and skills in algorithmic programming.
  - **Organization** of Helvetic Coding Contest: Swiss programming competition.
  - **Financial administration.**
  - **Training program** for students to participate ACM ICPC, CodeCup, Google Jam.

- ***09/2010 - present***: **Danubia Hungarian folk dance association**, Switzerland, *Committee member*
  - Web development and advertisement.

EDUCATION

- ***09/2007 - 09/2011***: **Ph.D. in Computer, Communication and Information Sciences**
  - Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, Adv.: Prof. Touradj Ebrahimi
  - Ranked as **Europe's #1** and **world's #15** university in the field of Engineering and CS in ARWU 2009
  - Thesis title: Object duplicate detection.

- ***09/2005 - 09/2006***: **M.Sc. in Computer Science**
  - Vrije Universiteit, Amsterdam, Netherlands, Adv.: Prof. A. E. Eiben
  - Thesis title: Comparing on-the fly selection adjustment mechanisms in Evolution Computing

- ***09/2001 - 09/2007***: **M.Sc. in Program Designer Mathematician**
  - Eötvös Loránd University, Budapest, Hungary, Adv.: Dr. Habil András Lörincz
  - Thesis title: Prediction algorithms embedding to JCommander filemanager

Awards

- **Best poster** in S3MR, 2011: "Swiss Cheese for visual search"
- IV. place in **Startup Weekend**, Lausanne, 2011: "Swiss Bomb"
- IV place in **Helvetic Coding Competition**. Best place from EPFL, 2011
- Finalist in **ACM Multimedia Grand Challenge**, 2010: "Social Game Epitome"
- **Excellent Student** of the Faculty Award (Faculty of informatics), 2005
- II. Place, Pázmány-Eötvös scientific competition for student, 2004
- III. Place, High School International Hungarian **Mathematics Competition**, 2000

Skills

Languages

- Hungarian: mother tongue
- English: fluent (CEF level C2)
- French: beginner (CEF level A2)
- German: beginner (CEF level A2)

Computer literacy

- Multimedia signal processing: visual features in images and videos, feature extraction and matching, image retrieval, pattern recognition, machine learning, social network data analysis, MPEG and IPTV streams metering
- Programming languages: C/C++, Java, Matlab, Mathematica, HTML, PHP, Perl, MySQL, Python, Ada, Erlang, Bash, SML, Assembly
- Software development: UML, Qt, GTK, MS Visual Studio, Eclipse, Matlab, MS Office, MS Project, Google API, Android, Oracle, PVM, MPI, RMI, Socket
- Operating Systems: MS Windows, Linux/Unix, Mac OS X, VMS

There are more things in heaven and earth, Horatio, Than are dreamt of in your philosophy.

Shakespeare (1564 — 1616)

Publications

Journal papers

I. Ivanov, P. Vajda, J.-S. Lee and T. Ebrahimi, *In Tags We Trust: Trust Modeling in Social Tagging of Multimedia Content*, IEEE Signal Processing Magazine (SPM), 2011.

I. Ivanov, P. Vajda, J.-S. Lee, L. Goldmann and T. Ebrahimi, *Geotag Propagation in Social Networks Based on User Trust Model*, Multimedia Tools and Applications (Springer), Special Issue on Social Mining and Search, 2010.

P. Vajda, I. Ivanov, L. Goldmann, J.-S. Lee and T. Ebrahimi, *Robust Duplicate Detection of 2D and 3D Objects*, International Journal of Multimedia Data Engineering and Management, Vol. 1, Nr. 3, pp. 19-40, 2010.

Conference papers

P. Vajda, I. Ivanov, L. Goldmann and T. Ebrahimi, *Omnidirectional object duplicate detection*, Proc. of the 14th Digital Signal Processing Workshop, 2011.

P. Vajda, I. Ivanov, L. Goldmann and T. Ebrahimi, *Let Epitome summarize your photo collection!*, Proc. of the 2011 IEEE International Conference on Multimedia and Expo, 2011.

P. Vajda, I. Ivanov, L. Goldmann and T. Ebrahimi, *Social game Epitome versus automatic visual analysis*, Proc. of the 2011 IEEE International Conference on Multimedia and Expo, 2011.

I. Ivanov, P. Vajda, J.-S. Lee and T. Ebrahimi, *Epitome - A Social Game for Photo Album Summarization*, Proc. of the ACM SIGMM International Conference on Multimedia, the First ACM International Workshop on Connected Multimedia, pp. 33-38, 2010.

P. Vajda, I. Ivanov, J.-S. Lee, L. Goldmann and T. Ebrahimi, *Propagation of geotags based on object duplicate detection*, Proc. of SPIE, Vol. 7798, 2010.

P. Vajda, I. Ivanov, L. Goldmann, J.-S. Lee and T. Ebrahimi, *3D Object Duplicate Detection for Video Retrieval*, Proc. of the 11th International Workshop on Image Analysis for Multimedia Interactive Services, 2010.

I. Ivanov, P. Vajda, L. Goldmann, J.-S. Lee and T. Ebrahimi, *Object-based Tag Propagation for Semi-automatic Annotation of Images*, Proc. of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval, pp. 497-506, 2010.

P. Vajda, L. Goldmann and T. Ebrahimi, *Analysis of the Limits of Graph-Based Object Duplicate Detection*, Proc. of IEEE International Symposium on Multimedia, 2009.

P. Vajda, F. Dufaux, T. Ha M. and T. Ebrahimi, *Graph-Based Approach for 3D Object Duplicate Detection*, Proc. of the 10th International Workshop on Image Analysis for Multimedia Interactive Services, 2009.

L. Goldmann, T. Adamek, P. Vajda, M. Karaman, R. Mörzinger, E. Galmar, T. Sikora, N. O'Connor, T. Ha-Minh, T. Ebrahimi, P. Schallauer and B. Huet, *Towards Fully Automatic Image Segmentation Evaluation*, Advanced Concepts for Intelligent Vision Systems, pp. 566-577, 2008.

P. Vajda, A.E. Eiben, and W. Hordijk, G. Rudolph, *Parameter Control Methods for Selection Operators in Genetic Algorithms*, Parallel Problem Solving from Nature, (PPSN X), Springer, 2008.