Indoor Place Recognition Using Support Vector Machines

Master's Thesis in Computer Science
Praca dyplomowa magisterska na kierunku Informatyka

ANDRZEJ PRONOBIS

Under the Supervision of

Barbara Caputo
Prof. dr hab. inż. Jacek Łęski

Computer Vision and Active Perception Laboratory
Department of Numerical Analysis and Computing Science
Royal Institute of Technology
Stockholm, Sweden

Instytut Informatyki
Wydział Automatyki, Elektroniki i Informatyki
Politechnika Śląska Gliwice

November 2005

Abstract

The ability to recognize places on the basis of visual perceptual information is a fundamental property of human beings, and thus determines the way we communicate and alter our surroundings. Consequently, it becomes indispensable to provide similar capabilities for machines aiming to interact with humans and man-made environment. In this thesis we address the problem of visual indoor place recognition. We propose a solution based on Support Vector Machines employing both global and local image descriptors. Since robustness and efficiency are crucial for every recognition system aiming to work in real-world settings, we put special emphasis on these properties. We build a database comprising several sets of pictures acquired in five rooms of different functionality, under various conditions. We then use it in order to evaluate the performance of our system, and achieve very good results in presence of variations that occur in real environments. Additionally, for sake of efficiency, we implement an algorithm allowing for an exact simplification of support vector solutions [19]. We further extend the original algorithm so that it could provide higher efficiency gain by means of approximation. The results reported in the thesis show great potential of our method in a wide range of computer vision applications and prove that support vector solutions can be successfully applied to the place recognition problems.

Contents

Chapter 1

Introduction

The ability to acquire, represent, and match the perceptual information to the memories of places stored in an internal cognitive map is a fundamental property of human beings and numerous animals. Although this task is performed in a variety of ways, it is always a complex process involving numerous perfectly cooperating mechanisms. In this thesis we address the problem of *visual indoor place recognition*. We put special emphasis on the robustness and efficiency, as these are the key properties of every recognition system aiming to be used in real-world applications.

The most successful visual place recognition system was designed by nature. Humans are one of the beneficiaries of this invention and therefore are highly effective in exploring the surrounding environment. Our visual recognition system is extremely robust to changing illumination conditions and variations in the environment as well as to noise and occlusions. We can easily recognize a familiar place when it is crowed in the middle of the day, even if we saw it for the first time empty during the night. In the recognition process, we make use of contextual information according to our experience. Moreover, our internal representation of the place is constantly updated due to continuous learning. As a result, we are able to recognize and understand complex scenes in less than 100 ms.

Inability to recognize places would prevent us from performing many basic tasks, as they require topographical orientation. Nowadays, more and more tasks are performed by robots which need efficient and robust localization algorithms so as to become mobile. Despite the fact that numerous non-visual localization methods has been developed (e.g. laser-based SLAM), utilization of visual-based methods is indispensable in order to perform real-world tasks. This is due to the fact that human perception is primarily visual, and understanding and making use of visual information is essential to provide human-robot interaction. Visual properties of a place can be used to determine its functional category (e.g. kitchen, office), and visual landmarks are commonly used by humans to plan paths and describe places. All this motivated many researchers to employ visual information in mobile robot localization. However, it is still an open issue how to make the algorithms efficient and provide robustness to variability of the environment and changing illumination

conditions.

Although, *topological localization* of mobile robots is the most natural application for visual place recognition, it may have many other uses. Place recognition systems can be, inter alia, a valuable source of contextual information for *content-based image retrieval systems*. Today, Internet technologies such as World Wide Web give access to huge amounts of data, large percentage of which are images. Consequently, the ability to retrieve images on the basis of a description of their content becomes indispensable. In case of pictures, the information about location is one of the most fundamental.

Place recognition may be also coupled with other computer vision algorithms. Such problems as *object* or *action recognition* may be greatly simplified by exploiting the knowledge about context in which they occur. We can also imagine place recognition systems cooperating with mobile devices. We could, for instance, create electronic guides giving information about the place observed through the lens of our digital camera. All these examples show that there are numerous applications for which the ability to recognize places can be extremely valuable.

The visual place recognition system described in this thesis was built around the Support Vector Machine classifier. Both global and local image features were employed in order to find the image representation that is best suited for the place recognition purposes. As it was already stated, special emphasis has been placed on the robustness and efficiency. Therefore, the system was tested under various conditions, and a support vector reduction algorithm was used in order to increase the recognition speed and decrease the memory requirements. More detailed information about the contribution of this thesis can be found in Section 1.2.

1.1 Related Work

The research on *place recognition* has been mostly conducted in the mobile robotics community, where the problem is referred to as *topological localization*. The use of visual cues for this purpose has increased in popularity, as the visual algorithms became more sophisticated, and constantly increasing computational power allowed many operations to be performed in real-time. As a result, several approaches to the vision-based topological localization have been proposed. These methods employ either regular cameras ([76, 73]) or omni-directional sensors [51] ([27, 77, 8, 46, 2]) in order to acquire images (Figure 2.3 contains comparison of images acquired using cameras of both types).

The main differences between the approaches relate to the way the scene is perceived, and thus the method used to extract characteristic features of the scene. *Landmark localization* techniques make use of either artificial or natural landmarks in order to extract information about position. An interesting approach to the problem was presented by Mata *et al.* [44, 45]. The system uses information signs as landmarks, and interprets them through its ability to read text and recognize icons. *Local image features* may also be regarded as natural landmarks. The SIFT

descriptor invented by Lowe [41] was successfully used for that purpose by Se *et al.* [69] and Andreasson *et al.* [2] (with modifications), while Tamimi and Zell [73] employed Kernel PCA to extract features from local patches. *Global features* are also commonly used for place recognition. Torralba [74] suggested to use a representation called the "gist" of the scene (used with modifications in [76, 75, 49]), which is a vector of principal components of outputs of a filter bank applied to the image. Several other approaches use color histograms [77, 8], eigenspace representation of images [27], Fourier coefficients of low frequency image components [46], or statistics of "textons" [65].

As it was already mentioned, the place recognition systems may also be coupled with other computer vision algorithms in order to achieve mutual performance improvement. Examples of a successful utilization of the information about place (derived from holistic representation of an image) to provide contextual priors for *object detection* and *recognition* are presented in works by Torralba [74], Murphy *et al.* [49], and Torralba *et al.* [75]. Their approach allows to simplify the object detection task by penalizing locations (and scales) where the objects are not expected to be found. On the same basis, during object recognition, the information about place is exploited to determine which types of objects are more likely to appear.

Another field of research which may benefit from developing robust place recognition algorithms is *content-based image retrieval*. Although, place recognition may be considered as a special case of global image annotation, it is still of interest to provide the ability to retrieve pictures imaging particular place (e.g. the office instead of an office). Detailed information about semantic description and modeling of natural scenes can be found in [81]. The interested reader is also referred to [71] for a review of 200 references in the content-based image retrieval.

Many of the previously mentioned approaches derive inspiration from studies on human scene perception. Human scene perception and recognition is studied in Section 2.4 which constitutes a review of a selection of publications on the subject.

1.2 Contribution of the Thesis

The presented thesis, as its primary contribution, provides a description of a visual indoor place recognition system aiming to work in real-world settings. During the design process of the system, the strongest emphasis has been placed on the robustness to the illumination conditions and variations in the environment, as well as, on the efficiency of the solution. The system determines the location on the basis of the analysis of one picture acquired using a regular camera. As no prior knowledge is required in order to determine the position, the system may be useful to solve variety of problems such as *global*[1] topological localization of mobile robots or content-based image retrieval. It may also be employed as a source of contextual information for other recognition systems.

[1]The adjective *global* indicates that no prior knowledge about the initial position of a robot is required in order to determine its current position.

The SVM algorithm [78, 17] was used in order to perform classification. To the knowledge of the authors, this is the first attempt to employ the Support Vector Machines for visual place recognition. Both local and global features of the scene were used and the performance of the algorithm was evaluated in each case. The SIFT descriptor [41], which has been shown to perform well for localization problems [69, 2], was used in order to represent the local image features. The local features could be combined with the SVM classifier thanks to the local kernel function presented in [82]. The experimental results with local features were then compared to those obtained using Composed Receptive Field Histograms [38] as global features. This type of features was previously used for object recognition problems [38], and was found to perform very well also for place recognition.

The presented system was extensively tested to achieve robustness to condition changes, which may occur in real environment. For the purpose of these experiments, a database comprising pictures of an indoor environment was created. The database was acquired using a regular camera within multi-room indoor environment over the span of two months, and contains pictures taken under three illumination and weather conditions. Detailed description of the database can be found in Chapter 3 and in [62].

The place recognition system introduced in this thesis, was designed in a manner allowing it to be implemented on a mobile robot in the future. For this reason, the employed algorithms must be efficient and low resource consuming. In order to achieve these goals, a method presented by Downs *et al.* [19] was used. The method allows for reducing the number of support vectors of a trained classifier on the basis of the fact, that the set of support vectors is usually not linearly independent in the feature space. This thesis contributes to this method by implementing the algorithm proposed in [19] using QR Factorization [28, 29] and introducing a threshold parameter which can be used to trade classification performance for the speed of the classifier. The modified method enables to achieve greater reduction, keeping the classification performance intact. Depending on the application, it is possible to reduce the solution even further by means of approximation.

1.3 Outline

The thesis is organized as follows. Chapter 2 defines the problem of place recognition, discusses several aspects of human scene perception, and finally presents an architecture of a typical visual recognition system. Chapter 3 gives a description of the KTH-INDECS database, which besides being a part of the contribution of this thesis, was used during all experiments with the place recognition.

More specific details about all parts of the system can be found in Chapters 4, 5, and 6. Chapter 4 presents the methods used to extract characteristic image features, based on which the images are classified using the SVM classifier (described in Chapter 5). The Support Vector Reduction algorithm is explained in Chapter 6.

The results of experiments with the place recognition system and Support Vector

Reduction algorithm are given in Chapters 7 and 8. The thesis concludes with a summary and suggestions for further research in Chapter 9.

List of publications Part of the work presented in this thesis has appeared in the following papers:

1. A. Pronobis and B. Caputo. The KTH-INDECS database. Technical Report 297, KTH, CVAP, 2005.

2. A. Pronobis and B. Caputo. The More you Learn, the Less you Store: Memory-Controlled Incremental SVM. In *Proceedings of the 9th European Conference on Computer Vision (ECCV06)*, Graz, Austria, 2006. (Submitted).

Chapter 2

Visual Indoor Place Recognition

The introductory Chapter 1 gave a brief overview of the place recognition problem and underlined the most important issues in designing place recognition systems. In this chapter we will study the subject in more detail. First in Section 2.1, we will try to define the objects of our interest - a place and a scene. We will study the properties of a coherent scene and present several constraints that have to be satisfied for a view to be called a scene. Such knowledge can be exploited in designing a place recognition system as well as psychophysical experiments on human perception. Then in Section 2.2, we will formulate the problem of visual indoor place recognition and define several key terms. In Section 2.3, we will present the structure of a typical pattern recognition system, based on which, we will describe the place recognition system being the subject of this thesis. Finally in Section 2.4, we will discuss how humans perceive scenes, and how the visual data is processed by human brain. This section constitutes a review of a selection of publications on the subject.

2.1 Places and Scenes

We start our considerations about place recognition with definitions of the terms scene and place. According to Henderson and Hollingworth [35], *scene* is a semantically coherent (and often nameable) human-scaled view of a real-world environment comprising background elements with multiple discrete objects arranged in a spatially licensed manner. Background is considered to consist of larger-scale immovable structures and surfaces, whereas objects are smaller-scale manipulable entities. This definition is, however, somewhat unspecific. First of all, it is difficult to precisely determine the boundaries of human scale. The distinction between the background and the objects is also arbitrary. A desk viewed from a distance can be regarded as an object; however, it may also constitute a background for such objects as a pen, a keyboard, or a cup. This problem is a consequence of the hierarchical properties of a scene. Finally, the objects must be arranged in a spatially licensed manner to form a coherent scene. Again we might ask: How to understand

Figure 2.1. Picture of an incoherent scene illustrating violations of the five relations introduced by Biederman [6]: *support* - the laptop is floating; *interposition* - the background appears through the briefcase; *probability* - the hydrant is mounted in the kitchen; *position* - the tap is mounted on a table; *size* - the chair appears to be larger than a table.

the word licensed? Does the picture presented in Figure 2.1 constitute a coherent scene? Can we define a set of rules that a scene must follow? An attempt to define several constraints, which must be satisfied for a scene to be regarded as coherent, was made by Biederman [6]. He introduced five relations which can be used to describe the difference between a well-formed scene and an array of unrelated objects. They can be easily explained using a graphic illustration of their violations (see Figure 2.1). The relations result from general physical (syntactic) constraints (*support* and *interposition*) as well as from the semantics and function of objects and the surrounding environment (*probability*, *position*, *size*).

We could try to formulate a definition of the term *place* on a similar basis. Places have a hierarchical structure as well. Building can be considered as one place; however, it may be decomposed into rooms such as offices or corridors which in turn may comprise several sections e.g. printing area or help desk. Consequently, it may be difficult to precisely determine the boundary between places. Different places may fulfill different functions, may have different appearance, or may just be separated by walls. As a result, a place can be regarded as a usually nameable segment of a real-world environment distinguished due to different functionality, appearance or artificial boundary.

Although both terms cannot be precisely defined, we may still draw several important conclusions. First, we may expect a relation between the type of a

place and objects that may occur within it. The fact that places usually differ with functionality is also an important cue. We may observe that the elements comprising the environment are related and must satisfy several constraints. The recognition systems must be able to cope with the hierarchical structure of places and scenes, as well as with situations when the boundary between places is not obvious, or the places only slightly differ in appearance.

2.2 Problem Statement

Visual indoor place recognition can be considered as a special case of *pattern recognition* constrained to visual representations of an indoor environment. To quote from [30], pattern is a quantitative or structural description of an object or some other entity of interest. According to this definition, a digital representation of a picture of a place, acquired from a particular view-point, under particular conditions, at a particular time, is also a pattern, as it constitutes a description of a nameable entity, in this case - a place. Usually, we can group patterns into *pattern classes* in respect of some common properties. These properties, also known as *features*, allow us to create *models* of the patterns in different classes. A model can be seen as a description of those features that are common within a class and different between classes. In case of place recognition, classes correspond to places, and we build models of places using their distinctive features, for example color.

Pattern recognition is a process aiming to assign class labels to sensed patterns based on the information contained in the models. The whole process should be automatic and should require as less human intervention as possible. The problem can be formulated as finding the value of the function $f : P \rightarrow \Re$ given the sensed pattern $\boldsymbol{p} \in P$ as an input argument. The value of the function $z \in \Re$ determines the membership of the pattern to one of the classes $\omega_1 \ldots \omega_c$, that is, if $z = i$, then the pattern belongs to class ω_i. At this point, it is important to note, that the recognition process can be divided into several stages (see Section 2.3), each of which is a field of research itself. Typically, before the classification algorithm (*classifier*) can be used, the distinctive features of the patterns need to be extracted.

Due to the complexity of the pattern recognition problem, it is usually not possible to guess the parameters of the models. Instead, we may try to find a representative set of samples (*training set*) and use it in order to train the classifier or even find the features that are best suited for the task at hand. Typically, in case of pattern recognition, the training samples are labeled, that is their membership to one of the classes is defined by a teacher before training (the training set can be denoted as $\{(\boldsymbol{p}_i, z_i) : \boldsymbol{p}_i \in P, z_i \in \langle 1; c \rangle\}_{i=1}^{n}$). Such approach is referred to as *supervised learning*, as opposed to *unsupervised learning* (also known as *clustering*), in which the groupings of the input patterns are found by the training algorithm itself. Supervised learning will be used to train the place recognition system described in this thesis.

During the training process, the parameters of the models are chosen in a manner

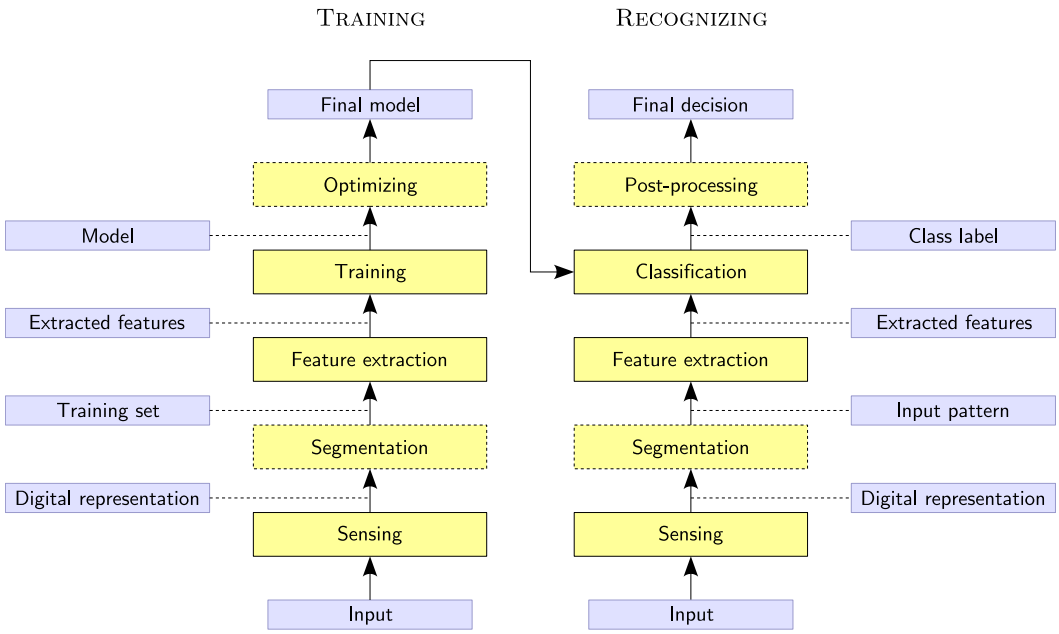


Figure 2.2. Structure and data flow of a typical pattern recognition system. The yellow rectangles represent components of the system, and correspond to operations being part of the training and recognition processes. Arrows show the direction of the data flow. Finally, the blue rectangles describe the type of data at every stage of the processes. The most fundamental operations, present in almost every pattern recognition system, are framed with a solid line.

allowing for reduction of the error on the training set. However, we hope that after training, the system will be able to properly recognize not only the training samples but also novel patterns encountered in the future. This ability is known under the name *generalization*, as the system is able to generalize its knowledge to classify patterns not available during training. Consequently, the training algorithm should not allow the situation when the classifier is adopted to the training set so well that it loses its generalization abilities. Such phenomenon is called *overfitting*.

The generalization performance is a crucial issue for place recognition systems. This is due to the fact that the environment and the conditions under which it is observed change continuously. Moreover, places can be viewed from multiple viewpoints. As a result, it is important to use such classification and learning techniques that provide good generalization abilities. The training set should be also carefully prepared in order to capture the variability of the environment.

2.3 Structure of a Typical Pattern Recognition System

In the previous section it was already mentioned that the pattern recognition process can be divided into several stages. In fact, the whole recognition system

can be seen as an assembly of specialized components cooperating by exchanging information. Figure 2.2 presents a structure of a typical pattern recognition system and shows how the data are passed between its components. The diagram was divided into two parts representing the training and recognition processes. Several components were marked with solid line, since they can be found in almost every pattern recognition system.

Below, we will study each component of the system in a separate section. We will also discuss how each part is implemented in the place recognition system presented in this thesis. Three first operations are common for both training and recognition processes and therefore will be described first.

2.3.1 Sensing

The wideness of the term *pattern* suggests how many kinds of stimuli can become an input of a pattern recognition system. In order to be processed, this information must be sensed and stored in a digital format. This is the task for various kinds of *sensors*. Although the construction and principle of operation of such sensors highly depend on the application, in most cases they can be regarded as analog to digital converters or digital measurement devices. Some examples are digital cameras, audio AD converters, laser scanners or sonar sensors. There are also applications which do not require specialized sensors because the input data is already in a digital format and can be directly processed by a feature extractor. Spam filtering is an example of such application.

In most cases, sensors can be characterized by several parameters. However the significance of these parameters depends on the problem and construction of the sensor. Some examples are: resolution, bandwidth and latency (especially important for real-time systems), sensitivity or signal-to-noise ratio (SNR). It is also desirable that the sensor is able to suppresses the within-class variability of the patterns (e.g. caused by changing illumination conditions) to the advantage of the between-class variability.

Two types of visual sensors are commonly used for visual place recognition. These are: a *regular digital camera* and an *omni-directional* (*catadioptric*) *camera* [51]. The regular camera is the most common visual sensor, and therefore a recognition system using it can be universally applied (wearable place recognition systems, topological localization for mobile robots, content-based image retrieval). The advantage of the omni-directional sensors is that they provide a horizontal field of view of 360° which simplifies the recognition task. However, cameras of this type are applied almost exclusively for mobile robot localization. See Figure 2.3 for the comparison of pictures acquired using cameras of both types. The visual indoor place recognition system presented in this thesis was tested on a database acquired using a regular camera. The database is described in detail in Chapter 3.

Figure 2.3. Comparison of images acquired in similar places using regular and omni-directional cameras (The picture acquired using the omni-directional camera was kindly provided by Patric Jensfelt).

2.3.2 Segmentation and Pre-processing

It is often the case that the pattern to be recognized is not isolated. Instead, it appears on some background or is partially overlapped by other elements. In such situation, each individual pattern needs to be segmented before proceeding with the recognition process. Another problem arises when the pattern consists of several disconnected parts. In such a case, the segmented parts must be properly combined in order to form a coherent entity. This operation is known as *grouping*.

Segmentation and *grouping* are challenging and complex problems in pattern recognition. They are crucial for such applications as optical character recognition, speech recognition or object recognition in real settings. The last example illustrates how difficult the problem is, since the objects must be segmented from a cluttered background. Segmentation is not necessary in case of visual place recognition systems. This is due to the fact that the whole picture acquired using a digital camera constitutes a single pattern.

Additional processing of the input data may be also required. This includes such operations as noise filtering or normalization.

2.3.3 Feature Extraction

The aim of the *feature extraction* is to provide a new representation of the input pattern, that would result in simplification of the classification problem. The new representation should be insensitive to the variability which can occur within a class (within-class variability), and should emphasize pattern properties that are different in different classes (between-class variability). In other words, it should consist of the values of *distinguishing features* of the patterns.

It is difficult to precisely define the division between feature extraction and classification. In ideal case, the feature extraction process would produce a representation making the classification problem trivial. In such case a separate classifier would be

unnecessary. On the other hand, we may imagine a classifier coming to a decision on the basis of the analysis of raw, unprocessed data. In practice, the algorithms usually cooperate and the boundary is defined by their capabilities. Moreover, the classifiers are usually more universal and can be used to solve various problems, while the feature extraction algorithm serves as an adaptation layer and is usually well-suited for one particular task.

The performance of the whole recognition system highly depends on the quality of the feature extractor. For this reason, it is crucial to properly identify the distinguishing features of the pattern. For example, in case of place recognition we will look for a descriptor that is invariant to translation, scale, as well as to variations in illumination and effects of small changes in the environment. The descriptor that performs best is usually chosen on the basis of experiments. The best features can be also selected automatically by the learning algorithm ([80]). All these issues in the context of visual place recognition are discussed in Chapter 4. The chapter also presents the local and global descriptors employed in the place recognition system.

2.3.4 Classifying

The *classifier* is the element of the recognition system which performs the actual recognition. Its task is to assign the input pattern, represented by the extracted features, to one of the classes. The complexity of the classification problem depends on how the feature values differ within each class in comparison to the differences between classes. This results from several factors. First, the task may be difficult itself. Place recognition is an example of a complicated problem due to its huge within-class variability. Then, the input data may be contaminated with noise. Finally, the complexity depends on the performance of the pre-processing operations such as segmentation and feature extraction. The quality of sensor also plays an important role as it may be more or less sensitive to unwanted variations (e.g. a camera may be equipped with automatic brightness control).

Other important issues related to the classification problem, such as generalization, were already discussed in Section 2.2. The visual indoor place recognition system presented in this thesis employs the Support Vector Machine classification algorithm. The algorithm is described in detail in Chapter 5.

2.3.5 Post-processing

The decision made by a single classifier does not have to be the final decision of a recognition system. In fact, in many cases, the performance and robustness of the system can be greatly improved by introducing additional mechanisms. These mechanisms can exploit other sources of information or just process the data provided by a single classification algorithm.

The pattern recognition system can make use of *a priori* constraints and information about the *context* in which the recognition occurs, or may exploit a history of previous results. The place recognition system mounted on a mobile robot can,

for instance, take into account that the robot only turned around and did not change its position. Consequently, the robot must be still in the same place and the decision can be made using the current and previous data. Such a system may also make use of the topological map of the environment. In such case, the robustness may be improved by verifying possible transitions between places (e.g. the robot cannot jump from the office to the kitchen without passing through the corridor).

The post-processing step is also required if the recognition system is built around several classifiers. Additionally, each classification algorithm may suggest several, most probable hypotheses. In such case, the classifiers can be regarded as experts specialized in different fields. The place recognition system may use several classifiers to classify pictures using more than one type of features. It may be also coupled with other algorithms such as object recognition.

In all presented cases, the final decision is made on the basis of *multiple cues*. Experiments show that such approach can cause a significant decrease in the recognition error (see e.g. [52]). First, additional information can be used to reduce the search space making the recognition more efficient. Then, it may be used in order to generate stronger cues and verify the result. In some cases this may lead to a correct result even if all the cues are separately wrong. Finally, it makes the system robust to the situations when not all cues are available. It is also motivated by the study on human perception, as human performance is strongly decreased if only one cue can be used (see e.g. [10]).

2.3.6 Training

The aim of the training, as well as the most important related issues, have been already discussed in Section 2.2. Here, we will describe how the process proceeds.

According to the *supervised learning* scheme, the training is performed using a selection of representative labeled patterns. The same procedure is followed during pattern acquisition as in case of recognition. The input data is first sensed, and the segmentation can be performed in order to isolate individual patterns. At this point, the isolated samples should be labeled by a teacher.

The set of patterns prepared in this way is now ready to be used for training the classifier as well as other parts of the system[1]. As it was already stated, it is important to not only minimize the error on the training samples, but also provide an ability to generalize to novel patterns. In order to achieve this goals the available samples can be divided into two subsets: the *training set* and the *test set*. The training algorithm will try to minimize the classification error on the training set, and the generalization performance will be evaluated using the test set. The division can be done several times, each time achieving different pair of subsets. This way the final performance is evaluated on the basis of several observations. Such method is known under the name *cross-validation*.

[1]The training process may be for example used to select the "best" subset of the distinguishing features of the patterns.

The simplest kind of cross-validation is *hold out cross-validation*. In this method the set of samples is simply divided into two subsets, one of which becomes the training set and the other is used for validation. Another method, called *K-fold cross-validation*, assumes that the samples are divided into K subsets. In that case, one of the subsets becomes the test set and the other subsets create the training set. Training and validation can be repeated K times for various combinations of the subsets. The value K may be equal to the number of available labeled patterns. In other words, the test set may consists of only one element, and the operation may be repeated for every pattern. Such variant is known as *leave-one-out cross-validation*, and is used mainly if the number of available samples is severely limited.

So far, we have considered training as a process which is performed before the recognition system is used. However, many applications could greatly benefit from the ability to update the knowledge of a pattern recognition system. Place recognition is an example of such application, as places may significantly change over time, and their models should be updated accordingly. This type of learning is known under the name of *incremental learning*, since it allows for updating the knowledge of the system incrementally without complete retraining.

2.3.7 Optimizing

Many applications require that the pattern recognition systems were not only robust and reliable but also efficient and low resource consuming. This is a crucial issue especially for systems aiming to work in real-time or performing continuous learning. In order to achieve these goals the employed algorithms can be improved, or additional optimization methods can be applied.

It is hard to indicate one part that is a bottleneck in every recognition system and should be optimized. Instead, the optimization method should be tailored to the particular algorithms. However, the position of this optimization operation in the diagram in Figure 2.2 is not accidental, and it refers to the optimization of the model for two reasons. First, the size of the model and the way it is stored may influence the efficiency of the pattern recognition system based on many types of classification algorithms. Second, it is definitely the case for Support Vector Machines which are employed in the place recognition system described in this thesis.

The *Support Vector Machine* classifier stores the model in the form of *support vectors* (a selection of training feature vectors) and corresponding weight coefficients (see Chapter 5 for a detailed description of SVMs). The time required to perform classification in case of SVMs is directly proportional to the number of support vectors. Storing a large number of support vectors may also require huge amounts of memory if the feature vectors are big. This is a problem for complex tasks such as visual place recognition. Such approach may also lead to reduction in the training time if the *incremental learning* scheme is used. As a result, several approaches were proposed ([11, 12, 57, 19]) for the optimization of the model produced by the standard training algorithm. Chapter 6 of this thesis describes the idea presented

by Downs *et al.* [19] and contributes to the method by introducing several extensions. Experiments presented in Chapter 7 prove that presented approach allows for significant decrease in the number of support vectors.

2.4 Human Perception of the Scene

Modern computer vision often exploits the knowledge of psychological and biological aspects of human (animal) perception and cognition. In fact, the research in such fields as pattern detection and recognition is held in parallel to psychophysical experiments with humans. This can be easily explained by the fact that in many tasks humans still far outperform the best available algorithms. Moreover, if we want to create machines that are able to interact with people and man-made environments, we need to learn the principles behind this interaction. As a result, numerous solutions, already used in computer vision and machine learning, are motivated by psychological and biological research. It is worth mentioning that the benefit in such case is mutual, as many problems can be better understood during implementation.

Humans are also extremely proficient in recognizing places on the basis of visual cues. For this reason, it is of particular interest to learn how do we process visual data, what do we see when we look at a scene, how do we represent the spatial knowledge, and finally how is it done so efficiently and robustly.

2.4.1 Scene Perception and Recognition

Numerous experiments report that humans are able to perceive and process scenes very rapidly. The task of recognizing a scene, determining its semantic category, extracting its general structural information, as well as recognizing some basic objects requires only one eye fixation and can be completed in less than 100 ms. This kind of information is usually referred to as the *gist* of a scene. Early experiments by Biederman *et al.* [7] indicate that during brief presentation of a scene enough information is acquired to affect the response of the participants to objects consistent and inconsistent with the scene. A different type of experiments conducted by Potter [59] show that subjects were able to detect a picture described by a label from a sequence of pictures presented at rates of 113 ms per picture. Schyns and Oliva [68] also report that subjects were able to properly recognize the type of scene (e.g. highway, city) after a very short presentation (150 or even 30 ms). Additionally, people seem to fixate the eyes on the most informative parts of the scene, which requires prior comprehension of the gist (see [34]).

The results of the experiments presented above show that the time required to recognize the scene is definitely too short to allow detailed analysis of all its elements. This raises the question of the features of a scene that people use during the first glance. Several hypotheses have been proposed. First, a *diagnostic object* (objects) could emerge and suggest a particular type of a scene ([26]). Another explanation given by Biederman [7] is that the gist is based on some scene-level

features. The hypothesis that global scene features are used in early recognition has been confirmed by the experiments conducted by Schyns and Oliva [68] in which they applied low-pass and high-pass filters to gray-scale images of scenes. The results of this operation were further combined in order to achieve images in which the low-frequency components originated from different scene than the high-frequency components. Consequently, the image could be perceived as one of two scenes depending on the spatial frequencies used to acquire the gist. Subjects were presented the images for the duration of 30 ms and in most cases were able to properly identify the scene represented by the *low-frequency components*, i.e. *coarse blobs*. The scene represented by high-frequency components was preferred if the images were presented for longer durations. Additional experiments [54] revealed that people are also able to use high frequencies for early recognition. More recent work by Oliva and Torralba [56] introduced a global scene representation called *spatial envelope*. Spatial envelope is a low-dimensional representation of the shape of a scene regarded as a single entity and encodes relations between its principal contours.

The color or texture may also be a cue facilitating scene recognition. Oliva and Schyns [55] present experiments in which participants were asked to name scenes presented in normal colors, abnormal colors and gray-scale. In cases when color was diagnostic of a scene category the presence of normal colors facilitated recognition, whereas the presence of abnormal colors made the task more difficult. No influence of color cues was found in cases when color was not diagnostic of the category. Walker Renninger and Malik [65] tried to investigate whether the early scene recognition can be explained with a texture recognition model. The authors compared results of experiments with humans to the performance of their texture recognition model and observed similar relationships.

The results of the experiments presented above indicate that coarse global information is used during the first glance at a scene and that details and local information are acquired later. Although people are able to extract high-frequency components of a scene very fast (see [54]), most scientists agree that during recognition the scene is decomposed, that is the more we look at a scene the more information we extract (see e.g. [50]).

Another interesting issue is whether natural scenes are processed separately from other stimuli such as faces or objects. This hypothesis is supported by the fact that there are numerous cases described in the literature of patients who were unable to recognize places although performed normally on test with objects and were able to understand spatial relationships between different points (see [22] and the references cited therein). The experiments conducted by Epstein and Kanwisher [24] proved that there exist a distinct area within human parahippocampal cortex which responds in *functional magnetic resonance imaging* (*fMRI*) to images of indoor or outdoor scenes as well as to landmarks ([22]). This region, named by the authors the *parahippocampal place area* (*PPA*), does not respond or responds weakly when the examined person views images of faces, objects, or other visual stimuli without spatial context. Additional experiments ([22]) demonstrated that PPA responds to

familiar and unfamiliar places in a similar way, and that it is activated stronger by novel than by repeated views. The precise function of PPA is still unclear. Current hypotheses postulate that it is either involved in scene perception or plays role in encoding of topographical information into memory or both. Together, the experiments suggest that there exist separate units in the brain dedicated for scene processing.

2.4.2 Scene Representation

In previous section we have described experiments conducted by Potter [59] demonstrating that people are able to detect and understand complex scenes presented in a sequence at rates of about 113 ms per picture. However, the same experiments revealed that at this rates the subjects were unable to remember the pictures and that additional delay is required for memory consolidation. Recent research suggests that people do not encode much of what they see (such conclusion can be drawn from the analysis of a phenomenon named *change blindness*, see [70]). This raises the questions of what information about a scene is encoded in the memory and how it is represented.

In order to answer the first question, Aginsky and Tarr [1] checked which of such visual properties of a scene as color, object position and object presence require special attention in order to be encoded and which of them are encoded automatically. Subjects were told to detect changes in images with and without cuing about the type of change. Changes always occurred in regions of marginal interest. The experiments demonstrated that cuing did not have influence on the detection time for position and presence changes, whereas the detection time was lower for color changes if the cue was provided. This suggests that properties that help to determine the spatial layout of a scene are better encoded than surface properties. Using similar method, Rensink *et al.* [66] found that detecting changes in items of marginal interest is much more difficult than in items of central interest, which are automatically encoded.

The problem of scene representation in the memory was also studied in order to discover whether it is viewpoint-specific or viewpoint-invariant. Experiments performed by Christou and Bülthoff [16] showed that immediately after training (passive or active), the representation is more likely viewpoint-specific. During training, the subjects were presented a computer model of an indoor environment from a limited set of directions. During test phase, the ability of the observers to recognize the same environment from unfamiliar directions was checked. The observers were able to recognize both familiar and novel views, however familiar views were recognized faster and more often. Similar conclusions have been drawn from experiments with fMRI and parahippocampal place area (PPA) [21]. However, additional analysis ([23]) suggests that the viewpoint-specific representation may evolve over time to become partially viewpoint-invariant.

2.4.3 Scene Context in Object Recognition

While discussing the definition of a coherent scene in Section 2.1, we have presented several relations introduced in [6] which occur between the elements of the scene and the scene itself. This raises the question of whether people make use of the fact that in the majority of cases real-world scenes are coherent with respect to these relations? Are such tasks as object recognition facilitated if the objects occur in a proper context? There is no simple answer to these questions.

Numerous psychological experiments have been conducted over the past three decades, and three competing models have been proposed by the scientists. The difference between the models concerns the stage of the object recognition process which the scene context is to influence. The *perceptual schema model* proposes that the information about scene can facilitate the perception of objects, that is, the object appearing in context is easier detected and its description is created more efficiently. The experiments performed by Biederman *et al.* [7] and Boyce *et al.* [9] indicate that the detection sensitivity is higher when the object is consistent with the meaning of the scene.

The *priming model* model proposes that scene context activates schema which facilitates the matching of the object description to the object representations in the long-term memory. The experiments conducted by Friedman [26] indicate that the eye fixation to the objects that are unexpected in the scene is longer which is claimed to be a consequence of the fact that such objects require more analysis of local visual details. More recent experiments performed by Bar and Ullman [4] confirm such hypothesis. They report that proper spatial relations between elements of the scene decreased response time and error rate during recognition. Additionally, occurrence of clearly recognizable objects facilitated the recognition of ambiguous objects.

Different model has been proposed by Hollingworth and Henderson [36]. They argue that the correlation between the existence of context and efficiency of object recognition might have been a result of methodological problems in each of the studies. Instead, they propose the *functional isolation model* in which object recognition is isolated from scene context. However, recent experiments performed by Auckland *et al.* [3] indicate that in spite of eliminating the methodological problems the influence of context is still visible. The scientists point to the *priming model* as an explanation.

2.5 Summary

In this Chapter, we discussed the main issues related to place recognition and scene perception with respect to both artificial recognition systems and humans. We started with defining the terms scene and place based on the literature on human scene perception. Then, we studied the place recognition problem as well as the key issues and terms related to designing and testing a place recognition system. We presented place recognition as a special case of pattern recognition and showed the complexity of this particular problem. Similarly, we used the structure of a typical

pattern recognition system in order to demonstrate the position and role of each part of our system within the recognition process. The parts will be described in detail in the next chapters. Finally, we presented a review of a selection of publications on human scene perception and recognition. The aim of this section was to present the current state of knowledge about the most robust and efficient place recognition system which becomes a motivation for many computer vision and machine learning researchers.

Chapter 3

The KTH-INDECS Database

The following chapter contains a detailed description of the *KTH-INDECS* database. The name INDECS is an acronym which stands for Indoor Environment under Changing conditionS. The database consists of several sets of pictures taken in five rooms of different functionality under various conditions.

The motivation for creating the database was the need for a flexible testing environment, which could be used to estimate the performance of the visual indoor place recognition system presented in this thesis. Robustness is a key property for a recognition system aiming to work in realistic settings, thus in building the database special emphasis was placed on capturing the variability of the environment and its intrinsic properties. For this purpose, pictures were taken under various illumination and weather conditions at different periods of time. Each room was observed from many viewpoints and angles (the need for multi-viewpoint representation during human passive learning was explained in Section 2.4.2). All this ensures that the response of the scene to the change of conditions was captured. Moreover, the normal activity in the rooms was recorded: people appear in the rooms, pieces of furniture are moved over time.

The database was acquired in a way that allows it to be used in experiments not only with indoor place recognition, but also with object recognition. The database may then be considered as a source of images containing objects in a cluttered scene. Thus, it is of potential interest not only for benchmarking of scene recognition systems, but also for context-based object recognition methods ([75, 49]) and visual attention algorithms.

The next sections provide information about the indoor environment in which the pictures were taken (Section 3.1), describe the acquisition procedure (Section 3.2), and present interesting examples illustrating several attributes of the database (Sections 3.3, 3.4, and 3.5). A summary is given in Section 3.6. Additional details such as precise maps of the environment can be found in the technical report [62]. The database is freely available on the Internet.

Barbara's office

Corridor

Elin's office

Kitchen

Surroundings of the printer

Figure 3.1. Pictures presenting the interior of each room.

3.1 Description of the Environment

The pictures included in the database were taken in the interior of The Computational Vision and Active Perception Laboratory. The environment consists of five rooms located on the same floor, performing various functions: the kitchen, the corridor, the surroundings of the printer (in fact a part of the corridor), and two offices (Barbara's office and Elin's office). Exemplary pictures of the rooms are shown in Figure 3.1.

A general map of the environment is presented in Figure 3.2. Boundaries between the rooms that were used in building the database are marked with dashed lines. Dotted lines were used to draw an outline of furniture. The map also contains positions of the points in which the pictures were taken. Several points were marked out and provided with arrows indicating the point and angle used to obtain the pictures in Figure 3.1. Detailed maps of the environment and coordinates of the points can be found in the technical report [62] describing the database.

As it was stated before, the rooms fulfill different functions that determine the activity that is likely to occur. Places like the corridor or the kitchen can be regarded as public, which implies that various people may be present and the furniture (e.g. chairs) is moved more often. On the other hand offices were photographed usually empty or with their owners at work.

The rooms are physically separated by a sliding glass door. However, the surroundings of the printer, which is a continuation of the corridor, is an exception to that rule, and was treated as a separate room due to its different functionality. In conclusion the exact border between the corridor and the surroundings of the

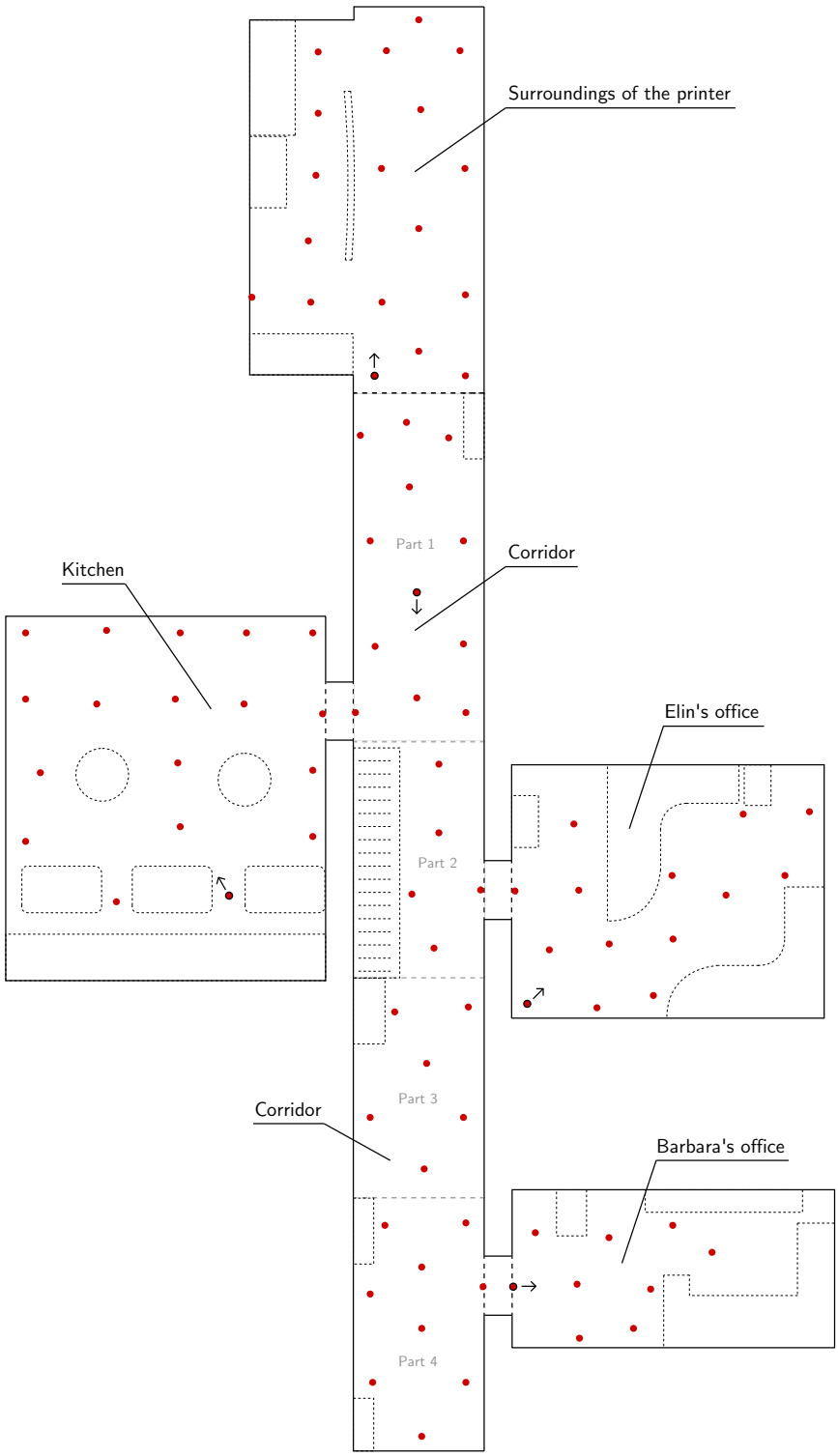


Figure 3.2. A general map of the environment.

Room	Number of markers	Total no. of pictures
Barbara's office	9	324
Corridor	32	1152
Elin's office	14	492
Kitchen	18	648
Surr. of the printer	18	648

Table 3.1. The number of markers and taken pictures for each room.

printer can be regarded as arbitrary.

The laboratory contains additional rooms which were not taken into consideration while creating the database. However, because of the glass door some of them can be visible in the pictures from the corridor.

3.2 Image Acquisition

The pictures were taken using an Olympus C-3030ZOOM digital camera mounted on a tripod. The height of the tripod was constant and equal to 76 cm (in order to imitate the perspective of a robot). All the images were acquired using the following camera settings:

- The resolution was set to 1024x768 pixels.
- No image compression was used.
- The flash was disabled.
- The zoom was set to the wide-angle mode.
- The auto-focus mode was enabled.

The tripod was always placed exactly over the markers on the floor, which were prepared in the beginning and kept in the same position during the whole acquisition process (the markers may be visible on the pictures as a small red and green dots on the floor). The markers were positioned approximately one meter from each other in areas accessible to people or a robot. The rough position of all markers is shown on the general map in Figure 3.2. Coordinates and detailed maps can be found in the technical report [62].

After the camera was placed above the marker, twelve pictures from twelve angles (every 30°) were taken. The camera was always turned clockwise, starting from the same direction. Each marker was assigned a unique number within the room. Table 3.1 contains the number of markers in each room, together with the total number of pictures of the room stored in the database.

The light used to illuminate the environment was not fixed or specially adjusted before the acquisition process, which means that the light suitable for the users

of the room was used. All the rooms have windows, however artificial light was sometimes used even during the day, especially in cloudy weather.

3.3 Observing Environment from Multiple Viewpoints and Angles

Due to the arrangement of the points where the camera was placed (markers), and the fact that twelve pictures from different angles were taken in each point, most parts of the environment were observed from multiple viewpoints. This ensures that the image of the scene was captured under illumination from various directions, and three-dimensional objects were viewed from several sides. The need for multi-viewpoint representation of the environment is also motivated by studies on human learning. Experiments show that people seem to encode the knowledge about places in a viewpoint-specific manner, and the viewpoint-invariant representation may evolve from the viewpoint-specific representation over time (see Section 2.4.2). Two exemplary sets of pictures are shown in Figures 3.3 and 3.4.

3.4 Capturing Variability of the Environment

The database comprises pictures taken over a span of two months, under three different outdoor illumination and weather conditions: in cloudy weather, in sunny weather, and at night. As a result, the database always contains three pictures obtained from exactly the same viewpoint[1]. The following variations may be noticed in the pictures:

- The illumination conditions change because of variation in the outdoor and artificial light.
- Objects are moved and new objects appear.
- Furniture is moved.
- People appear in the rooms.

The most significant changes are caused by the illumination, as it can be observed especially in the pictures taken in front of a window on sunny days. The camera is equipped with an automatic exposure system, which, in this case, causes the pictures to darken.

Examples of scenes photographed under different weather and illumination conditions are presented in Figure 3.5.

[1]The pictures taken at one of the points in Elin's office in the sunny weather are missing.

Figure 3.3. One part of the environment observed from various viewpoints and angles.

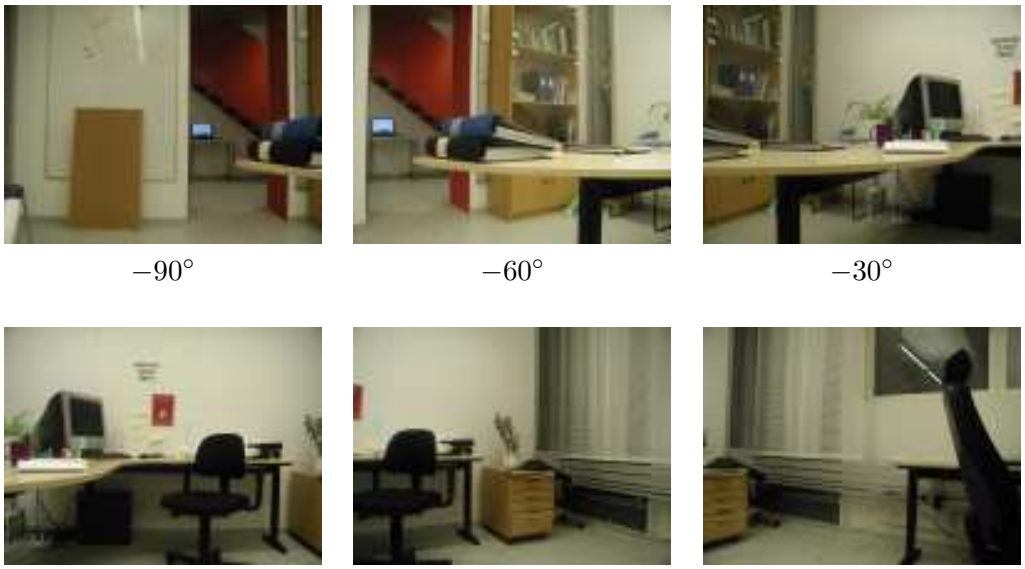


Figure 3.4. Pictures taken from the same viewpoint and several angles.

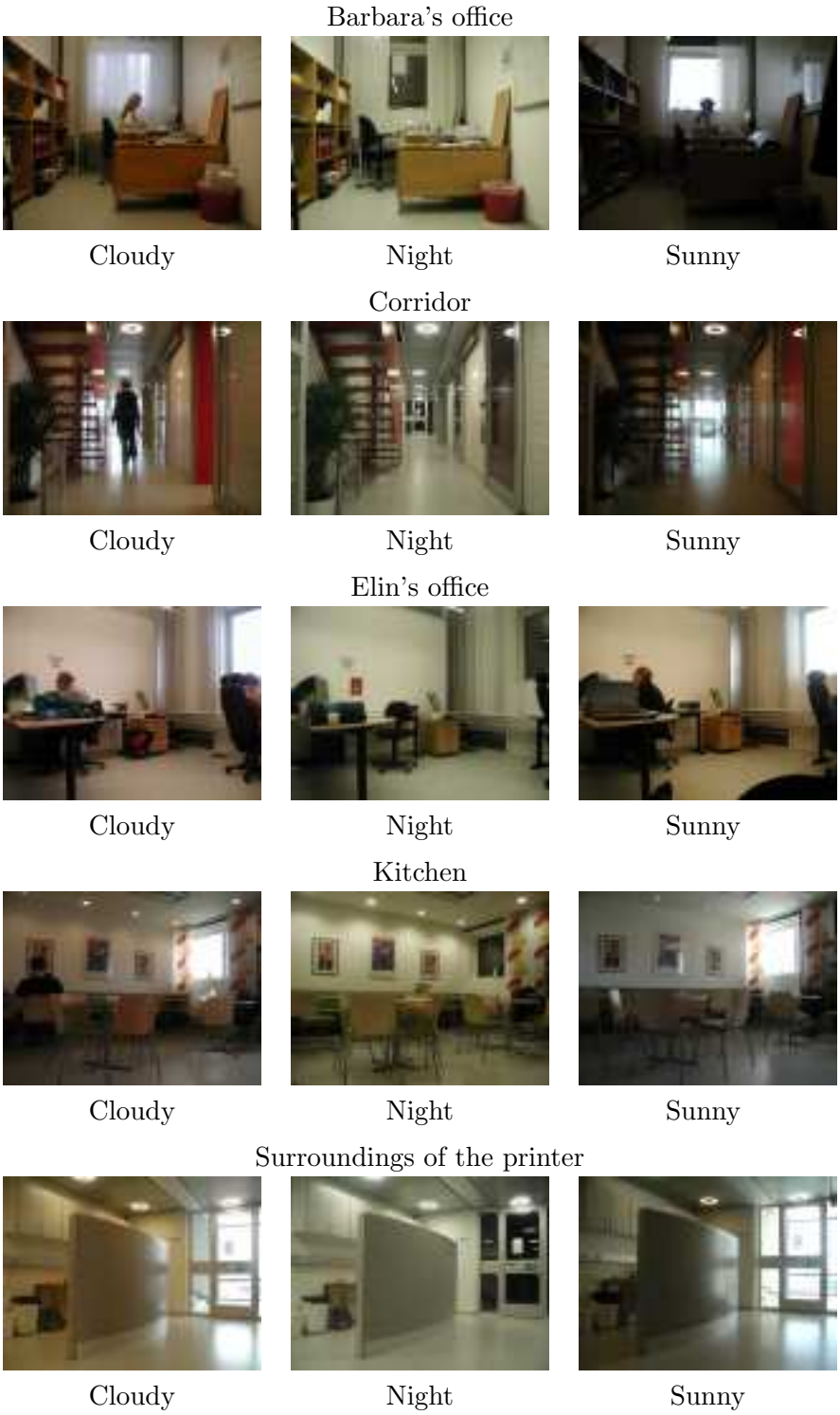


Figure 3.5. Examples of pictures taken under three different illumination and weather conditions for each of the five rooms.

Figure 3.6. Examples of non-informative pictures.

Figure 3.7. Examples of pictures taken near the edge of the corridor.

3.5 Difficult Examples

The database was created according to the assumption that pictures should be taken in the marked points from every of the twelve angles, irrespective of the contents of the picture. In result, several pictures can be regarded as non-informative, since they contain very little clues about the place where they were taken (e.g. pictures of blank walls). Figure 3.6 gives examples of non-informative pictures in the database.

Another difficulty that a place recognition system may encounter is caused by relatively narrow angle of view of the digital camera. The problem is observable especially near the edge of the room, because some pictures contain information coming only form the adjoining room. An example can be found in Figure 3.7, which contains pictures taken near the edge of the corridor.

3.6 Summary

This Chapter provided a description of a database of pictures of five places within an indoor environment. The database captures the variability of the environment as each place was photographed multiple times under various conditions and from multiple view-points. The database was created in order to evaluate the performance of the place recognition system presented in this thesis, however due to high quality of the pictures it may be used by any other place or object recognition system. Additionally, it is freely available on the Internet.

Several extensions to the database are planned in the future. Currently, the

database contains pictures taken on one floor of the laboratory under three weather and illumination conditions. However, it would be of interest to extend the database to other floors (which are similar) and acquire more data in the points used so far.

As it was mentioned before in Section 3.5, the pictures are of different significance in respect of the amount of the position information they contain. The database may therefore be filtered to exclude the non-informative pictures. To avoid using any prior knowledge, the number of corners detected in the image may be used as a measure of significance. Future work will also concentrate on recreating the database using a camera mounted on a robot.

Chapter 4

Feature Extraction

In this chapter, we will provide a detailed description of the methods employed in our visual place recognition system for extracting the *distinguishing features* from the pictures of places. As it was already stated in Chapter 2, the *feature extraction* process aims to provide a new representation of the input data that is less sensitive to the within-class variability and emphasizes the differences that occur between classes. In case of visual data, such representation can be derived from the whole image or can be computed locally based on its salient parts. We speak then of *global* or *local features* respectively.

Most of the currently available approaches to the place recognition and visual-based topological localization make use of global features and a holistic representation of the input images. Consequently, several approaches have been proposed employing different global descriptors. Color histograms [77, 8], eigenspace representations [27], Fourier coefficients of low frequency image components [46], or vectors of principal components of filter bank outputs [76, 75, 49] are just a few examples. Local image features have also been tried on place recognition and localization. Several authors propose: reading graphical landmarks containing text and icons [44, 45] or using the Kernel PCA algorithm [73] or the SIFT descriptor [69, 2] for local feature extraction.

In this thesis, we evaluate the performance of both global and local image descriptors for place recognition. Our place recognition system employs the Composed Receptive Field Histograms [38] as global features and the SIFT descriptor [41, 42] in order to extract the features from local patches. In the second case, the Harris-Laplace detector [47] is used to detect the interest points. The algorithms are described in Sections 4.2 and 4.3. Two more image descriptors were used to extract features from the pictures of textures in the experiments with Support Vector Reduction presented in Chapter 7: *MR8* [79] and *Local Binary Patterns* (*LBP*) [53]. Due to the fact that they are texture descriptors, and that material categorization is not directly related to the subject of this thesis, they will not be described in more detail. Interested reader is directed to the references given above as well as to [13] and the references cited therein.

Before presenting the details of the image descriptors mentioned above, we will introduce, in Section 4.1, the common theoretical framework.

4.1 Theoretical Framework

We will start our considerations about feature extraction with introducing theoretical framework required to explain both local and global image descriptors employed in our place recognition system. In Section 4.1.1 we will present a brief description of scale-space theory - a framework for handling images at multiple scales. Next, in Section 4.1.2, we will describe several basic image operators used at an early stage of the feature extraction process.

4.1.1 Scale-Space Theory

While defining the terms scene and place in Section 2.1 we used the adjective *human-scaled*. We also said that places and scenes have a hierarchical structure. If we look at a building from a distance, we perceive it as a single entity. However, if we take a closer look through the window, we will see a room containing many objects such as tables or chairs appearing on a background of the walls and the floor. If we looked even closer we would notice that there are several pens and a cup on the table. We could now take a picture of the table from a distance of a few centimeters or even use a microscope to analyze its surface. Would the results still image a table, or maybe we would rather speak of the material the table is made of or even molecules? This simple example illustrates an inherent property of all real-world objects - they are meaningful entities only at a certain range of scales. Consequently, they are perceived differently depending on the scale.

Scale-space theory is a framework for analyzing the images (or any other signal) at multiple scales. Since no *a priori* information about the scale is usually available it becomes necessary to create a multi-scale representation on the basis of the original image, in which the fine-scale structures are successively suppressed as the scale increases. The finest scale is represented by the original image and is influenced by the parameters of the sensor (e.g. resolution). This idea is illustrated in Figure 4.1.

The problem has been formulated in variety of ways (see [39]), and several constraints (*scale-space axioms*) have been introduced regarding the transformation used to derive the images at a given scale from the original image. These are mainly linearity and shift invariance as well as a rule that new structures should not be created as a result of the transformation. Witkin [84] proposed using a convolution with Gaussian kernel of increasing variance in order to smooth the images. Such approach is consistent with the scale-space axioms and is motivated by the results of neuropsychological research (see e.g. [85]).

Formally, the scale-space representation $L : \Re^D \times \Re_+ \rightarrow \Re$ constructed from a signal $s : \Re^D \rightarrow \Re$ is given by

$$L(\boldsymbol{x}, t) = g(\boldsymbol{x}, t) * s(\boldsymbol{x}) \tag{4.1}$$

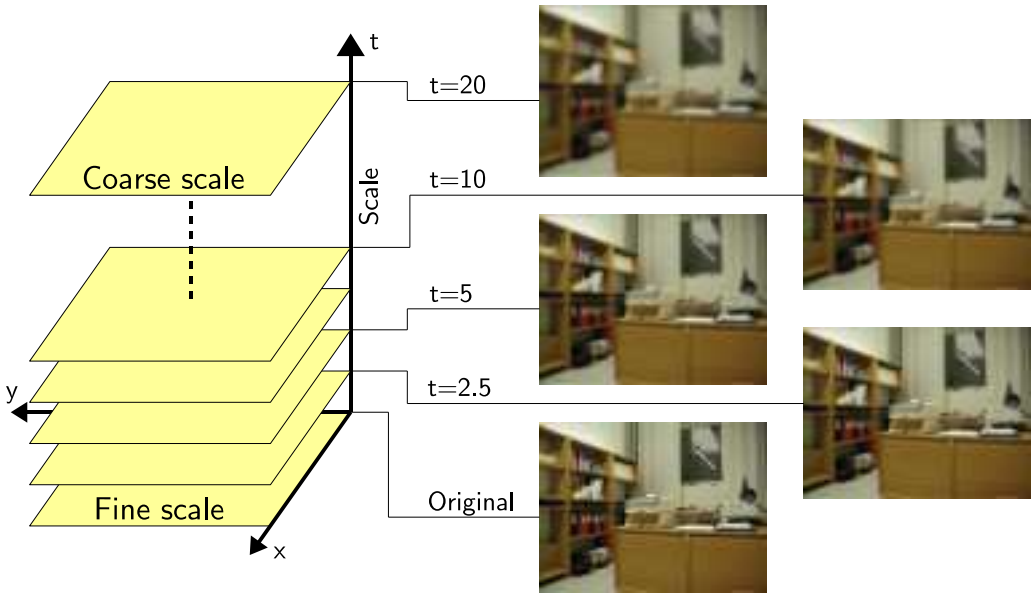


Figure 4.1. Multi-scale representation derived from the original image.

and

$$L(\boldsymbol{x}, 0) = s(\boldsymbol{x}), \tag{4.2}$$

where t is the scale parameter, $*$ denotes the convolution operation, $\boldsymbol{x} \in \Re^D$, and $g : \Re^D \times \Re_+ \setminus \{0\} \rightarrow \Re$ is the Gaussian kernel given by

$$g(\boldsymbol{x}, t) = \frac{1}{(2\pi t)^{\frac{D}{2}}} e^{-\frac{\boldsymbol{x}^T \boldsymbol{x}}{2t}} = \frac{1}{(2\pi t)^{\frac{D}{2}}} e^{-\frac{\sum_{i=1}^{D} x_i^2}{2t}}. \tag{4.3}$$

The scale parameter t can be expressed in terms of the variance of the Gaussian kernel σ: $t = \sigma^2$.

The interpretation of the scale-space can be obtained by solving the diffusion equation describing the heat distribution over time t in a homogeneous medium with uniform conductivity. The diffusion equation yields

$$\frac{\partial L}{\partial t} = \frac{1}{2} \nabla^2 L \tag{4.4}$$

with initial condition

$$L(\boldsymbol{x}, 0) = s(\boldsymbol{x}). \tag{4.5}$$

Now, consider that the scale-space representation is created based on an image $s : \Re^2 \rightarrow \Re$, which is a two-dimensional signal. In such case, Eq. 4.1 can be written as

$$L(x, y, t) = \frac{1}{(2\pi t)^{\frac{D}{2}}} e^{-\frac{x^2+y^2}{2t}} * s(x, y). \tag{4.6}$$

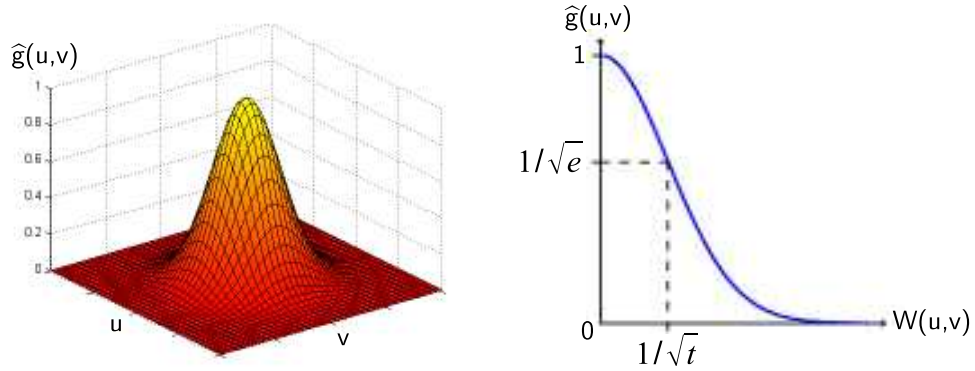


Figure 4.2. Impulse response of the Gaussian filter in the Fourier domain.

The convolution with the Gaussian kernel can be also expressed as a multiplication in the Fourier domain:

$$\widehat{L}(u,v,t) = e^{-\frac{t\,(u^2+v^2)}{2}} \cdot \widehat{s}(u,v) = e^{-\frac{t\,W(u,v)^2}{2}} \cdot \widehat{s}(u,v), \tag{4.7}$$

where $W(u,v) = \sqrt{u^2+v^2}$. This illustrates that convolving with the Gaussian kernel is in fact filtering with the low-pass Gaussian filter and that the cut-off frequency of the filter depends on the parameter t. The impulse response of the filter in the Fourier domain is shown in Figure 4.2. It is important to notice, that the Fourier transform of the Gaussian kernel is still a Gaussian function, and its width decreases as t grows.

By filtering with the Gaussian filter, we lose information about the high-frequency components. This corresponds to the blurring operation and simulates an effect of a viewer moving away from the image. However, we may think of the higher-frequency components as of noise contaminating the structures existing at higher scales. Thus, the filtering operation allows us to analyze the structures characteristic for certain scale. Additionally, the Gaussian filter suppresses the real noise which may occur in the image.

4.1.2 Basic Image Operators

Computing more sophisticated image descriptors such as Composed Receptive Field Histograms or SIFT requires applying several low-level operators to the analyzed image. These are mainly derivatives and operators that can be expressed in terms of derivatives e.g the Laplacian, the gradient, or the determinant of Hessian. Additionally, the computations need to be performed at a certain scale in the scale-space.

Let us consider each operator in turn.

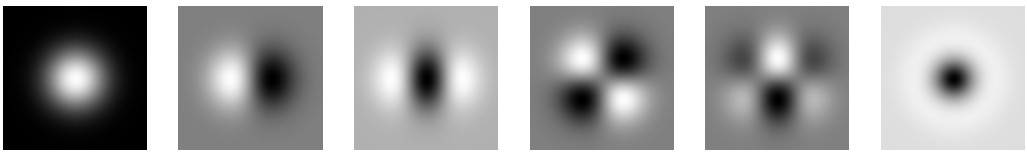


Figure 4.3. Two-dimensional kernels for the Gaussian, Gaussian derivatives and the Laplacian.

Gaussian derivatives In accordance with Eq. 4.1, the derivative computed from the scale-space representation at a certain scale t can be denoted as:

$$L_{x_1^{d_1} \cdots x_D^{d_D}}(\boldsymbol{x}, t) = \frac{\partial^{d_1}}{\partial x_1^{d_1}} \cdots \frac{\partial^{d_D}}{\partial x_D^{d_D}} L(\boldsymbol{x}, t) = \frac{\partial^{d_1}}{\partial x_1^{d_1}} \cdots \frac{\partial^{d_D}}{\partial x_D^{d_D}} \left(g(\boldsymbol{x}, t) * s(\boldsymbol{x}) \right). \quad (4.8)$$

Since differentiation is linear and shift invariant, there exists a kernel $K_\partial(\boldsymbol{x})$ that can be used in order to compute the derivative using convolution:

$$L_{x_1^{d_1} \cdots x_D^{d_D}}(\boldsymbol{x}, t) = K_\partial(\boldsymbol{x}) * L(\boldsymbol{x}, t) = K_\partial(\boldsymbol{x}) * g(\boldsymbol{x}, t) * s(\boldsymbol{x}). \quad (4.9)$$

As the convolution is commutative

$$\begin{aligned} L_{x_1^{d_1} \cdots x_D^{d_D}}(\boldsymbol{x}, t) &= \underbrace{\left[\frac{\partial^{d_1}}{\partial x_1^{d_1}} \cdots \frac{\partial^{d_D}}{\partial x_D^{d_D}} g(\boldsymbol{x}, t) \right]}_{\text{Gaussian deriv. } g_{x_1^{d_1} \cdots x_D^{d_D}}} * s(\boldsymbol{x}) \\ &= g(\boldsymbol{x}, t) * \left[\frac{\partial^{d_1}}{\partial x_1^{d_1}} \cdots \frac{\partial^{d_D}}{\partial x_D^{d_D}} s(\boldsymbol{x}) \right]. \end{aligned} \quad (4.10)$$

We see from the Eq. 4.8 and 4.10 that the scale-space derivative can be computed in three different ways. It is important to notice that it is possible to prepare a kernel for the *derivative of Gaussian* and convolve it with the signal in order to obtain its derivative at certain scale. Examples of 2D kernels for the derivatives of Gaussian up to third order are presented in Figure 4.3.

In practice, the kernel $K_\partial(\boldsymbol{x})$ is approximated e.g. using Taylor Series. The most commonly used discrete derivative approximations are

- non-central operator
$$\frac{\partial}{\partial x_i} : \begin{bmatrix} 1 & -1 \end{bmatrix} \quad (4.11)$$
- central operator
$$\frac{\partial}{\partial x_i} : \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad (4.12)$$

for the first order derivative and

$$\frac{\partial^2}{\partial x_i^2} : \begin{bmatrix} 1 & -1 \end{bmatrix} * \begin{bmatrix} 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \tag{4.13}$$

for the second order derivative. Kernels for higher order derivatives can be computed by convolving the operators presented above.

One of the properties of the differentiation operation is that it amplifies high frequencies ($\mathcal{F}\left\{\frac{\partial f}{\partial x_1}\right\} = i\omega_1 \mathcal{F}\{f\}$). For this reason it also emphasizes the high-frequency noise which appears in the image. Due to its low-pass characteristics, the Gaussian filter has the ability to suppress high-frequency noise, which makes the Gaussian derivatives more stable. However, another consequence of this fact is that the amplitude of the derivatives decrease with scale. This is a problem for applications in which comparing derivatives of the scale-space representation is essential. As a solution, Lindeberg [40] introduced the *normalized derivative operator*:

$$\frac{\partial^d}{\partial \xi_i^d} = t^{\frac{\gamma d}{2}} \frac{\partial^d}{\partial x_i^d}, \tag{4.14}$$

where γ is a free parameter to be tuned to the task at hand (if $\gamma = 1$, the derivatives are perfect scale-invariant). The definition presented in Eq. 4.14 constitutes a basis for the *automatic scale selection* method presented by Lindeberg [40].

In this thesis we are interested in analyzing images, which are two-dimensional signals. For this reason, we will restrict ourselves to the two-dimensional representation of the scale-space derivative given by

$$L_{x^{d_x} y^{d_y}}(x, y, t) = \frac{\partial^{d_x}}{\partial x^{d_x}} \frac{\partial^{d_y}}{\partial y^{d_y}} g(x, y, t) * s(x, y)$$

Gradient The *gradient* is a linear, first-order differential vector operator defined as follows:

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_D} \end{bmatrix}^T. \tag{4.15}$$

The gradient can be computed from the scale-space representation, and in the two-dimensional case yields:

$$(\nabla L)(x, y, t) = \begin{bmatrix} L_x(x, y, t) & L_y(x, y, t) \end{bmatrix}^T. \tag{4.16}$$

It is widely used in edge detection as it provides information about the direction and strength of edges in the image. The gradient magnitude

$$|\nabla L|(x, y, t) = \sqrt{L_x^2(x, y, t) + L_y^2(x, y, t)} \tag{4.17}$$

is an isotropic non-linear operator which gives a measure of the amount of difference between neighboring pixels. The direction of the greatest difference can be obtained by calculating the gradient orientation

$$\phi(\nabla L)(x, y, t) = \tan^{-1} \frac{L_y(x, y, t)}{L_x(x, y, t)}. \tag{4.18}$$

Hessian The *Hessian* is a matrix of second derivatives defined as follows:

$$\nabla\nabla^T = \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2}{\partial x_1 \partial x_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_D \partial x_1} & \cdots & \frac{\partial^2}{\partial x_D \partial x_D} \end{bmatrix}. \tag{4.19}$$

The operator $trace(\nabla\nabla^T)$ is known as *Laplacian*, and $det(\nabla\nabla^T)$ is a non-linear rotationally invariant operator. The *determinant of the Hessian* applied to the scale-space representation of an image can be denoted as

$$det(\nabla\nabla^T L)(x, y, t) = L_{xx}(x, y, t)L_{yy}(x, y, t) - L^2_{xy}(x, y, t). \tag{4.20}$$

Normalized trace of the Hessian and normalized determinant of the Hessian are commonly used for *automatic scale selection*.

Laplacian The *Laplacian* is a linear, isotropic, second-order differential operator defined as

$$\nabla^2 = \sum_{i=1}^{D} \frac{\partial^2}{\partial x_i^2} \tag{4.21}$$

Again, it can be computed from the scale-space representation of an image as follows:

$$(\nabla^2 L)(x, y, t) = L_{xx}(x, y, t) + L_{yy}(x, y, t). \tag{4.22}$$

Since the convolution is associative, it is possible to combine the Gaussian kernel with the Laplacian operator:

$$(\nabla^2 L)(x, y, t) = \underbrace{\left[\frac{\partial^2}{\partial x^2} g(x, y, t) + \frac{\partial^2}{\partial y^2} g(x, y, t)\right]}_{\text{Laplacian of Gaussian}} * s(x, y) = (\nabla^2 g)(x, y, t) * s(x, y) \tag{4.23}$$

The resulting operator is referred to as the *Laplacian of Gaussian* and its zero-crossings are widely used as an edge detector.

4.2 Global Features - Composed Receptive Field Histograms

The *Composed Receptive Fields Histograms of Higher Dimensionality* (*CRFH*) [38] have been used as global features during the experiments presented in this thesis. This multi-dimensional histogram representation has been so far applied mainly for object recognition problems (see e.g. [38, 67]); however, in Chapter 8 we will show that it performs very well also for place recognition.

CRFH is a multi-dimensional statistical representation of the occurrence of responses of several image descriptors applied to the image. This idea is illustrated

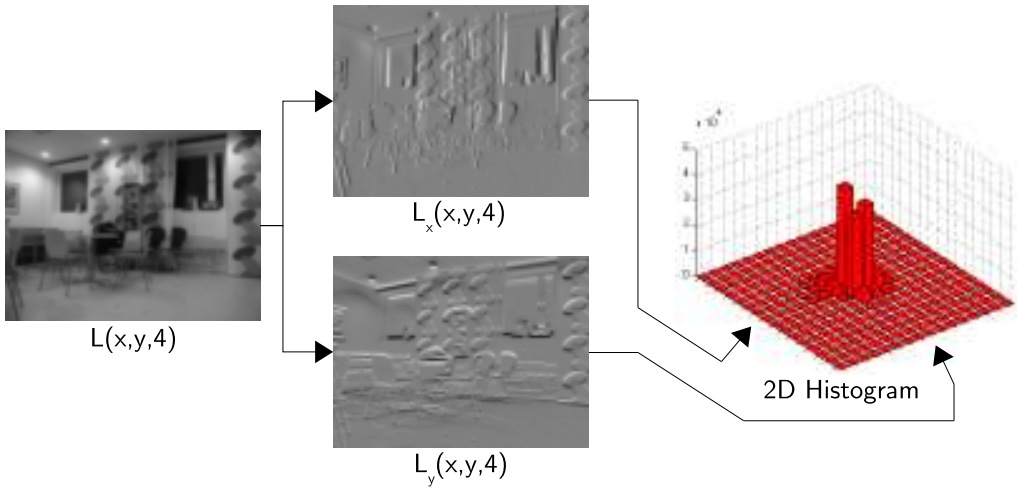


Figure 4.4. The process of generating multi-dimensional receptive field histograms shown on the example of the first-order derivatives computed at the same scale $t = 4$.

in Figure 4.4. Each dimension corresponds to one descriptor and the cells of the histogram count the pixels sharing similar responses of all descriptors. Such approach allows to capture various properties of the image as well as relations that occur between them.

Multi-dimensional histograms can be extremely memory consuming and computationally expensive if the number of dimensions grows. For example, a 9-dimensional histogram with 16 quantization levels per dimension contains approximately $7 \cdot 10^{10}$ cells. In [38] Linde and Lindeberg suggest to exploit the fact that most of the cells are usually empty and store only those that are non-zero. Such approach is consistent with the sparse format used by the *libSVM library* [14] – the implementation of the SVM classifier used in experiments in Chapter 8. The histogram can be stored as an array $[c_1, v_1, c_2, v_2, \ldots, c_n, v_n]$, where c_i denotes the index of the cell containing the non-zero value v_i. The index c_i for a D-dimensional histogram with the quantization levels $q_1, \ldots, q_D$ can be computed as follows:

$$c_i = \sum_{k=1}^{D} \left(m_{ik} \prod_{j=1}^{k-1} q_j \right), \tag{4.24}$$

where the values $m_{i1}, \ldots, m_{iD}$, $0 \leq m_{ik} < q_k$ denote the coordinates of the cell. Such representation of the histogram allows not only to reduce the amount of required memory, but also to perform such operations as histogram accumulation and comparison in an efficient way.

In the experiments presented in Chapter 8 we built multi-dimensional histograms using combinations of the following image descriptors applied to the scale-space representation at various scales:

- Intensity L
- R, G, B color channels
- Chromatic cues:
  $C_1 = \frac{R-G}{2}$ and $C_2 = \frac{R+G}{2} - B$
- First-order, normalized Gaussian derivatives:
  $L_{x,norm} = \sqrt{t}L_x$ and $L_{y,norm} = \sqrt{t}L_y$,
- Second-order, normalized Gaussian derivatives:
  $L_{xx,norm} = tL_{xx}$, $L_{yy,norm} = tL_{yy}$, and $L_{xy,norm} = tL_{xy}$
- Normalized gradient magnitude
  $|\nabla_{norm} L| = \sqrt{t(L_x^2 + L_y^2)}$
- Normalized Laplacian
  $\nabla^2_{norm} L = t(L_{xx} + L_{yy})$
- Normalized determinant of the Hessian
  $det(\nabla_{norm} \nabla^T_{norm} L) = t^2(L_{xx}L_{yy} - L^2_{xy})$

4.3 Local Features

The idea behind *local features* is to represent the appearance of the image only around a set of characteristic points known as the *interest points*. In order to determine the resemblance between two images using such representation, the local descriptors from both images are matched. Consequently, the degree of resemblance is usually a function of the number of properly matched descriptors. Local features are known to be robust to occlusions, as the fact that some of the interest points are not available do not affect the features extracted from other local patches.

The process of local features extraction consists of two stages: *interest point detection* and *description*. The interest point detector aims to identify a set of characteristic points in the image that could be re-detected even in spite of various transformations (e.g. rotation and scaling) and variations in illumination conditions. The role of the descriptor is to extract robust features from the local patches located at the detected points.

The place recognition system presented in this thesis employs the Harris-Laplace detector [47], described in Section 4.3.1, and the SIFT descriptor [41, 42], discussed in Section 4.3.2. Comparisons of local descriptors and interest point detectors conducted by Mikolajczyk and Schmid [48] show that both algorithms are highly reliable. Moreover, the SIFT descriptor was shown to perform well for object classification ([18]) and geometric mobile robot localization ([69, 2]).

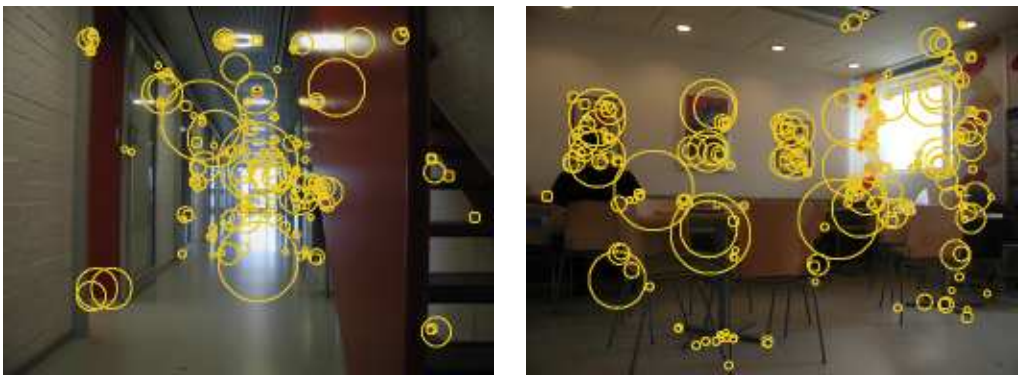


Figure 4.5. Interest points detected using the Harris-Laplace detector. The radius of the circles illustrate the scale at which the points were detected.

4.3.1 Harris-Laplace Interest Point Detector

The *Harris-Laplace detector* [47] is a scale, rotation, and translation (and partially affine) invariant interest point detection algorithm that was also shown to be robust to illumination changes ([48]). The algorithm employs the scale-adopted *Harris detector* [32] in order to identify the interest points in the scale-space and the Laplacian measure for automatic scale selection. Examples of points detected by the Harris-Laplace detector are shown in Figure 4.5.

The original Harris function is based on the second moment matrix. It has shown a good performance in presence of image rotations, illumination changes and perspective deformations; however, it is sensitive to variations in the image resolution. In order to cope with this problem, the scale-adopted second moment matrix is used:

$$\mu(x, y, t_I, t_D) = t_D g(x, y, t_I) * \begin{bmatrix} L_x^2(x, y, t_D) & L_x L_y(x, y, t_D) \\ L_x L_y(x, y, t_D) & L_y^2(x, y, t_D) \end{bmatrix}, \tag{4.25}$$

where $t_I = \sigma_I^2$ is the integration scale, and $t_D = \sigma_D^2$ is the differentiation scale. The Gaussian derivatives are computed at the scale t_D and the result is smoothed with the Gaussian window $g(x, y, t_I)$ of width σ_I as defined in Eq. 4.3. The scale parameters are usually related by the equation $\sigma_D = s\sigma_I$, where s is a constant factor.

The scale-adopted Harris interest function yields

$$det(\mu)(x, y, t_I, t_D) - \alpha(trace(\mu)(x, y, t_I, t_D))^2. \tag{4.26}$$

The function can be seen as a measure of cornerness at the point (x, y) and the scale t_D. As a result, the interest points corresponding to corners can be detected by finding its local maxima.

In order to locate the interest points in the scale-space, the Harris-Laplace detector computes the Harris function at multiple scales and searches for local maxima.

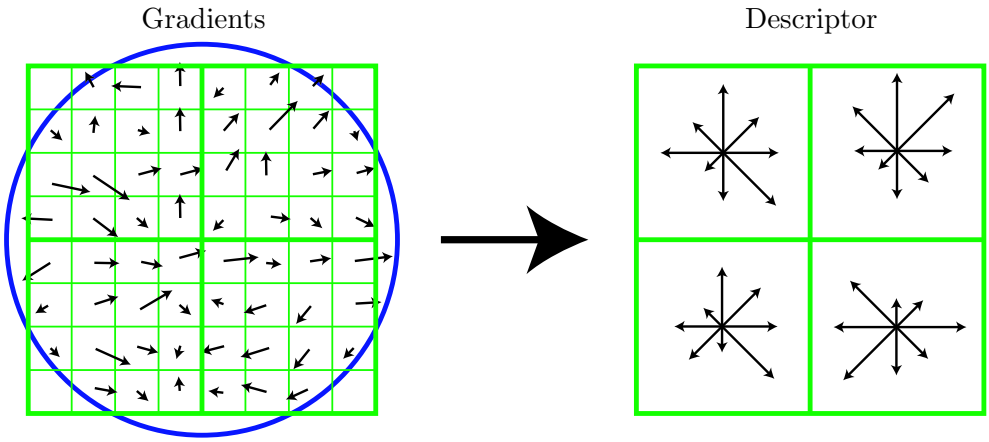


Figure 4.6. The SIFT local descriptor consisting of a 2×2 array of histograms (from Lowe [42]).

The maxima are thresholded in order to reject those of small cornerness. Finally, the algorithm checks whether the detected points correspond to the extremum of the Laplacian of Gaussian computed over scale. Those points for which the LoG attains no maximum and for which it is below some threshold are rejected. Consequently, according to the *automatic scale selection* theory [40], the interest points always correspond to the characteristic scale determined by the extremum of the Laplacian measure.

4.3.2 SIFT Descriptor

The *Scale-Invariant Feature Transform* (*SIFT*) descriptor invented by Lowe [41, 42] represents the features of local patches characterized by coordinates in the scale-space in the form of histograms of gradient directions. The gradient magnitudes and gradient directions used for further computations are obtained using pixel differences according to the equations

$$|\nabla L|(x,y,t) = \sqrt{(L(x+1,y,t) - L(x-1,y,t))^2 + (L(x,y+1,t) - L(x,y-1,t))^2}$$

$$\phi(\nabla L)(x,y,t) = tan^{-1}\left(\frac{L(x,y+1,t) - L(x,y-1,t)}{L(x+1,y,t) - L(x-1,y,t)}\right), \tag{4.27}$$

where $L(x,y,t)$ is the scale-space representation of an image and t denotes the scale at which the interest point was detected.

The first step of the algorithm is to assign a characteristic orientation to the interest point. This is done by detecting peaks in a gradient direction histogram. The histogram is computed in such way that the contribution of each pixel belonging to the local patch is weighted by the gradient magnitude and a Gaussian window of width equal to 1.5 times the scale of the interest point. Finally, the highest peak in the histogram determines the orientation of the point. Additionally, any

other peak of height of at least 80% of the highest peak is used to generate a new interest point. All further measurements are stored relatively to this orientation which provides invariance to rotation.

As it was already mentioned, the local descriptor consists of a set of gradient direction histograms. The histograms are computed on a 4×4 neighborhood using the same procedure as in case of orientation detection. Figure 4.6 illustrates the process for a 2×2 descriptor array. In practice, the best results are obtained for a 4×4 array of histograms with 8 bins in each. Consequently, the local descriptor consists of a vector of $4 \times 4 \times 8 = 128$ elements.

In order to increase the robustness to illumination variations, the feature vector is post-processed. It is normalized and large values in the vector are rejected by thresholding. Such approach is motivated by the fact that illumination changes are more likely to largely influence the relative magnitude of some gradients than the gradient orientation.

4.4 Summary

In this chapter, we described two image descriptors used in our experiments presented further: global - Composed Receptive Field Histograms, and local based on the Harris-Laplace interest point detector and SIFT descriptor. The same theoretical background is required in order to explain the algorithms. In view of this fact, we first presented the fundamentals of the scale-space theory, and several basic image operators commonly used during the feature extraction process.

Both global and local descriptors presented here have proved to perform very well for such tasks as object detection and recognition; however, to the knowledge of the authors, they have not been tried for place recognition. In Chapter 8 we will show that they can be successfully applied to this difficult problem, giving very good results in spite of presence of huge variations in viewpoint and illumination conditions.

Chapter 5

Classification Using Support Vector Machines

It was stated in Chapter 2 that a *classifier* is an algorithm that performs the actual recognition on the basis of the feature vectors extracted from the input patterns. Consequently, *classification* together with feature extraction are essential stages of the recognition process that have the largest influence on the performance and robustness of the system. In this chapter we will show how to solve the classification problem using the *Support Vector Machines* (SVM) [78, 17, 33]. Due to their superior performance and well developed theoretical background, in the recent years, the Support Vector Machines classifier attracted considerable attention and was successfully applied to numerous applications from computer vision to computational biology. As most of the general issues connected with classification have already been discussed in Chapter 2, here we will focus on the details of this particular method.

The wide variety of the available classification algorithms can be roughly divided into two categories depending on the method employed for representing the *model*. The first group is constituted by *generative classifiers* which aim at computing the probability that a pattern belongs to a certain class by estimating the probability of observing a pattern in a certain class. In contrast, *discriminative classifiers* avoid this intermediate step by forming *discriminant functions* mapping input patterns to class labels directly. The Support Vector Machines constitute an example of a discriminant classifier inspired from the Vapnik's statistical learning theory [78]. Details will be given in successive sections of this chapter.

In Section 5.1 we will present the principles of discriminative classification. On this basis, we will describe the linear Support Vector Machines. In Section 5.2 we will show how the classification problem can be solved in a high-dimensional feature space. Next, in Section 5.3 we will present several multi-class extensions to the SVM. We will conclude with a summary in Section 5.4.

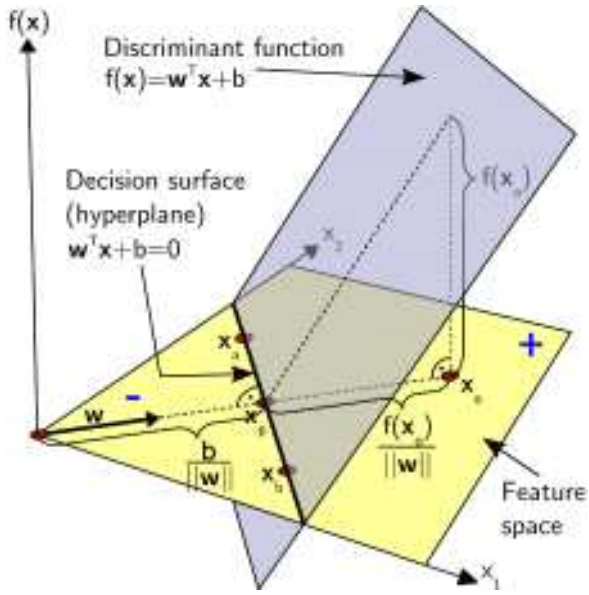


Figure 5.1. The linear discriminative function $f(\boldsymbol{x})$ dividing the feature space into two half-spaces by a hyperplane decision surface.

5.1 Support Vector Machines as a Linear Discriminative Classifier

This section aims to describe the Support Vector Machines as a linear, binary, large-margin, discriminant classifier. For this reason, the fundamentals of discriminative classification are given in Section 5.1.1. This knowledge is then used in Section 5.1.2 to explain the algorithm finding the optimal separating hyperplane for a linearly separable case. Finally, Section 5.1.3 extends this method to non-linearly separable problems by presenting the idea of the soft margin hyperplane.

5.1.1 Linear Discriminative Classifier

In case of *supervised learning* the classifier is built upon a set of labeled training samples. For a two-class problem, the set can be denoted as $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, where $\boldsymbol{x}_i \in \Re^N$ is a feature vector, and $y_i \in \{-1, 1\}$ determines the membership of the vector to one of the two classes. An assumption is made that the positive value indicates the class ω_1 and the negative value indicates the class ω_2.

Every feature vector can be considered as a point in an N-dimensional *feature space*. Consequently, in classification the aim is to find a *discriminant function* $f : \Re^N \rightarrow \Re$ distinguishing between the points belonging to the different classes. If $f(\boldsymbol{x}) > 0$, then the point $\boldsymbol{x}$ is classified to the class ω_1, and if $f(\boldsymbol{x}) < 0$, it is classified to the class ω_2. The problem is illustrated in Figure 5.1.

The linear discriminant function is given by

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b, \tag{5.1}$$

where $\boldsymbol{w}$ is the *weight vector* and b is the *bias*. The function divides the feature space into two half-spaces by a *hyperplane decision surface*

$$f(\boldsymbol{x}) = 0 = \boldsymbol{w}^T\boldsymbol{x} + b. \tag{5.2}$$

On the basis of the Eq. 5.1 we may say that the training set is *linearly separable* if the following equations hold:

$$\begin{array}{ll} \boldsymbol{w}^T\boldsymbol{x}_i + b \geq 0 & \text{for } y_i = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_i + b < 0 & \text{for } y_i = -1 \end{array} \tag{5.3}$$

We will now study several basic properties of the discriminant function that will be helpful in understanding the theory of the SVMs. Consider the two points $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ visible in Figure 5.1. As both points lie exactly on the hyperplane

$$\begin{array}{c} f(\boldsymbol{x}_a) = f(\boldsymbol{x}_b) = 0 \\ \boldsymbol{w}^T\boldsymbol{x}_a + b = \boldsymbol{w}^T\boldsymbol{x}_b + b \\ \boldsymbol{w}^T(\boldsymbol{x}_a - \boldsymbol{x}_b) = 0. \end{array} \tag{5.4}$$

This shows that every vector parallel to the hyperplane is normal to the weight vector $\boldsymbol{w}$. The discriminant function $f(\boldsymbol{x})$ can be also regarded as a measure of the distance of the point $\boldsymbol{x}$ to the hyperplane [20]. Consider the point $\boldsymbol{x}_o$ and its normal projection to the hyperplane $\boldsymbol{x}_p$, both shown in Figure 5.1. We may express the coordinates of the point $\boldsymbol{x}_o$ using the point $\boldsymbol{x}_p$ and the unit vector $\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$:

$$\boldsymbol{x}_o = \boldsymbol{x}_p + d\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \tag{5.5}$$

where d is the algebraic distance between the point $\boldsymbol{x}_o$ and the hyperplane. Since $f(\boldsymbol{x}_p) = 0$ it follows that

$$f(\boldsymbol{x}_o) = \boldsymbol{w}^T\left(\boldsymbol{x}_p + d\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) + b = f(\boldsymbol{x}_p) + d\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|} = d\|\boldsymbol{w}\|, \tag{5.6}$$

and

$$d = \frac{f(\boldsymbol{x}_o)}{\|\boldsymbol{w}\|}. \tag{5.7}$$

The algebraic distance is positive if the point lies on the positive half-space and negative if it lies on the negative half-space. In order to obtain a measure that is always positive the algebraic distance may be multiplied by the class label y_i.

On the same basis, we may express the distance between the origin and the separating hyperplane as

$$\frac{f(\boldsymbol{0})}{\|\boldsymbol{w}\|} = \frac{b}{\|\boldsymbol{w}\|}. \tag{5.8}$$

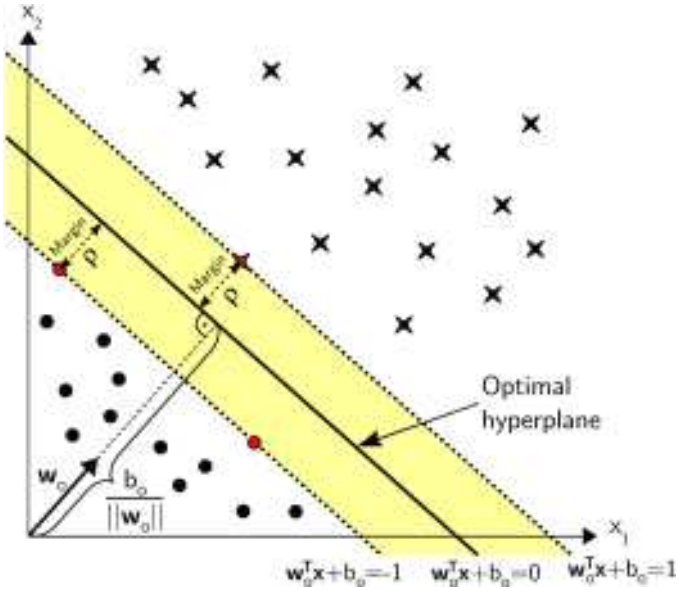


Figure 5.2. The optimal separating hyperplane maximizing the margin. The support vectors are marked in red.

5.1.2 Optimal Separating Hyperplane

As it was already mentioned, in classification the problem is to find the form of the discriminant function $f : \Re^N \to \Re$. Since in linear case the function $f(\boldsymbol{x})$ is defined as a linear combination of the elements of the vector x (see Eq. 5.1 and 5.2), the problem is reduced to finding the parameters of the separating hyperplane. The Support Vector Machines belong to the group of the so-called large-margin classifiers. This is due to the fact that in the linearly separable case SVMs obtain the *optimal separating hyperplane* with maximal distance to the samples from both classes. This distance is referred to as *margin* ρ, as it is shown in Figure 5.2.

The margin ρ can be defined as the distance between the hyperplane and the nearest of the points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$. Then, it follows that

$$\rho = \frac{\min\limits_{i=1,\ldots,n} y_i f(\boldsymbol{x}_i)}{\|\boldsymbol{w}\|} = \frac{\min\limits_{i=1,\ldots,n} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|}. \tag{5.9}$$

We see that the margin may be maximized by either maximizing the absolute value of the discriminant function at the nearest point ($\min_{i=1,\ldots,n} y_i f(\boldsymbol{x}_i)$) or by minimizing the length of the weight vector $\|\boldsymbol{w}\|$. Consequently, some constraint must be imposed in order to find a unique solution. This is usually done by assuming that the value of the function at the nearest point is equal to 1. In that case, the linear separability condition can be written as

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{x}_i + b \geq +1 \qquad & \text{for } y_i = +1,\ i = 1, 2, \ldots, n \\ \boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1 \qquad & \text{for } y_i = -1,\ i = 1, 2, \ldots, n, \end{aligned} \tag{5.10}$$

or

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \qquad \text{for } i = 1, 2, \ldots, n. \tag{5.11}$$

The feature vectors for which the the inequalities become equations for the optimal separating hyperplane are known under the name of *support vectors*.

As a result, we can formulate the following optimization problem [33]: Given the labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, find the optimal value of the weight vector $\boldsymbol{w}_o$ and the bias b_o such that they satisfy the constraints

$$y_i(\boldsymbol{w}_o^T\boldsymbol{x}_i + b_o) \geq 1 \qquad \text{for } i = 1, 2, \ldots, n. \tag{5.12}$$

and the weight vector $\boldsymbol{w}_o$ minimizes the cost function:

$$\phi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \tag{5.13}$$

This is a constrained optimization problem called the *primal problem*. It may be solved by constructing the *Lagrangian function* [64]

$$J(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{n} \alpha_i \left[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\right], \tag{5.14}$$

where α_i are called *Lagrange multipliers*. The solution of the optimization problem corresponds to the saddle point of the Lagrangian, which has to be minimized with respect to $\boldsymbol{w}$ and b and maximized with respect to α_i. Consequently, the following conditions can be defined:

$$\frac{\partial J(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = \boldsymbol{0} \tag{5.15}$$

and

$$\frac{\partial J(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = 0. \tag{5.16}$$

Differentiating the Lagrangian function gives

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \tag{5.17}$$

and

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{5.18}$$

It is important to note that according to the *Karush-Kuhn-Tucker theorem* [64, 33], the following equation is satisfied at the saddle point of the Lagrangian:

$$\alpha_{io}\left[y_i(\boldsymbol{w}_o^T\boldsymbol{x}_i + b_o) - 1\right] = 0 \qquad \text{for } i = 1, 2, \ldots, n \tag{5.19}$$

This shows that $\alpha_{io} \neq 0$ only for those feature vectors x_i for which $y_i(\boldsymbol{w}_o^T\boldsymbol{x}_i + b_o) = 1$, i.e. for the support vectors.

The primal optimization problem can be transformed into *dual problem*. This can be done by substituting Eq. 5.17 and 5.18 into the Lagrangian function presented in Eq. 5.14. The resulting equation yields

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j. \tag{5.20}$$

Similarly to the primal problem, we may now formulate the following dual optimization problem [33]: Given the labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, find the Lagrange multipliers $\{\alpha_{i,o}\}_{i=1}^{n}$ that maximize the objective function presented in Eq. 5.20 subject to the constraints

1. $\sum_{i=1}^{n} \alpha_{i,o} y_i = 0$
2. $\alpha_{i,o} \geq 0 \qquad$ for $i = 1, 2, \ldots, n$.

At this point it is important to notice that the only operation that is performed on the feature vectors is the inner product. In Section 5.2 we will show how to exploit this fact to perform classification in a very high-dimensional feature space.

The Lagrange multipliers determined as a result of the optimization process can be used in order to compute the optimal weight vector

$$\boldsymbol{w}_o = \sum_{i=1}^{n} \alpha_{i,o} y_i \boldsymbol{x}_i. \tag{5.21}$$

The optimal bias can be computed using any support vector x_s according to Eq. 5.12, that is

$$\begin{gathered} y_i(\boldsymbol{w}_o^T \boldsymbol{x}_s + b_o) = 1 \\ b_o = 1 - \boldsymbol{w}_o^T \boldsymbol{x}_s \qquad \text{for } y_s = 1. \end{gathered} \tag{5.22}$$

Finally, we can use the optimal parameters $\boldsymbol{w}_o$ and b_o to formulate the discriminant function defining the optimal separating hyperplane. Since $\alpha_{i,o}$ for non-support vectors equals to 0, the discriminant function can be expressed only in terms of support vectors

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_{i,o} y_i \boldsymbol{x}_i^T \boldsymbol{x} + b_o, \tag{5.23}$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ are the support vectors and $\alpha_{i,o}$ are the corresponding Lagrange multipliers. Again, we see the only operation that is performed on the feature vectors is the inner product. We may also notice that the knowledge of the classifier (model) is represented in form of a subset of the training samples, the corresponding Lagrange multipliers and the bias.

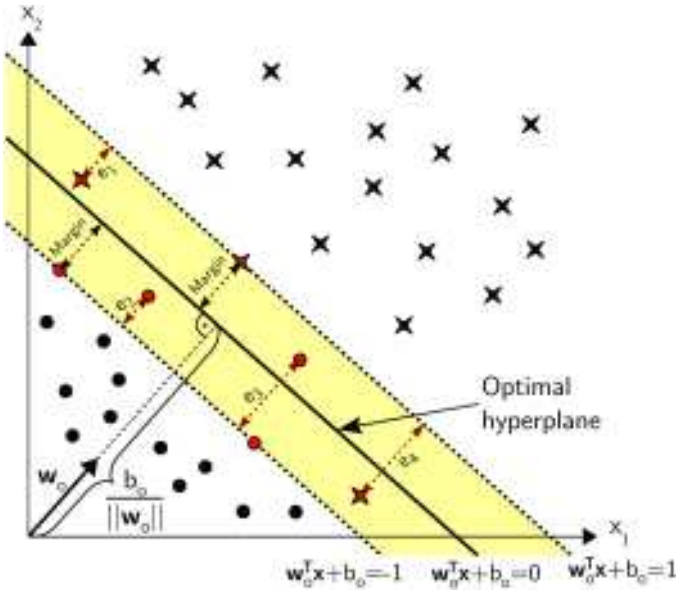


Figure 5.3. Illustration of the soft margin hyperplane. Two points cross the boundary of the margin but are located on the right side of the hyperplane. Other two points are located on the wrong side. The variables e_i denote the errors. The support vectors are marked in red.

5.1.3 Soft Margin Hyperplane

In this section we consider the non-linearly separable case, very common in practical applications. Although the training samples cannot be discriminated using a hyperplane, we may try to formulate the optimization problem in a way that will allow to minimize the classification error on the training set.

The problem can be illustrated as in Figure 5.3. The hyperplane roughly separates the points belonging to different classes; however, several points violate the rule of linear separability for SVMs given in Eq. 5.11. The violation can either cause a misclassification of a training sample, or the sample may just cross the boundary of the margin but be still located on the correct half-space. Since it is impossible to eliminate the errors completely, the optimization problem should be formulated in such a way that it leads to a solution for which the errors are minimized. We start by redefining the condition presented in Eq. 5.11 so that it makes allowance for the errors explicitly. It follows that

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n, \tag{5.24}$$

where the variables ξ_i are non-negative and are referred to as the *slack variables*. The slack variables can be seen as a measure of the violation of the margin. If $0 < \xi_i < 1$, then the point crosses the boundary of the margin; however it is still properly classified. The value greater than 1 means that the point falls on the wrong side of the hyperplane.

The slack variables can be used to state the goal of minimizing the average error on the training set in formal way [33]. For this reason, we introduce a functional

$$\phi(\boldsymbol{\xi}) = \sum_{i=1}^{n} \xi_i, \tag{5.25}$$

which should be minimized with respect to w. It may be incorporated into the cost function defined in Eq. 5.13 as follows:

$$\phi(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{n} \xi_i, \tag{5.26}$$

where C is a parameter controlling the trade-off between the complexity of the classifier and the amount of errors and it has an influence on the generalization performance. The parameter is adjusted by the user.

We can now reformulate the primal optimization problem [33]: Given the labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, find the optimal value of the weight vector $\boldsymbol{w}_o$ and the bias b_o such that they satisfy the constraint presented in Eq. 5.24 for non-negative values of the ξ_i variables and such that the vector $\boldsymbol{w}$ and the slack variable ξ_i minimize the cost function given in Eq. 5.26. The Lagrangian function for this problem is given by

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{u}, \boldsymbol{\xi}) &= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{n} \xi_i - \sum_{i}^{n} \mu_i\xi_i - \\ &\quad \sum_{i=1}^{n} \alpha_i \left[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\right], \end{aligned} \tag{5.27}$$

where μ_i are Lagrangian multipliers introduced to enforce the non-negativity of the slack variables.

The corresponding dual problem does not depend on the slack variables and the only difference lies in the second constraint which is replaced by:

$$0 \leq \alpha_{i,o} \leq C \qquad \text{for } i = 1, 2, \ldots, n. \tag{5.28}$$

The optimal weight vector as well as the optimal bias are computed using the same algorithm as in the previous case. It is important to note that the definition of the support vectors is also the same as before. As a result, the final discriminant function in both cases is defined by Eq. 5.23.

It should be pointed out that the idea presented in this section is usefull not only for non-linearly separable problems. Another advantage is that it is now possible to avoid situations in which the hyperplane is optimal with respect to the errors on the training set but does not generalize well to the novel samples (overfitting). This ability can be controlled by the user by adjusting the value of the variable C.

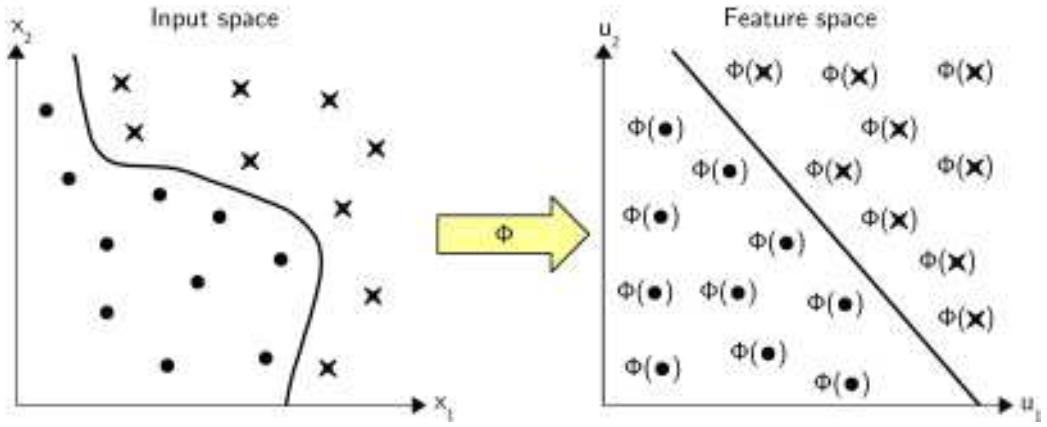


Figure 5.4. Non-linear mapping allowing the non-linearly separable points to be separated by a hyperplane in a very high-dimensional feature space.

5.2 Non-linear Support Vector Machines

The previous section showed how to find the optimal separating hyperplane in the linearly separable and non-separable case. In this section we will describe a method allowing for making the SVM classifier non-linear by non-linearly mapping the input vectors to a very high-dimensional *feature space* and constructing the linear decision surface in that space. This can be done efficiently by exploiting the fact that it is possible to compute the inner product of the input vectors in the feature space using the so called *kernel functions* (*kernels*) without explicitly determining the high-dimensional representation of the vectors. This idea is referred to as the *kernel trick*.

Section 5.2.1 shows how to perform classification in the high-dimensional feature space using linear Support Vector Machines. Several commonly used kernel functions are described Section 5.2.2. The section contains a description of the local kernel function used to combine the local feature representation with the SVM classifier.

5.2.1 The Kernel Trick

Let us consider a simple example of a non-linearly separable classification problem: Find a decision surface separating 4 samples in a two-dimensional space located on the corners of a rectangle in such way that the samples connected by a diagonal belong to the same class. Such problem is commonly referred as the *XOR problem*. Naturally, it is not possible to separate the samples using a hyperplane in two dimensions (a line). However, if we imagine that the rectangle is a sheet of paper we can easily fold it along the diagonal and separate by a hyperplane in three dimensions (e.g. another sheet of paper). This shows that a non-linearly separable problem may become linearly separable after mapping to a higher-dimensional

space. In general, it is possible to increase the separability of the points in $\Re^N$ by mapping them to some space $\mathcal{H}$ through a non-linear function ϕ as illustrated in Figure 5.4. From now on the term feature space will be used to refer to the space $\mathcal{H}$ in which the classification is performed.

The function ϕ can be defined as follows:

$$\phi : \Re^N \to \mathcal{H}. \tag{5.29}$$

It can be used to define the discriminant function of the SVM classifier in the feature space $\mathcal{H}$:

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i y_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) + b, \tag{5.30}$$

where all the parameters are defined as in Eq. 5.23. We see that such representation required computing the inner product of the vectors in the feature space. However, performing computations in such space could be extremely costly due to its high dimensionality. This problem can be solved by exploiting the so-called *kernel trick*.

It was already stated in the previous section that the only operation that is performed on the feature vectors is the inner product. This is true for both the optimization process and classification. Consequently, it is possible to avoid determining the feature space representation of the vectors by introducing the *kernel function* defined by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y}). \tag{5.31}$$

We may now substitute Eq. 5.31 into Eq. 5.30 obtaining

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{5.32}$$

The same substitution should be made in Eq. 5.20 defining the objective function of the dual optimization problem.

The kernel function $K(\boldsymbol{x}, \boldsymbol{y})$ can be seen as a similarity measure between the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. However, in order to ensure that there exist a space in which this measure corresponds to an inner product, the function must satisfy the *Mercer's theorem*, that is, the *kernel matrix* $\boldsymbol{K}$ given by

$$\boldsymbol{K} = \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix} \tag{5.33}$$

must by positive semi-definite. In other words it must have only non-negative eigenvalues. The subject is comprehensively studied in [17]. Practical examples show that it is still possible to use kernels that do not satisfy the Mercer's condition for Support Vector Machines. Although, in that case, it is not guaranteed that there exist a space in which the kernel function is an inner product, the performance of a classifier employing such function may still be very good. Examples of commonly used kernel functions are presented in the next section.

5.2.2 Kernel Functions

It was already mentioned that the kernel functions can be considered as a similarity measure between two vectors. For this reason, numerous specialized kernels have been proposed in order to classify various kinds of data (see e.g. [82, 15, 5]). There are, however, several widely known kernel functions that perform well in a variety of applications. These are inter alia:

- Polynomial kernel
$$K(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{x}^T \boldsymbol{y} + p\right)^d \tag{5.34}$$
A special case $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$ is referred to as the *linear kernel.*

- Gaussian kernel
$$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{2\sigma^2}} = e^{-\gamma\|\boldsymbol{x}-\boldsymbol{y}\|} \tag{5.35}$$

- Sigmoid kernel
$$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh\left(\kappa \boldsymbol{x}^T \boldsymbol{y} + \theta\right) \tag{5.36}$$
The Mercer's theorem for the sigmoid kernel is satisfied only for some values of κ and θ.

The parameters of the kernels are specified by the user, usually experimentally.

In the experiments in Chapters 7 and 8, we employed two kernels specialized in a particular type of input data: the χ^2 kernel [15, 5] for classifying the Composed Receptive Field Histograms, and the local kernels [82] performing matching of the local image features. The χ^2 kernel is given as follows:

$$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\gamma \chi^2(\boldsymbol{x}, \boldsymbol{y})}, \tag{5.37}$$

where the χ^2 measure is given by

$$\chi^2(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{5.38}$$

The χ^2 kernel is proved to be a Mercer's kernel [5] and was shown to be effective in experiments with multi-dimensional histograms [38, 15].

As it was stated in Chapter 4, comparing image representations based on local features is usually done by matching, i.e. each local descriptor extracted from the first image is compared to the descriptors extracted from the second image. Such approach requires using a specialized kernel with the SVMs. In experiments presented in this thesis the *local kernels* proposed by Wallraven *et al.* [82] were employed. The local kernel function is given by

$$K(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{2}\left[\widetilde{K}(\boldsymbol{L}_h, \boldsymbol{L}_k) + \widetilde{K}(\boldsymbol{L}_k, \boldsymbol{L}_h)\right], \tag{5.39}$$

with

$$\widetilde{K}(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{n_h} \sum_{i=1}^{n_h} \max_{j=1,\ldots,n_k} K_l(\boldsymbol{l}_{h,i}, \boldsymbol{l}_{k,j}), \tag{5.40}$$

where $\boldsymbol{L}_h = \{\boldsymbol{l}_{h,i}\}_{i=1}^{n_h}$ is a vector of local features extracted from one image, and $K_l(\boldsymbol{x}, \boldsymbol{y})$ is a kernel used as a measure of similarity between the local descriptors x and y. In our experiments we used the Gaussian kernel given in Eq. 5.35 as K_l. This is due to the fact that the Gaussian kernel employs the Euclidean distance which is the most commonly used measure for comparing the SIFT descriptors. The kernel presented above is an example of a non-Mercer's kernel which, however, performs very well in practice.

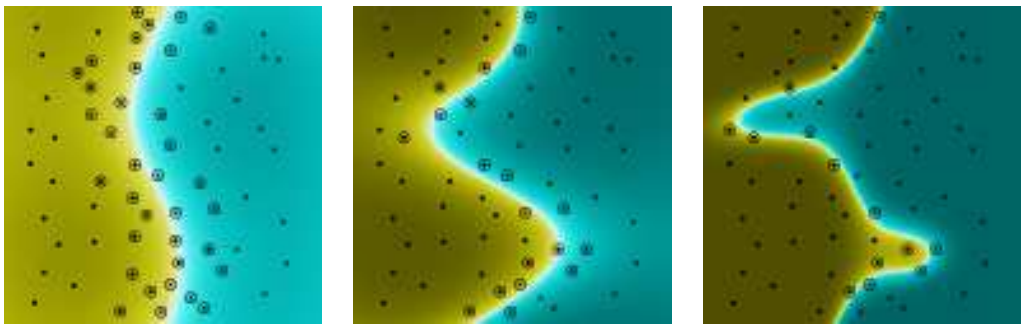
5.3 Multi-class Extensions to Support Vector Machines

Support Vector Machines were designed for binary classification. However, in many practical applications the number of classes is greater than two. For this reason, several extensions of SVMs allowing for multi-class classification were proposed in the literature. In general, they can be divided into two groups: the *"all-together"* methods (see e.g. [83]) which try to solve the multi-class problem in one step by reformulating the optimization problem discussed in Section 5.1, and the *binary-based* methods which employ several binary SVM classifiers for this purpose. According to the comparison presented in [37] the accuracy of all algorithms is very similar; however, the usage of the all-together methods is currently limited to small data sets as they require solving a much larger optimization problem. In this section we present two algorithms belonging to the second group of methods: the *one-against-one* and the *one-against-all* methods. Let us describe each algorithm in turn. We will consider a c-class problem for classes $\{\omega_i\}_{i=1}^{c}$.

One-against-all The one-against-all method employs c classifiers. The i-th classifier is trained to discriminate between the class ω_i and all the other classes. During the test phase, the sample is classified using all classifiers and the final decision is made on the basis of the values of the discriminant functions as follows:

$$c = \arg\max_{i=1,\ldots,c} f_i(\boldsymbol{x}) \tag{5.41}$$

One-against-one In the case of the one-against-one approach, $\frac{c(c-1)}{2}$ classifiers are trained to discriminate between each pair of classes. In order to make the final decision, each classifier votes on one class depending on the sign of its discriminant function. Consequently, the class which collects the highest number of votes is selected.



$C = 1$ $C = 100$ $C = 10000$

Figure 5.5. A simple two-dimensional non-linearly separable classification problem solved using the SVMs with the Gaussian kernel ($\gamma = 5$) for three different values of the parameter C. The color saturation depends on the value of the discriminant function. Support vectors are marked with circles.

5.4 Summary

In this chapter we provided a detailed description of the Support Vector Machines classifier, as well as the fundamentals of the discriminative classification in general. First, we studied the optimization techniques used to obtain the optimal separating hyperplane in a linearly separable case. As a consequence, we derived the discriminant function of the SVM classifier. Then, we extended the problem to the non-linearly separable case and showed how to find the soft margin hyperplane minimizing the errors on the training set. The same method applies to the cases when the optimal hyperplane defined in a standard way could lead to overfitting. Finally, we showed how a non-linear mapping may increase the separability of the data and how to perform the mapping using the inner-product kernels. This lead us to the definitions of several commonly used kernel functions as well as of those used in our experiments to classify pictures based on the global histogram representation or local descriptors. It is worth mentioning that we omitted the issue of the efficiency of the classifier purposely. In the next chapter, which we devote to this subject, we will show a method for improving the efficiency and decreasing the memory requirements of the Support Vector Machines.

In the end, we would like to illustrate the concepts discussed in this chapter by a simple real example of a classification problem solved using the Support Vector Machines. Figure 5.5 presents a two-dimensional non-linearly separable case and three solutions obtained for a Gaussian kernel and different values of the parameter C. We see that C can be used to control the generalization ability of the classifier.

Chapter 6

Support Vector Reduction

This chapter provides a detailed description of a method for improving the efficiency and decreasing the memory requirements of the Support Vector Machines. In numerous cases, these are crucial parameters of a classifier that determine its usefulness for certain types of applications. Consider the example of a place recognition system. Due to the high complexity of the problem (large within-class variability, constantly changing environment) the system may require huge amounts of memory in order to store its knowledge. On the other hand, it is likely to be mounted either on a robot or on another mobile platform that usually have limited resources. Moreover, it may be necessary to deliver real-time performance which makes the efficiency of the classifier of utmost importance. The Support Vector Machines are known to provide excellent generalization capabilities and in many domains are recognized as state-of-the-art (see e.g. [13]); however can be considerably slower in test phase than other classification methods. As it was shown in the previous chapter, the discriminant function of the SVMs is parametrized by a subset of the training vectors - the support vectors. Consequently, the number of support vectors is a crucial factor influencing the speed[1] and the amount of memory required by the classifier.

Several authors suggested that the solution generated by the standard SVC learning algorithm is not always minimal ([12, 11, 63]), that is, it is possible to generate another solution offering identical generalization performance while having a smaller number of support vectors. On the other hand, experiments performed by Syed *et al.* [72] showed that rejecting even a small number of randomly selected support vectors may cause a strong decrease in performance. This raises the question of whether the complexity of the support vector solution can be reduced while preserving its optimal performance.

The phenomenon described above can be observed in a simple example in Figure 6.1, upper left. The illustration presents the solution of a two-dimensional classification problem obtained using a linear kernel and the Sequential Minimal

[1] In case of incremental learning, the number of support vectors may affect not only the speed in the test phase but also the training time (see [63]).

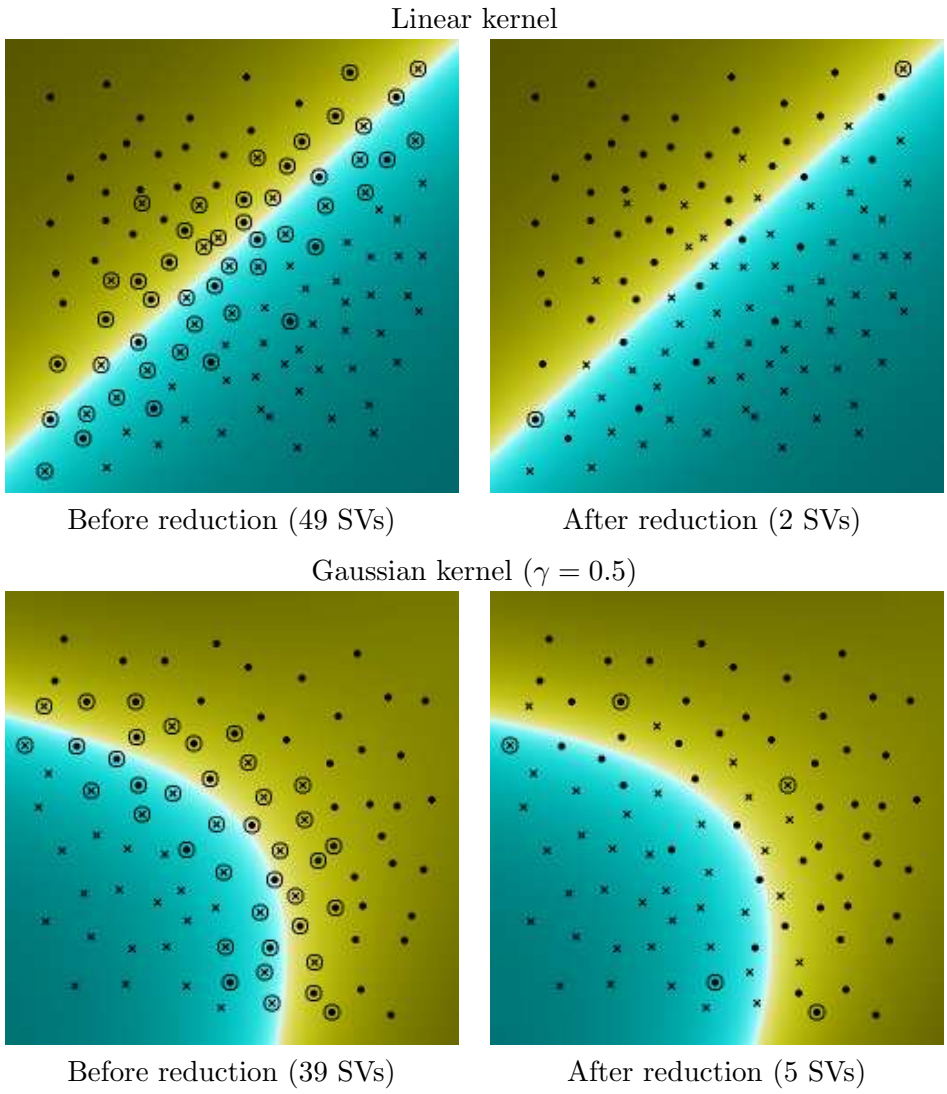


Figure 6.1. Simple examples illustrating the result of applying the Support Vector Reduction algorithm presented in this thesis to the classifiers trained using two-dimensional data. In each case, training was performed using the SMO algorithm and $C = 100$. The color saturation depends on the value of the discriminant function, and the support vectors are marked with circles.

Optimization (SMO) [58, 14] training algorithm. In such a case, the support vectors can be regarded as basis vectors used to transform the input vector into new coordinates before it is multiplied by the α_i coefficients. Intuitively, in order to uniquely represent any point in a two-dimensional space, it is enough to provide two linearly independent basis vectors. As we see in the example, the standard algorithm found 49 support vectors, whereas any two linearly independent vectors should be enough.

The idea that the standard learning algorithm may generate a set of support vectors that is not linearly independent in the space in which the linear classification is performed was first proposed and experimentally evaluated by Downs *et al.* [19]. Their method allows to reduce the number of support vectors of a trained classifier by eliminating those that can be expressed as a linear combination of the others in the feature space. The weights α_i are updated accordingly, which ensures that the decision function is exactly the same as the original one. This results in a reduction of the complexity of the classifier, without any loss in performance. Experiments presented in [19] show that the algorithm can be successfully applied to the polynomial and Gaussian kernels.

In this chapter we present the theoretical background of the method proposed by Downs *et al.* [19] as well as our implementation employing the QR factorization with column pivoting [28, 29] as a method for selecting the linearly dependent support vectors. We further extend the original reduction algorithm by introducing a threshold value, which can be used to find the optimal trade-off between the complexity of the classifier (and thus the memory requirements and the speed in the test phase) and the classification performance. The method is thoroughly tested on visual data for multi-class problems of different complexity drawn from two domains: material categorization and place recognition. The results of experiments are reported in Chapters 7 and 8. Simple examples are presented in Figure 6.1. We see that the algorithm leads substantial reduction in the number of support vectors while keeping the solution unchanged.

Several other approaches aiming to decrease the complexity of the SVM classifier were proposed in the literature. Burges [11] and Burges and Schölkopf [12] describe a method for approximating the solution using smaller number of vectors, which, however, are not support vectors. Additionally, this method seems to be computationally expensive. Another approach by Osuna and Girosi [57] employs Support Vector Regression in order to approximate the discriminant function. These methods are generally approximate methods and only the algorithm proposed by Downs *et al.* [19] guarantees to keep the solution intact.

The rest of this chapter is organized as follows: Section 6.1 provides a description of the original method proposed by Downs *et al.*. Section 6.2 presents the algorithm of the QR factorization with column pivoting. Finally, Section 6.3 shows how to use the QR factorization in order to identify the linearly dependent support vectors, and how to control the trade-off between the complexity of the classifier and the classification performance. The chapter concludes with a summary in Section 6.4.

6.1 Linear Dependence in the Feature Space

The idea behind the algorithm by Downs *et al.* [19] is that the set of support vectors is not guaranteed to be linearly independent in the feature space. Recall

the standard discriminant function of the SVM classifier given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b, \tag{6.1}$$

where $\{\boldsymbol{x}_i\}_{i=1}^m$ is the set of support vectors. Let us suppose that the linearly dependent supports vectors were already identified and sorted so that the first r support vectors are linearly independent, and the remaining $m-r$ depend linearly on those in the feature space:

$$\underset{j=r+1,\ldots,m}{\forall} : \phi(\boldsymbol{x}_j) \in span\{\phi(\boldsymbol{x}_i)\}_{i=1}^r. \tag{6.2}$$

Then for all $\boldsymbol{x}_j$, $j = r+1, \ldots, m$, it holds

$$\phi(\boldsymbol{x}_j) = \sum_{i=1}^{r} c_{ij} \phi(\boldsymbol{x}_i), \tag{6.3}$$

and

$$K(\boldsymbol{x}, \boldsymbol{x}_j) = \sum_{i=1}^{r} c_{ij} K(\boldsymbol{x}, \boldsymbol{x}_i). \tag{6.4}$$

Eq. 6.4 can now be substituted into Eq. 6.1. This leads to the discriminant function of the form

$$f(\boldsymbol{x}) = \left(\sum_{i=1}^{r} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{j=r+1}^{m} \alpha_j y_j \sum_{i=1}^{r} c_{ij} K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right). \tag{6.5}$$

We see that the function is not anymore parametrized by the linearly dependent support vectors. If we define the coefficients γ_{ij}, such that $\alpha_j y_j c_{ij} = \alpha_i y_i \gamma_{ij}$ and $\gamma_i = \sum_{j=r+1}^{m} \gamma_{ij}$, then Eq. 6.5 can be written as

$$\begin{aligned} f(\boldsymbol{x}) &= \left(\sum_{i=1}^{r} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{i=1}^{r} \alpha_i y_i \sum_{j=r+1}^{m} \gamma_{ij} K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right) \\ &= \left(\sum_{i=1}^{r} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{i=1}^{r} \alpha_i y_i \gamma_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right) \\ &= \left(\sum_{i=1}^{r} \alpha_i (1+\gamma_i) y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right) \\ &= \left(\sum_{i=1}^{r} \widetilde{\alpha}_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right), \end{aligned} \tag{6.6}$$

where

$$\widetilde{\alpha}_i = \alpha_i (1+\gamma_i) = \alpha_i \left(1 + \sum_{j=r+1}^{m} \frac{\alpha_j y_j c_{ij}}{\alpha_i y_i} \right) \tag{6.7}$$

The α_i coefficients can be pre-multiplied by the class labels $\alpha'_i = \alpha_i y_i$ which results in a simple equation that can be used to obtain the weights of the reduced classifier:

$$\widetilde{\alpha}'_i = \begin{cases} \alpha'_i + \sum_{j=r+1}^{m} \alpha'_j c_{ij} & \text{for } i = 1, 2, \ldots, r \\ 0 & \text{for } i = r+1, r+2, \ldots, m. \end{cases} \tag{6.8}$$

In conclusion, the resulting discriminant function (Eq. 6.6) requires now $m - r$ less kernel evaluations than the original one (Eq. 6.1).

6.2 QR Factorization

The previous section provided a simple expression for computing the weights of the SVM classifier after reducing multiple linearly dependent support vectors at once. However, it is necessary to first identify the linearly dependent support vectors and to determine the coefficients c_{ij}. We employ the *QR factorization with column pivoting* [28, 29] for this purpose.

The QR factorization with column pivoting algorithm is a widely used method for selecting the independent columns of a matrix. The algorithm allows to reveal the numerical rank of the matrix with respect to a parameter τ, which acts as a threshold in defining the condition of linear dependence. Additionally, it performs a permutation of the columns of the matrix so that they are ordered according to the degree of their relative linear independence. Consequently, if for a given value of τ the rank of the matrix is r, then the linearly independent columns will occupy the first r positions.

The QR factorization with column pivoting of a matrix $\boldsymbol{K} \in \Re^{n \times m}, n \geq m$ is given by

$$\boldsymbol{K}\boldsymbol{\Pi} = \boldsymbol{Q}\boldsymbol{R}, \tag{6.9}$$

where $\boldsymbol{\Pi} \in \Re^{m \times m}$ is a permutation matrix, $\boldsymbol{Q} \in \Re^{n \times n}$ is orthogonal, and $\boldsymbol{R} \in \Re^{n \times m}$ is upper triangular. If we assume that the rank of the matrix K with respect to the parameter τ equals r, then the matrices can be decomposed as follows:

$$\begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{R}_{22} \end{bmatrix}, \tag{6.10}$$

where the columns of $\boldsymbol{K}_1 \in \Re^{n \times r}$ create a linearly independent set, the columns of $\boldsymbol{K}_2 \in \Re^{n \times m-r}$ may be expressed as a linear combination of the columns of $\boldsymbol{K}_1$, $\boldsymbol{Q}_1 \in \Re^{n \times r}$ is orthogonal, $\boldsymbol{Q}_2 \in \Re^{n \times n-r}$ is orthogonal, $\boldsymbol{R}_{11} \in \Re^{r \times r}$ is upper triangular, $\boldsymbol{R}_{12} \in \Re^{r \times m-r}$, and $\boldsymbol{R}_{22} \in \Re^{n-r \times m-r}$.

The products of the QR factorization can be used to obtain the coefficients determining how the linearly dependent columns depend on the independent ones:

$$\begin{cases} \boldsymbol{K}_1 \boldsymbol{C} = \boldsymbol{K}_2 \\ \boldsymbol{K}_1 = \boldsymbol{Q}_1 \boldsymbol{R}_{11} \end{cases}$$

$$\begin{cases} \boldsymbol{C} = \boldsymbol{K}_1^{-1} \boldsymbol{K}_2 \\ \boldsymbol{K}_1 = \boldsymbol{Q}_1 \boldsymbol{R}_{11} \end{cases}$$

$$\boldsymbol{C} = (\boldsymbol{Q}_1\boldsymbol{R}_{11})^{-1}\boldsymbol{K}_2 = \boldsymbol{R}_{11}^{-1}\boldsymbol{Q}_1^{-1}\boldsymbol{K}_2. \tag{6.11}$$

Since $\boldsymbol{Q}_1$ is orthogonal

$$\boldsymbol{C} = \begin{bmatrix} c_{1,r+1} & \cdots & c_{1,m} \\ \vdots & \ddots & \vdots \\ c_{r,r+1} & \cdots & c_{r,m} \end{bmatrix} = \boldsymbol{R}_{11}^{-1}\boldsymbol{Q}_1^{T}\boldsymbol{K}_2, \tag{6.12}$$

where $\boldsymbol{C} \in \Re^{r\times m-r}$ and c_{ij} is a coefficient determining how the j-th column depends on the i-th one.

The algorithm computing the QR factorization with column pivoting employed during experiments presented in this thesis utilizes *the Householder transformations* [29]. Consequently, the matrix $\boldsymbol{Q}$ can be seen as a product of the *Householder matrices* that were used to transform the matrix $\boldsymbol{K}$ into the upper triangular matrix $\boldsymbol{R}$ as follows:

$$\underbrace{\underbrace{\underbrace{\boldsymbol{Q}^T\boldsymbol{K}\boldsymbol{\Pi}}_{\boldsymbol{P}^{(m+1)}=\boldsymbol{R}} = \boldsymbol{H}_m\,\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1\underbrace{\boldsymbol{K}\boldsymbol{\Pi}}_{\boldsymbol{P}^{(1)}}}_{\boldsymbol{P}^{(2)}}}_{\boldsymbol{P}^{(m)}} = \boldsymbol{R}. \tag{6.13}$$

At the k-th stage of the algorithm $\boldsymbol{P}^{(k+1)} = \boldsymbol{H}_k\boldsymbol{P}^{(k)}$, and the matrix $\boldsymbol{H}_k \in \Re^{n\times n}$ is given by

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{I}_{k-1} & \boldsymbol{0} \\ \boldsymbol{0} & \widetilde{\boldsymbol{H}}_k \end{bmatrix}, \tag{6.14}$$

where $\boldsymbol{I}_k \in \Re^k$ is an identity matrix, and $\widetilde{\boldsymbol{H}}_k \in \Re^{n-k+1\times n-k+1}$ denotes the Householder matrix given by

$$\widetilde{\boldsymbol{H}}_k = \boldsymbol{I}_{n-k+1} - \frac{2}{\boldsymbol{v}_k^T\boldsymbol{v}_k}\boldsymbol{v}_k\boldsymbol{v}_k^T. \tag{6.15}$$

The vector $\boldsymbol{v}_k \in \Re^{n-k+1}$ is the *Householder vector* given by

$$\boldsymbol{v}_k = \boldsymbol{p}_{k,k}^{(k)} - \|\boldsymbol{p}_{k,k}^{(k)}\|_2\boldsymbol{e}_{n-k+1}, \tag{6.16}$$

where $\boldsymbol{e}_k$ is the first column of the identify matrix $\boldsymbol{I}_k$, and $\boldsymbol{p}_{i,j}^{(k)} \in \Re^{n-i+1}$ is a vector of elements of the matrix $\boldsymbol{P}^{(k)}$ such that

$$\boldsymbol{P}^{(k)} = \begin{bmatrix} p_{1,1}^{(k)} & \cdots & p_{1,j}^{(k)} & \cdots & p_{1,m}^{(k)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{i,1}^{(k)} & \cdots & \begin{bmatrix} \\ \boldsymbol{p}_{i,j}^{(k)} \\ \\ \end{bmatrix} & \cdots & \begin{bmatrix} \\ \boldsymbol{p}_{i,m}^{(k)} \\ \\ \end{bmatrix} \\ \vdots & & & & \\ p_{n,1}^{(k)} & \cdots & & \cdots & \end{bmatrix}, \tag{6.17}$$

and $\boldsymbol{P}^{(k)} \in \Re^{n \times m}$. Each multiplication by the matrix $\boldsymbol{H}^{(i)}$ zeroes all the elements below the diagonal in the i-th column which in consequence leads to the triangular matrix $\boldsymbol{R}$. Before the k-th step the algorithm searches for the column corresponding to the maximal Euclidean norm of the vectors $\boldsymbol{p}_{k,j}^{(k)}$, $j = k, \ldots, m$:

$$l = \underset{j=k,\ldots,m}{\arg\max} \|\boldsymbol{p}_{k,j}^{(k)}\|_2, \tag{6.18}$$

and then the k-th and l-th columns of $\boldsymbol{P}^{(k)}$ are interchanged. This operation generates the permutation matrix $\boldsymbol{\Pi}$.

The algorithm requires $4nmr - 2r^2(n+m) + 4r^3$ floating point operations. Additional details about the method as well as pseudocodes for the algorithms can be found in [29].

6.3 QR Factorization for Support Vector Reduction

The linearly independent subset of the support vectors as well as the coefficients c_{ij} can be found by applying the QR factorization with column pivoting to the *support vector matrix* given by

$$\boldsymbol{K} = \begin{bmatrix} K(\boldsymbol{y}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{y}_1, \boldsymbol{x}_m) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{y}_n, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{y}_n, \boldsymbol{x}_m) \end{bmatrix}, \tag{6.19}$$

where $\{x_i\}_{i=1}^m$ are the support vectors and $\{y_i\}_{i=1}^n$ are all the training samples. The computations are performed for a given value of the threshold parameter τ. This results in the permutation matrix $\boldsymbol{\Pi} \in \Re^{m \times m}$, the matrix of coefficients $\boldsymbol{C} \in \Re^{r \times m-r}$ presented in Eq. 6.12, and the number of the linearly independent support vectors $r \in \mathcal{N}$ (The rank of the matrix $\boldsymbol{K}$ with respect to the parameter τ). Consequently, we can express Eq. 6.8 using matrix notation as follows:

$$\boldsymbol{\Pi}\boldsymbol{\alpha}' = \begin{bmatrix} \boldsymbol{\alpha}'_1 \\ \boldsymbol{\alpha}'_2 \end{bmatrix} \quad \boldsymbol{\alpha}'_1 = \begin{bmatrix} \alpha'_1 \\ \vdots \\ \alpha'_r \end{bmatrix} \quad \boldsymbol{\alpha}'_2 = \begin{bmatrix} \alpha'_{r+1} \\ \vdots \\ \alpha'_m \end{bmatrix}$$

and

$$\begin{cases} \widetilde{\boldsymbol{\alpha}}'_1 = \boldsymbol{\alpha}'_1 + \boldsymbol{C}\boldsymbol{\alpha}'_2 \\ \widetilde{\boldsymbol{\alpha}}'_2 = \boldsymbol{0} \end{cases}, \tag{6.20}$$

where $\boldsymbol{\alpha}'$ is the vector of unsorted weights pre-multiplied by the class labels, the vector $\boldsymbol{\alpha}'_1$ contains the weights corresponding to the linearly independent support vectors, and $\boldsymbol{\alpha}'_2$ is a vector of weights corresponding to the dependent support vectors. The permutation matrix $\boldsymbol{\Pi}$ is used to sort the α'_i coefficients according to the degree of linear independence of the support vectors. Naturally, the columns of the matrix $\boldsymbol{K}$ and the support vectors are permuted in exactly the same way.

By substituting Eq. 6.12 into Eq. 6.20 we obtain the final equation which can be used to compute the weights of the reduced classifier:

$$\begin{cases} \widetilde{\boldsymbol{\alpha}}'_1 = \boldsymbol{\alpha}'_1 + \boldsymbol{R}_{11}^{-1}\boldsymbol{Q}_1^T\boldsymbol{K}_2\boldsymbol{\alpha}'_2 \\ \widetilde{\boldsymbol{\alpha}}'_2 = \mathbf{0} \end{cases} \tag{6.21}$$

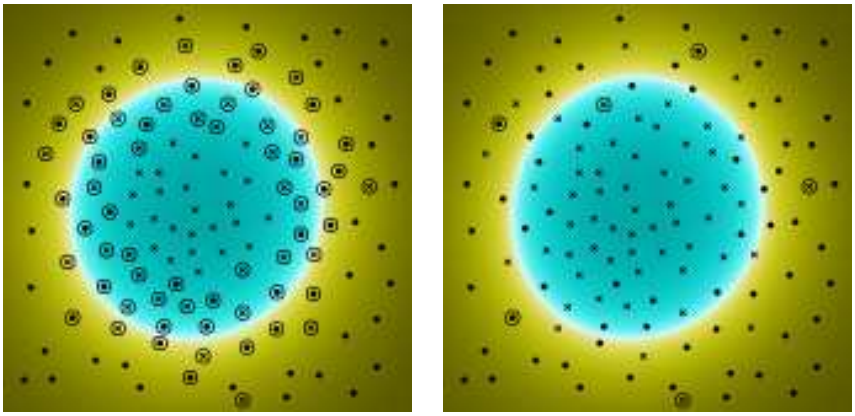
Additional normalization step can be performed before the QR factorization is computed. Namely, the matrix $\boldsymbol{K}$ can be normalized so that the Euclidean norm of each column is equal to one. This influences the column pivoting strategy. Naturally, the result of the factorization must be scaled afterwards in order to provide correct values of the c_{ij} coefficients.

As it was already stated, the parameter τ of the factorization algorithm can be seen as a measure of the linear independence between the first r columns of the matrix. Consequently, if applied to the support vector matrix K presented in Eq. 6.19 it allows to control the number of support vectors regarded as linearly independent in the feature space, and thus stored in the memory after the reduction. Clearly, as the value of τ grows, Eq. 6.6 becomes more and more an approximation of the exact solution. However, it is important to underline that the informative content of the discarded support vectors $\{\boldsymbol{x}_i\}_{i=r+1}^m$ is not completely lost, as their weights $\{\boldsymbol{\alpha}_i\}_{i=r+1}^m$ are used to compute the updated value of the weights $\widetilde{\alpha}_i$ for the remaining support vectors. This should result in a graceful decrease of classification performance compared to the optimal solution. Thus, the parameter τ can be used as an effective way to trade performance for memory requirements and speed during classification, depending on the task at hand.

A simple example illustrating the issues discussed above is shown in Figure 6.2. We see that the algorithm is able to provide considerable reduction ($\sim 90\%$) without any loss in generalization performance (compare Figure 6.2 upper left and upper right). Further reduction can be achieved by increasing the value of the threshold parameter τ. Although, in that case, it is possible to notice small variations of the decision surface, the reduction rate can be increased up to $\sim 97\%$.

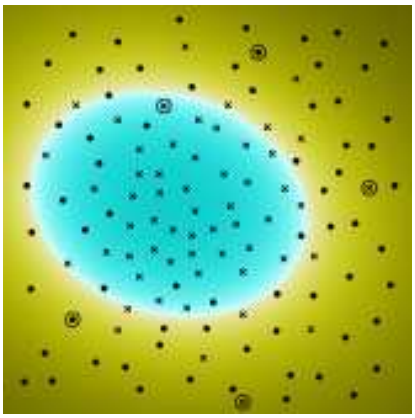
6.4 Summary

In this chapter, we described a method for exact simplification of the support vector solutions based on the fact that the set of support vectors is usually not linearly independent in the feature space [19]. We also showed that the QR factorization with column pivoting algorithm can be used to identify the linearly dependent support vectors. Finally, we extended the original simplification method by Downs *et al.* by introducing a parameter allowing to effectively trade the performance of the classifier for memory requirements and speed during classification. This extended version can be seen as an alternative approach to approximate SVM methods like [57, 12, 11] and is still able to provide an exact solution if desired. We will show in the next chapter that this can be exploited to achieve greater reduction without any loss in classification rate. Additional experiments presented in

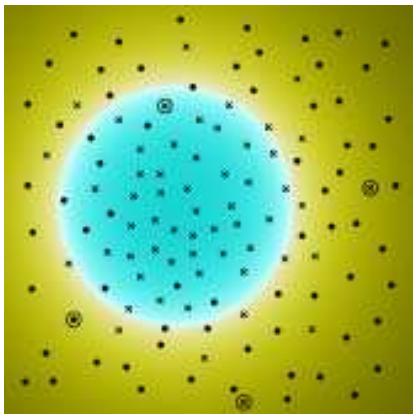


Not reduced (64 SVs)

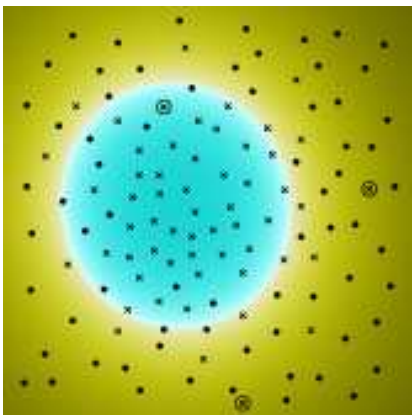
Reduced $\tau = 0.01$ (6 SVs)



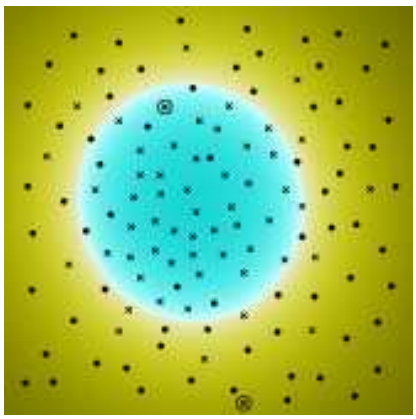
Reduced $\tau = 0.02$ (5 SVs)



Reduced $\tau = 0.04$ (4 SVs)



Reduced $\tau = 0.6$ (3 SVs)



Reduced $\tau = 0.6$ (2 SVs)

Figure 6.2. The result of applying the support vector reduction algorithm, with various values of threshold τ, to the classifier trained using two-dimensional data. The experiment was performed for the Gaussian kernel ($\gamma = 0.5$), and $C = 100$. The support vectors are marked with circles.

[63] revealed that the algorithm may also be used as a good selector for "the most important" support vectors.

Chapter 7

Experiments with Support Vector Reduction

In this chapter we present the results of an experimental evaluation of the support vector reduction algorithm described in detail in Chapter 6. The experiments reported here were conducted on the KTH-TIPS2 database [43] which was previously used for experiments with material categorization [13]. Additionally, the algorithm was evaluated in the domain of place recognition on the KTH-INDECS database described in Chapter 3. These results are, however, reported in Chapter 8. In every experiment, before the reduction was applied, the classifier was trained using the Sequential Minimal Optimization (SMO) [58, 14] algorithm.

The databases used in the experiments are of different complexity and contain visual data divided into multiple classes. In both cases, two different kinds of features were extracted from the images: MR8 [79] and Local Binary Patterns (LBP) [53] in case of the TIPS2 database and Composed Receptive Field Histograms (CRFH) [38] and local features (SIFT [41]) in case of the INDECS database (details about these descriptors can be found in Chapter 4). Moreover, various kernel types, multi-class SVM algorithms, and training parameters were tested. To the best of our knowledge, the original reduction algorithm proposed by Downs *et al.* [19] has been so far tried mainly on two-class problems[1] and non-visual data.

This chapter is organized as follows: Section 7.1 provides a brief description of the KTH-TIPS2 database used in experiments reported in succeeding sections. The experimental procedure is explained in detail in Section 7.2. In Section 7.3 we study several properties of the algorithm as well as the role of the threshold parameter in controlling the trade-off between the reduction rate and classification rate. Then, in Section 7.4, we examine the influence of the kernel parameters and the value of the parameter C on the performance of the reduction algorithm. Finally, in Section 7.5, we evaluate the algorithm on classifiers trained using various kernel types and multi-class methods. A summary is given in Section 7.6.

For space reasons, this chapter contains only a small selection of the available

[1] The Contraceptive Method Choice database is divided into three classes.

Figure 7.1. The variations within each category of the KTH-TIPS2 database (from Caputo *et al.* [13]). Each row shows one example image from each of four samples of a category. In addition, each sample was imaged under varying pose, illumination and scale conditions.

experimental results. The interested reader is referred to [60] for the complete set of results. The reduction algorithm was implemented on top of a modified version of the libSVM library [14]. This resulted in a new executable file that can be used in order to reduce a previously trained classifier.

7.1 The KTH-TIPS2 Database

The *KTH-TIPS2* database [43, 13] contains images of 4 planar samples of each of 11 materials (see Figure 7.1 for examples). Many of these materials have 3D structure, implying that their appearance can change considerably as pose and lighting are changed. TIPS2 contains images at 9 scales equally spaced logarithmically over two octaves. At each scale, materials were imaged under 3 poses (frontal, rotated 22.5° left and 22.5° right) and 4 illumination conditions (frontal, 45° from the top and 45° from the side, all taken with a desk-lamp with a Tungsten light bulb; the fourth illumination condition consisted of fluorescent lights). In total there are $9 \times 3 \times 4 = 108$ images per sample.

Two types of rotationally invariant descriptors were used in the experiments reported in succeeding sections: the *MR8 descriptor* [79] and *Local Binary Patterns* (*LBP*) [53]. The parameters of the descriptors were set to the same values as in the experiments with material categorization described in [13].

7.2 Experimental Setup

All the experiments reported in this chapter followed the same procedure, which can be divided into two steps. First, the classifier was trained using the SMO algorithm [58, 14] on certain training set. The support vectors were counted and the

classifier was evaluated on a test set in order to obtain the initial classification rate. Then, starting from the obtained discriminant functions, the reduction algorithm was applied for increasing values of the threshold parameter τ and the normalization turned on or off. The same value of threshold was always used for all discriminant functions of a multi-class classifier. This led to a progressive reduction in the number of support vectors and, after reaching certain threshold value, in the classification rate. After each reduction, the support vectors were counted and the performance of the classifier was evaluated on the test set. The process was stopped when the classification rate dropped below 70% of its initial value.

The reduction algorithm was evaluated on a classifiers trained using four different kernel types and several values of the kernel parameters: the χ^2 kernel ($\gamma = 10^{-3}, 10^{-2}, \ldots, 10^3$), the Gaussian kernel ($\gamma = 10^{-3}, 10^{-2}, \ldots, 10^3$), the polynomial kernel for $p = 0$ ($K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^d$, $d = 1, 2, \ldots, 5$), and the polynomial kernel for $p = 1$ ($K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y} + 1)^d$, $d = 1, 2, \ldots, 5$). Additionally, four different values of the C parameter were tested: 1, 10, 100, and 1000. The results for which the initial classification rate was lower than 30% were rejected. The experiments were also performed for two multi-class SVM algorithms: one-against-one and one-against-all.

For every experiment, the TIPS2 database was divided into a training and test set. The training set consisted of one, two, or three samples per material, while the test set was created from the remaining samples. In each case, four possible splittings were considered. As it was already mentioned, every experiment was repeated for both MR8 and LBP features. As a result, we performed $(2 \times 7 + 2 \times 5) \times 4 \times 2 \times 3 \times 4 \times 2 = 4608$ experiments in total. Naturally, this chapter presents only a small representative selection of the results. The remaining can be found in [60]. In particular, in the succeeding sections we will report only those results that were obtained for three samples per material in the training set.

As we consider multi-class problems, it is important to underline that the support vectors are counted in such way that each vector is taken into account only once, even if it is used by several discriminant functions. Consequently, the reported number of support vectors always correspond to the amount of memory needed to store the vectors and the number of required kernel evaluations in the test phase.

In most cases the results reported here (as well as in [60]) were averaged over all four possible splittings into the training and test sets. Consequently, they are presented in the form of the mean value accompanied by the uncertainty which is always one standard deviation.

7.3 Parameters of the Support Vector Reduction Algorithm

The support vector reduction algorithm is parametrized by the threshold τ. Additionally, as it was mentioned in Chapter 6, the support vector matrix can be normalized before the reduction process begins. This section studies the influence of the increasing value of threshold to the amount of reduction and the performance of the classifier.

Figure 7.2 presents the relationships between the reduction rates and the classification rates as well as between the value of the threshold and the reduction rates obtained during experiments conducted for two feature types and two different kernels. It can be observed from Figure 7.2, left, that in case of the polynomial kernel, reducing $\sim$ 40% percent of the support vectors did not cause any variations in the classification rate. We also see from Figure 7.2, right, that these vectors were considererd as linearly dependent with respect to the threshold nearly equal to 0. Thus, this part of the reduction process can be regarded as an exact simplification. On the same basis, in case of the χ^2 kernel, it would be possible to achieve the reduction rates of only $\sim 5\%$ (LBP) or even less (MR8). We can also observe that relatively bigger threshold is required in order to reduce even small amount of support vectors. This suggests that the set of vectors is in this case "more" linearly independent. This is consistent with the results published by Downs *et al.* [19]. They report lower reduction rates for the Gaussian kernel than for the polynomial kernel.

Figure 7.2 indicates that it is possible to achieve much higher reduction rates without any loss in classification rate, even if the χ^2 kernel was used in order to train the classifier ($\sim 50\%$ and $\sim 40\%$ for the χ^2 kernel, and $\sim 70\%$ for the polynomial kernel). However, we need to modify the condition of linear dependence by increasing the threshold value. In that case, the resulting classifier can be seen as an approximation of the initial solution. It will be shown in the next sections that this way it is possible to achieve similar reduction rates for all kernel types. What is more, if the aim is to meet certain requirements regarding the memory requirements or the speed in the test phase, the number of support vectors can be further decreased.

Additional conclusions can be drawn from the plots in Figure 7.2. First of all, there usually exist a point where the classification rate is higher than the initial value. This may suggest that the simplification process may lead to a classifier of better generalization performance. Irregular variations can also be seen in the plots. This can be explained by the fact that a multi-class classifier is built from several discriminant functions which may react differently to the reduction. In spite of that, the experiments reported in the next sections prove that the reduction algorithm can be successfully applied to multi-class problems.

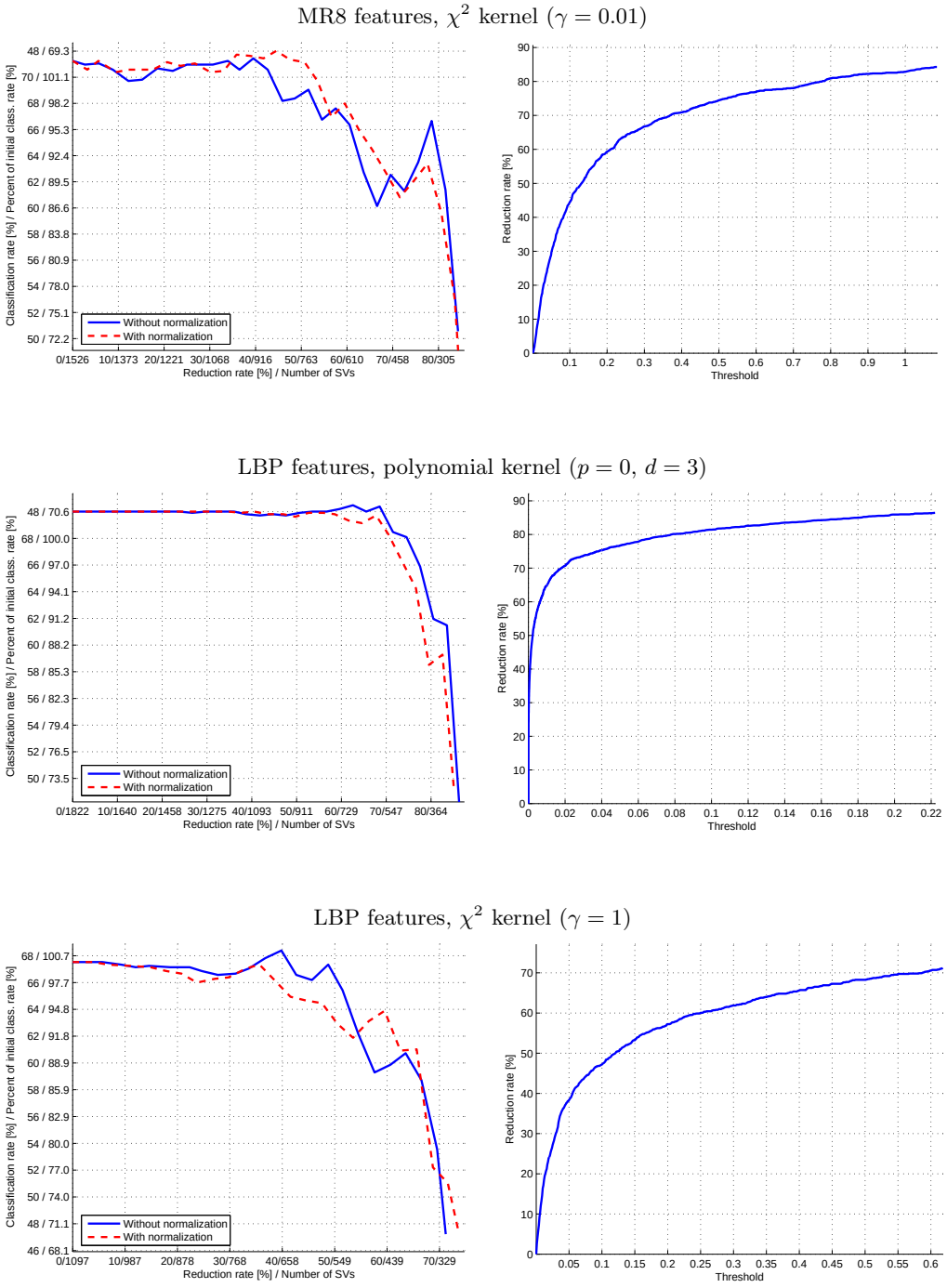


Figure 7.2. Relationship between the reduction rate and classification rate (left) as well as between the value of the threshold parameter τ and the reduction rate (right) for various feature and kernel types. The plots on the right were obtained with the normalization turned off.

7.4 Experiments with Kernel and Training Parameters

As it was already stated, the experiments were conducted for various values of the kernel parameters and the parameter C. Tables 7.1 and 7.2 show all the results obtained for the MR8 features, the Gaussian kernel, and the one-against-one multi-class extension. The results were averaged over the four possible splittings into the training and test sets, as it was described in Section 7.2. The tables present the classification rates that are guaranteed to be preserved and the maximal reduction rates which can be achieved under such constraint. These dependencies between the classification rates and the number of support vectors are illustrated in the form of plots in Figures 7.3 and 7.4

First of all, it can be seen that the reduction rate depends strongly on the parameters of the kernel and decreases as γ grows. The same property holds for the χ^2 kernel and the polynomial kernel for growing d. Similar results were also reported by Downs *et al.* [19].

It can be observed from the plots, that the smallest number of support vectors is usually obtained as a result of reduction of the best trained classifier. In other words, even if the aim is to achieve large reduction not high classification rate, it is better to use the strongest classifier as a starting point. This suggests that the classifiers that are more sensitive to the reduction also suffer from overfitting (see the plot for $\gamma = 10$). Similar behavior was also observed for other kernel types. Although, this property may not always hold exactly, in general it is a better choice to perform the reduction starting from the best trained classifier. Consequently, in the next section we will report the results obtained for the classifier with the highest initial classification rate.

7.5 Experiments with Kernel Type and Multi-class SVM Algorithms

Tables 7.3 and 7.4 presents the results of the evaluation of the reduction algorithm applied to the classifier trained on two feature types using various kernels and the one-against-one multi-class algorithm. Corresponding results obtained for the one-against-all multi-class extension are given in Tables 7.5 and 7.6. In both cases, C was equal 100 and the best kernel parameters were determined by cross validation.

First, the algorithm was able to provide the reduction rates up to $\sim 83\%$ without affecting the classification rate of the resulting classifier. In general, about half of the vectors were excluded from the final solution. Additional 5 to 10 percent of reduction can be achieved if a 2 percent loss in the classification rate is accepted.

It is difficult to find any stable relationship between the kernel type and the amount of reduction. Consequently, we can repeat after [19] that this value is both kernel and problem dependent and does not appear to be predictable *a priori*. Interesting observation can be made on the basis of the results obtained for the LBP

γ	Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
0.001	ORIGINAL		58.86 ± 3.07	—	2957 ± 54
	R E D.	100	58.85 ± 3.07	78.07 ± 2.30	649 ± 76
		98	57.68 ± 3.01	82.16 ± 2.97	528 ± 98
		95	55.92 ± 2.91	85.69 ± 1.19	422 ± 29
		90	52.97 ± 2.76	88.24 ± 1.25	346 ± 35
		80	47.09 ± 2.45	90.87 ± 1.35	269 ± 37
0.01	ORIGINAL		69.62 ± 5.37	—	2001 ± 61
	R E D.	100	69.61 ± 5.37	54.32 ± 12.80	918 ± 271
		98	68.23 ± 5.26	69.67 ± 3.46	608 ± 85
		95	66.14 ± 5.10	74.55 ± 3.84	511 ± 88
		90	62.66 ± 4.83	78.93 ± 3.09	423 ± 71
		80	55.70 ± 4.30	82.87 ± 1.80	343 ± 41
0.1	ORIGINAL		69.35 ± 5.99	—	1383 ± 55
	R E D.	100	69.34 ± 5.99	50.28 ± 6.42	689 ± 109
		98	67.97 ± 5.87	55.44 ± 5.19	618 ± 95
		95	65.89 ± 5.69	60.83 ± 2.92	542 ± 61
		90	62.42 ± 5.39	68.82 ± 4.24	432 ± 72
		80	55.48 ± 4.79	73.36 ± 3.88	369 ± 64
1	ORIGINAL		69.32 ± 6.67	—	1518 ± 34
	R E D.	100	69.30 ± 6.67	46.95 ± 7.03	805 ± 111
		98	67.93 ± 6.54	50.98 ± 3.31	744 ± 56
		95	65.85 ± 6.34	59.43 ± 4.65	616 ± 77
		90	62.39 ± 6.01	64.46 ± 5.13	539 ± 82
		80	55.45 ± 5.34	71.82 ± 5.24	428 ± 83
10	ORIGINAL		62.36 ± 8.40	—	2655 ± 33
	R E D.	100	62.35 ± 8.39	22.44 ± 5.71	2061 ± 175
		98	61.12 ± 8.23	36.75 ± 7.09	1681 ± 209
		95	59.25 ± 7.98	45.67 ± 1.93	1442 ± 69
		90	56.13 ± 7.56	57.21 ± 5.08	1136 ± 135
		80	49.89 ± 6.72	76.93 ± 6.00	611 ± 159

Table 7.1. Average results of the evaluation of the reduction algorithm on the MR8 features. The classifier was trained using the one-against-one multi-class algorithm, the Gaussian kernel with various values of γ and $C = 100$.

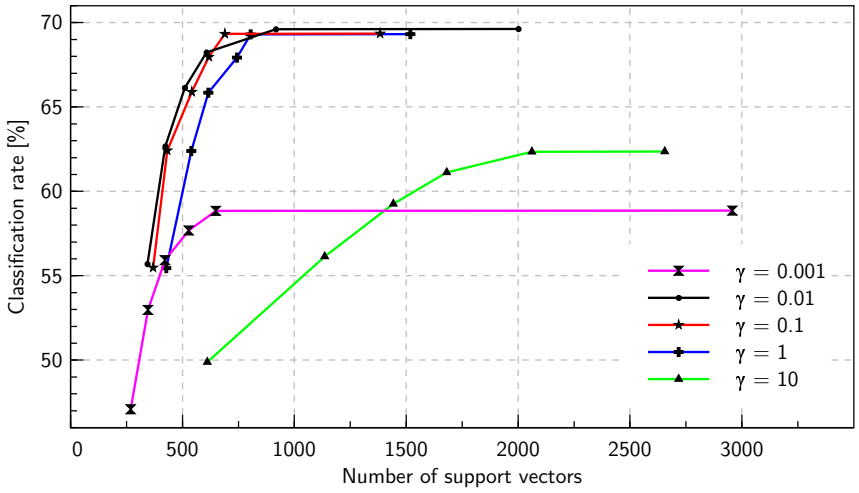


Figure 7.3. Illustration of the results given in Table 7.1.

C	Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
1	ORIGINAL		40.09 ± 3.15	0.00 ± 0.00	3436 ± 53
	RED.	100	40.08 ± 3.15	82.77 ± 18.78	589 ± 638
		98	39.29 ± 3.09	92.44 ± 3.30	260 ± 113
		95	38.09 ± 2.99	94.27 ± 2.68	196 ± 91
		90	36.08 ± 2.84	95.80 ± 1.03	144 ± 35
		80	32.07 ± 2.52	96.94 ± 0.60	105 ± 22
10	ORIGINAL		58.84 ± 3.09	0.00 ± 0.00	2959 ± 53
	RED.	100	58.82 ± 3.09	75.35 ± 3.25	730 ± 104
		98	57.66 ± 3.03	82.34 ± 2.56	523 ± 85
		95	55.89 ± 2.94	86.28 ± 1.17	405 ± 29
		90	52.95 ± 2.78	87.97 ± 1.34	355 ± 37
		80	47.07 ± 2.47	91.10 ± 1.14	262 ± 31
100	ORIGINAL		69.62 ± 5.37	0.00 ± 0.00	2001 ± 61
	RED.	100	69.61 ± 5.37	54.32 ± 12.80	918 ± 271
		98	68.23 ± 5.26	69.67 ± 3.46	608 ± 85
		95	66.14 ± 5.10	74.55 ± 3.84	511 ± 88
		90	62.66 ± 4.83	78.93 ± 3.09	423 ± 71
		80	55.70 ± 4.30	82.87 ± 1.80	343 ± 41
1000	ORIGINAL		69.12 ± 5.96	0.00 ± 0.00	1366 ± 49
	RED.	100	69.10 ± 5.96	48.84 ± 3.53	698 ± 60
		98	67.74 ± 5.84	58.82 ± 6.15	563 ± 93
		95	65.66 ± 5.66	63.00 ± 4.48	506 ± 73
		90	62.21 ± 5.36	68.79 ± 4.76	427 ± 78
		80	55.29 ± 4.77	73.82 ± 4.64	359 ± 75

Table 7.2. Average results of the evaluation of the reduction algorithm on the MR8 features. The classifier was trained using the one-against-one multi-class algorithm, the Gaussian kernel with $\gamma = 0.01$ and various values of C.

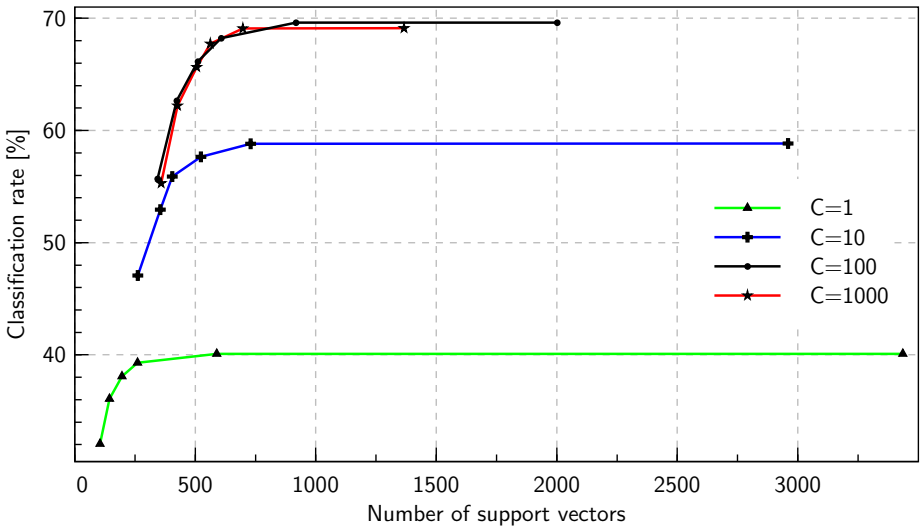


Figure 7.4. Illustration of the results given in Table 7.2.

descriptor and both multi-class algorithms. We see that although the initial number of support vectors is much larger in the case of the one-against-all method, the reduction leads to solutions containing similar number of vectors for both algorithms. Other experiments indicate that in general the reduction rate is higher for the one-against-all method.

7.6 Summary

In this chapter we presented a selection of results of a comprehensive experimental evaluation of the support vector reduction algorithm described in Chapter 6. The results were carefully chosen so that they were representative and led to conclusions that are true also for experiments not reported here.

First of all, we showed how the parameters of the algorithm influence the reduction process. Then, we presented a series of results proving that our method can be successfully applied to classifiers trained on multi-class visual data, and it performs well for various kernel types and multi-class algorithms. In each case, the algorithm provided substantial reduction in the number of support vectors. The threshold parameter can be used to tune the trade-off between the amount of reduction and the classification rate. This can be exploited to either reduce the complexity of the classifier without any loss in its performance or to make it compatible with certain requirements regarding the number of support vectors. Moreover, the results indicate that small decrease in performance may lead to much larger reduction in the number of support vectors.

In conclusion, the algorithm evaluated in this chapter can be a powerful and flexible method for decreasing the memory requirements and increasing the speed in the test phase of the Support Vector Machine classifier.

Kernel	Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
χ^2 $\gamma = 0.01$	ORIGINAL		72.28 ± 6.10	—	1525 ± 36
	REDUCED	100	72.27 ± 6.09	45.69 ± 21.25	833 ± 340
		98	70.83 ± 5.97	53.77 ± 10.94	708 ± 180
		95	68.67 ± 5.79	60.38 ± 6.70	606 ± 114
		90	65.05 ± 5.49	68.89 ± 4.64	475 ± 80
		80	57.82 ± 4.88	78.01 ± 3.73	336 ± 62
Gaussian $\gamma = 0.01$	ORIGINAL		69.62 ± 5.37	—	2001 ± 61
	REDUCED	100	69.61 ± 5.37	54.32 ± 12.80	918 ± 271
		98	68.23 ± 5.26	69.67 ± 3.46	608 ± 85
		95	66.14 ± 5.10	74.55 ± 3.84	511 ± 88
		90	62.66 ± 4.83	78.93 ± 3.09	423 ± 71
		80	55.70 ± 4.30	82.87 ± 1.80	343 ± 41
Polynomial $p = 0$ $d = 4$	ORIGINAL		68.85 ± 7.62	—	1719 ± 30
	REDUCED	100	68.84 ± 7.62	38.19 ± 7.99	1064 ± 150
		98	67.47 ± 7.47	45.56 ± 7.10	935 ± 125
		95	65.41 ± 7.24	50.13 ± 7.09	856 ± 122
		90	61.97 ± 6.86	59.69 ± 6.86	692 ± 118
		80	55.08 ± 6.10	69.38 ± 5.40	526 ± 95
Polynomial $p = 1$ $d = 5$	ORIGINAL		69.09 ± 6.87	—	1543 ± 31
	REDUCED	100	69.08 ± 6.87	43.60 ± 7.30	870 ± 115
		98	67.71 ± 6.74	50.15 ± 6.68	768 ± 103
		95	65.64 ± 6.53	56.58 ± 4.95	670 ± 79
		90	62.18 ± 6.19	62.26 ± 5.49	582 ± 89
		80	55.27 ± 5.50	71.35 ± 5.67	442 ± 91

Table 7.3. Average results of the evaluation of the reduction algorithm on the MR8 features. The classifier was trained using the one-against-one multi-class algorithm, various kernel types and $C = 100$.

Kernel	Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
χ^2 $\gamma = 1$	ORIGINAL		71.05 ± 2.16	—	1105 ± 27
	REDUCED	100	71.03 ± 2.16	36.66 ± 12.37	698 ± 125
		98	69.63 ± 2.11	45.31 ± 4.43	604 ± 59
		95	67.50 ± 2.05	53.10 ± 4.68	517 ± 47
		90	63.94 ± 1.94	59.28 ± 4.18	449 ± 45
		80	56.84 ± 1.73	66.87 ± 5.72	364 ± 57
Gaussian $\gamma = 10$	ORIGINAL		71.67 ± 2.78	—	1077 ± 9
	REDUCED	100	71.65 ± 2.78	44.51 ± 11.71	598 ± 131
		98	70.23 ± 2.72	53.85 ± 1.94	497 ± 23
		95	68.08 ± 2.64	57.16 ± 2.14	461 ± 26
		90	64.50 ± 2.50	61.07 ± 2.98	419 ± 34
		80	57.33 ± 2.22	66.40 ± 3.21	361 ± 35
Polynomial $p = 0$ $d = 3$	ORIGINAL		70.45 ± 1.74	—	1823 ± 17
	REDUCED	100	70.44 ± 1.74	68.03 ± 5.30	581 ± 91
		98	69.04 ± 1.70	73.08 ± 3.25	490 ± 55
		95	66.93 ± 1.65	78.01 ± 2.52	400 ± 43
		90	63.41 ± 1.56	79.78 ± 2.00	367 ± 33
		80	56.36 ± 1.39	83.19 ± 2.05	305 ± 35
Polynomial $p = 1$ $d = 5$	ORIGINAL		71.54 ± 3.04	—	896 ± 8
	REDUCED	100	71.53 ± 3.04	42.64 ± 10.23	513 ± 91
		98	70.11 ± 2.97	49.73 ± 4.20	450 ± 37
		95	67.97 ± 2.88	54.25 ± 2.91	409 ± 28
		90	64.39 ± 2.73	57.54 ± 3.71	379 ± 34
		80	57.23 ± 2.43	63.66 ± 2.61	325 ± 25

Table 7.4. Average results of the evaluation of the reduction algorithm on the LBP features. The classifier was trained using the one-against-one multi-class algorithm, various kernel types and $C = 100$.

Kernel	Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
χ^2 $\gamma = 0.01$	ORIGINAL		72.88 ± 6.86	0.00 ± 0.00	1733 ± 48
	REDUCED	100	72.87 ± 6.86	44.97 ± 25.51	962 ± 452
		98	71.42 ± 6.73	54.03 ± 20.21	804 ± 361
		95	69.24 ± 6.52	66.57 ± 13.02	583 ± 231
		90	65.59 ± 6.18	75.03 ± 8.89	434 ± 157
		80	58.30 ± 5.49	86.12 ± 2.60	241 ± 50
Gaussian $\gamma = 0.1$	ORIGINAL		69.95 ± 6.14	0.00 ± 0.00	1665 ± 53
	REDUCED	100	69.93 ± 6.14	52.18 ± 7.03	795 ± 118
		98	68.55 ± 6.02	61.94 ± 3.42	634 ± 69
		95	66.45 ± 5.84	68.70 ± 5.13	521 ± 90
		90	62.95 ± 5.53	76.62 ± 3.35	389 ± 59
		80	55.96 ± 4.92	83.57 ± 3.27	274 ± 62
Polynomial $p = 0$ $d = 2$	ORIGINAL		69.53 ± 7.40	0.00 ± 0.00	1447 ± 40
	REDUCED	100	69.52 ± 7.40	50.39 ± 9.83	721 ± 158
		98	68.14 ± 7.25	57.75 ± 8.82	613 ± 140
		95	66.06 ± 7.03	67.26 ± 11.30	477 ± 173
		90	62.58 ± 6.66	78.29 ± 6.78	316 ± 104
		80	55.63 ± 5.92	85.94 ± 5.45	205 ± 83
Polynomial $p = 1$ $d = 4$	ORIGINAL		69.50 ± 7.01	0.00 ± 0.00	1489 ± 34
	REDUCED	100	69.48 ± 7.00	44.61 ± 11.07	826 ± 173
		98	68.11 ± 6.87	52.06 ± 6.67	715 ± 109
		95	66.02 ± 6.66	63.86 ± 4.42	538 ± 73
		90	62.55 ± 6.31	77.58 ± 7.93	335 ± 123
		80	55.60 ± 5.60	83.30 ± 7.17	250 ± 110

Table 7.5. Average results of the evaluation of the reduction algorithm on the MR8 features. The classifier was trained using the one-against-all multi-class algorithm, various kernel types and $C = 100$.

Kernel	Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
χ^2 $\gamma = 1$	ORIGINAL		72.89 ± 4.80	± 0.00	1579 ± 69
	REDUCED	100	72.88 ± 4.79	45.98 ± 15.46	853 ± 249
		98	71.43 ± 4.70	61.17 ± 4.41	611 ± 64
		95	69.25 ± 4.56	69.19 ± 4.82	483 ± 59
		90	65.60 ± 4.32	76.14 ± 1.13	377 ± 31
		80	58.31 ± 3.84	82.33 ± 1.86	279 ± 38
Gaussian $\gamma = 10$	ORIGINAL		72.85 ± 4.24	0.00 ± 0.00	1658 ± 57
	REDUCED	100	72.83 ± 4.24	64.98 ± 4.63	579 ± 74
		98	71.39 ± 4.15	71.01 ± 3.41	480 ± 59
		95	69.20 ± 4.03	74.56 ± 3.51	420 ± 56
		90	65.56 ± 3.82	78.44 ± 3.08	357 ± 53
		80	58.28 ± 3.39	82.88 ± 2.62	283 ± 43
Polynomial $p = 0$ $d = 5$	ORIGINAL		71.66 ± 4.67	0.00 ± 0.00	2422 ± 47
	REDUCED	100	71.64 ± 4.66	83.03 ± 5.27	410 ± 127
		98	70.22 ± 4.57	87.39 ± 1.99	305 ± 50
		95	68.07 ± 4.43	89.55 ± 1.40	253 ± 35
		90	64.49 ± 4.20	90.68 ± 0.85	225 ± 22
		80	57.32 ± 3.73	92.14 ± 0.49	190 ± 13
Polynomial $p = 1$ $d = 5$	ORIGINAL		72.52 ± 4.65	0.00 ± 0.00	1671 ± 74
	REDUCED	100	72.50 ± 4.65	65.22 ± 5.74	576 ± 70
		98	71.07 ± 4.55	71.20 ± 3.90	478 ± 46
		95	68.89 ± 4.41	76.58 ± 2.71	389 ± 39
		90	65.27 ± 4.18	81.12 ± 2.60	315 ± 46
		80	58.01 ± 3.72	86.04 ± 1.63	232 ± 23

Table 7.6. Average results of the evaluation of the reduction algorithm on the LBP features. The classifier was trained using the one-against-all multi-class algorithm, various kernel types and $C = 100$.

Chapter 8

Experiments with Place Recognition

Previous chapters presented the structure of our visual indoor place recognition system as well as provided details about each of its parts. In this chapter we report the results of an extensive experimental evaluation of the system on the KTH-INDECS database, specially designed for this purpose. Moreover, we show that the support vector reduction algorithm introduced in this thesis may provide an efficiency gain even for complex problems such as place recognition.

Our place recognition system is built around the Support Vector Machine classifier [78, 17]. We evaluate both local and global descriptors in order to find the image representation that is best suited for the place recognition purposes. We use the Composed Receptive Field Histograms [38] as global image representation and the combination of the Harris-Laplace detector [47] and the SIFT descriptor [42] to extract local features. Various parameters of the descriptors were modified in the experiments. In particular, several different receptive fields and their combinations have been tested. The descriptors are coupled to the SVM classifier using specialized kernel functions: the χ^2 kernel [79] and the local kernels introduced in [82]. Finally, we employ our support vector reduction algorithm in order to improve the efficiency of the system. Since the local kernels do not satisfy the Mercer's theory, this chapter provides an answer to the question of whether the reduction algorithm can be successfully applied for non-Mercer's kernels.

In designing the system, special emphasis has been placed on the robustness to variations that may occur in real-world environments. This motivated the creation of the KTH-INDECS database, on which the system was tested (the database is described in detail in Chapter 3). In short, the database contains pictures of five rooms of different functionality imaged from various viewpoints and locations. For each room, the pictures were acquired at different times of the day, under various weather conditions, across a span of time of more than two months. All this ensures that the performance of the system was evaluated in the presence of illumination variations and normal activity that occur in the rooms (people appear in the rooms, pieces of furniture and objects are moved over time). Moreover, the experiments were designed in a way allowing to test the robustness and generalization abilities of

the system, since training and testing were always performed on pictures acquired under different illumination conditions.

The experiments reported in this chapter were divided into two parts depending on the type of descriptor used to extract the features from the pictures of places. The chapter starts with a description of those parts of the experimental procedure that were common for all experiments. Then, in Section 8.2, the performance of the system is evaluated for local features. The results of experiments with global features are presented in Section 8.3. This chapter concludes with a summary in Section 8.4.

For space reasons, this chapter presents only a summary of the available experimental results. Detailed results can be found in [61].

8.1 Experimental Setup

All the experiments reported in this chapter were performed on the pictures included in the KTH-INDECS database resampled to the resolution of 512x384 pixels. The database was divided into training and test sets with respect to the room in which the pictures were taken as well as to the illumination conditions. For every experiment, the system was trained using all the pictures acquired under one illumination condition and tested using the remaining pictures divided into 10 test sets. Table 8.1 presents the combinations of the training and test sets used in the experiments. The fact that the test data is divided into subsets allowed to evaluate the performance of the system separately for each room and illumination condition. In order to calculate a single measure of performance, all the results obtained for one training set were averaged with equal weights. Consequently, each room was equally important independently of the number of pictures used for testing.

A modified version of the libSVM library [14] extended to the one-against-all multi-class algorithm, implementation of the local kernels[1], and the support vector reduction algorithm was used in the experiments. The library implements the Sequential Minimal Optimization (SMO) [58] training algorithm.

Since, the experimental process differs for the local and global features, it will be described in detail in the following sections.

8.2 Experiments with Local Descriptors

As it was already mentioned, the local image features were obtained using the Harris-Laplace interest point detector [47] and the SIFT descriptor [41] (both algorithms are described in Section 4.3). The SIFT descriptor is widely used for object classification and recognition (see e.g. [18]) and was also shown to perform well for geometric mobile robot localization ([69, 2]). The image representation based on

[1]The modified version of the libSVM library extended to the one-against-all multi-class algorithm and implementation of the local kernels was kindly provided by Barbara Caputo.

Training set		Test sets		
Illumination conditions	No. of pictures	Room	Illumination conditions	No. of pictures
Cloudy	1092	Barbara's office (BO)	Night	108
			Sunny	108
		Corridor (CR)	Night	384
			Sunny	384
		Elin's office (EO)	Night	168
			Sunny	168
		Kitchen (KT)	Night	216
			Sunny	216
		Surr. of the printer (PR)	Night	216
			Sunny	216
Night	1092	Barbara's office (BO)	Cloudy	108
			Sunny	108
		Corridor (CR)	Cloudy	384
			Sunny	384
		Elin's office (EO)	Cloudy	168
			Sunny	168
		Kitchen (KT)	Cloudy	216
			Sunny	216
		Surr. of the printer (PR)	Cloudy	216
			Sunny	216
Sunny	1080	Barbara's office (BO)	Cloudy	108
			Night	108
		Corridor (CR)	Cloudy	384
			Night	384
		Elin's office (EO)	Cloudy	156
			Night	156
		Kitchen (KT)	Cloudy	216
			Night	216
		Surr. of the printer (PR)	Cloudy	216
			Night	216

Table 8.1. Training and test sets used in the experiments with place recognition.

local descriptors requires using a specialized kernel with the Support Vector Machines. In our experiments, we employed the local kernels proposed by Wallraven *et al.* [82] (see Section 5.2). The kernel compares two feature vectors by performing matching of the local descriptors, and the sum of Euclidean distances between the descriptors determine the value of the kernel function.

The experiments with local features consisted of two parts. First, the optimal kernel parameters and the value of the parameter C were estimated using the cross-

Training set			Cloudy	Night	Sunny
Parameters			$\gamma = 2.585$ $C = 100$	$\gamma = 2.585$ $C = 100$	$\gamma = 2.239$ $C = 100$
Test set	BO	Cloudy	—	48.98 %	30.61 %
		Night	52.58 %	—	34.02 %
		Sunny	28.87 %	27.84 %	—
	CR	Cloudy	—	97.53 %	96.70 %
		Night	97.53 %	—	93.41 %
		Sunny	97.24 %	93.65 %	—
	EO	Cloudy	—	83.44 %	76.43 %
		Night	88.31 %	—	76.62 %
		Sunny	73.79 %	73.79 %	—
	KT	Cloudy	—	68.39 %	78.76 %
		Night	77.08 %	—	63.54 %
		Sunny	84.97 %	70.47 %	—
	PR	Cloudy	—	63.03 %	67.30 %
		Night	69.67 %	—	52.13 %
		Sunny	66.51 %	48.80 %	—
Avg. classification rate			73.65 %	67.59 %	66.95 %

Table 8.2. Final results of the experiments with indoor place recognition and local descriptors.

validation technique. The results are given in Section 8.2.1. Then, the support vector reduction algorithm was applied to the obtained solution. This part of the experiments is discussed in Section 8.2.2.

8.2.1 Evaluation of the Performance of the System

In order to test the robustness and generalization abilities of the system, training was always performed on all the pictures acquired under similar illumination conditions and testing was done on the remaining pictures (see Table 8.1 and Section 8.1 for a description of the three splits into the training and test sets used in the experiments). Additionally, the pictures containing less than 5 interest points were rejected from all the training and test sets.

The optimal parameters of the local kernel as well as the value of the parameter C were estimated using the hold out cross-validation method separately for each training set. The average over the classification rates obtained for each test subset was used as a measure of performance, as it was described in Section 8.1. The final results of the experiments, together with the parameters used during training, are presented in Table 8.2.

The results indicate that the highest classification rates were obtained for the *cloudy* training set (73.65%). The explanation for this is straightforward: the illu-

Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
ORIGINAL		69.40 ± 3.70	—	968 ± 17
REDUCED	100	69.40 ± 3.70	3.90 ± 6.55	929 ± 56
	99	68.71 ± 3.66	10.07 ± 4.26	870 ± 38
	98	68.01 ± 3.62	16.74 ± 4.57	805 ± 46
	97	67.32 ± 3.59	24.79 ± 1.56	728 ± 28
	96	66.62 ± 3.55	27.47 ± 1.80	702 ± 28
	95	65.93 ± 3.51	31.50 ± 2.10	663 ± 23
	90	62.46 ± 3.33	49.16 ± 5.71	492 ± 63
	85	58.99 ± 3.14	59.63 ± 2.16	390 ± 26
	80	55.52 ± 2.96	70.63 ± 3.39	284 ± 35

Table 8.3. Average results of the experiments with support vector reduction algorithm applied to the classifier trained on the local features extracted from the KTH-INDECS database. The uncertainties are given as one standard deviation.

mination conditions on a cloudy day can be seen as intermediate between those at night (only artificial light) and on a sunny day (direct natural light dominates). The same conclusion can be drawn from the analysis of the classification rates for other training sets. It can be observed that the system trained on the *night* or *sunny* training data usually performs best on the pictures acquired on a cloudy day.

There are large differences between the classification rates for individual rooms. In particular, the corridor can be seen as an attractor, while Barbara's office was relatively rarely recognized properly. This is not only a consequence of an unbalanced training set (384 pictures in case of the corridor, and 108 pictures in case of the Barbara's office), but also results from the complexity of the problem. Since the rooms are physically separated by sliding glass doors, and in case of the surroundings of the printer and the corridor the boundary has been chosen arbitrarily, many pictures labeled as the corridor in fact image another room.

8.2.2 Experiments with Support Vector Reduction

The experiments with support vector reduction were conducted in a similar manner to those described in Chapter 7. First, the classifier was trained on a training set using the optimal settings determined in experiments reported in the previous section. The support vectors were counted and the classifier was evaluated on a test set in order to obtain the initial classification rate. Then, the reduction algorithm was applied for increasing values of the threshold parameter τ and the normalization turned on or off. After each reduction, the support vectors were counted and the performance of the classifier was evaluated on the test sets. The process was stopped when the classification rate dropped below 70% of its initial value.

Table 8.3 presents the obtained relationship between the reduction rate and

the classification rate that was guaranteed to be preserved after the reduction. The results were averaged over the three training sets: *cloudy*, *night*, and *sunny*. Comparing the results to those reported in Chapter 7, we see that lower reduction rates were obtained if the aim was to preserve the initial classification rate. In other words, the support vectors were "less" linearly dependent. However, the algorithm was still able to provide substantial reduction at a cost of small decrease in the performance of the classifier. Additionally, the experiments revealed that the support vector reduction algorithm can be successfully applied to classifiers trained using a non-Mercer's kernel.

8.3 Experiments with Global Descriptors

Most of the currently available approaches to the place recognition problem make use of global descriptors (see Chapter 4 for more information). This is consistent with the results of studies on human scene perception which suggest that people prefer to use coarse global information during the first glance at a scene. In this section we evaluate the performance of an indoor place recognition system based on multi-dimensional histograms of responses of several basic image descriptors (CRFH [38]). We built the histograms using various combinations of descriptors applied to the scale-space representation of an image at various scales. The χ^2 measure was used in order to calculate the distance between the histograms.

The experiments with global features were organized as follows: first, various combinations of image descriptors and scales were tested. Additionally, the optimal number of histogram quantization levels was determined for several descriptors (see Section 8.3.1). The best performing combination of descriptors was chosen and the values of the γ and C parameters were precisely estimated by cross-validation (see Section 8.3.2). Finally, the support vector reduction algorithm was applied to the resulting solution (see Section 8.3.3).

8.3.1 Experiments with Descriptor Parameters

As it was already stated, the Composed Receptive Field Histograms used in the experiments were built on the basis of several combinations of basic image descriptors. The complete list of these combinations is presented in Table 8.4. All derivative-based descriptors were normalized; more details can be found in Chapter 4. The descriptors were applied to the scale-space representation of an image at various scales; however, the same scales were always used for all the descriptors used to build one histogram. We tried all the combinations of one, two, three, and four scales from the following set: $\{1, 2, 3, 4, 5\}$. Each experiment was repeated for 13 values of the γ parameter $(10^{-3}, 10^{-2.5}, \ldots, 10^{3})$, $C = 100$, and three splits into the training and test sets. The best results for each combination of descriptors are presented in Table 8.4.

Similarly to the experiments with local features, the highest classification rate was obtained for the *cloudy* training set. It can be observed that using histograms

Descriptors	Scales	Dim.	Class. rate / tr. set [%]			Average class. rate [%]
			Cloudy	Night	Sunny	
L	1,2,3,5	4	55.26	49.08	52.50	52.28 ± 3.10
L_x, L_y	1,2,4	6	76.33	65.73	67.91	69.99 ± 5.60
L_{xx}, L_{xy}, L_{yy}	1,4	6	72.77	65.94	68.13	68.95 ± 3.49
L_x, L_y, L_{xy}	1,4	6	75.10	67.24	68.93	70.42 ± 4.14
L_x, L_y, L_{xx}, L_{yy}	1,5	8	77.20	69.61	70.40	72.40 ± 4.17
L_x, L_y, L_{xx}, L_{xy}, L_{yy}	2	5	74.20	65.12	65.82	68.38 ± 5.05
C_1, C_2	1	2	34.11	36.93	34.99	35.34 ± 1.45
$C_{1,x}$, $C_{1,y}$, $C_{2,x}$, $C_{2,y}$	1,2	8	56.29	50.15	50.84	52.42 ± 3.36
$C_{1,xx}$, $C_{1,xy}$, $C_{1,yy}$, $C_{2,xx}$, $C_{2,xy}$, $C_{2,yy}$	2	6	48.93	43.11	43.02	45.02 ± 3.38
$C_{1,x}$, $C_{1,y}$, $C_{2,x}$, $C_{2,y}$, L_x, L_y	2	6	68.28	62.27	61.51	64.02 ± 3.71
R, G, B	1	3	40.52	36.10	34.14	36.92 ± 3.27
R_x, R_y, G_x, G_y, B_x, B_y	5	6	66.49	59.16	61.78	62.48 ± 3.72
$\|\nabla L\|$	1,2,3,5	4	50.11	43.29	45.79	46.39 ± 3.45
$\|\nabla L\|$, L_{xx}, L_{xy}, L_{yy}	1,5	8	76.42	67.92	69.55	71.29 ± 4.51
$\|\nabla C_1\|$, $\|\nabla C_2\|$	1,4	4	37.43	39.95	36.79	38.05 ± 1.67
$\nabla^2 L$	1,2,4,5	4	66.20	59.32	57.77	61.09 ± 4.49
$\nabla^2 L$, L_x, L_y	1,4	6	76.89	67.68	70.58	71.71 ± 4.71
$\nabla^2 L$, $\|\nabla L\|$	1,2,4,5	8	69.98	62.77	60.47	64.40 ± 4.96
$\nabla^2 C_1$, $\nabla^2 C_2$	1,2,4,5	8	52.70	48.91	47.86	49.82 ± 2.54
$det(\nabla\nabla^T L)$	1,3,5	3	58.65	55.07	52.32	55.35 ± 3.17
$det(\nabla\nabla^T C_1)$, $det(\nabla\nabla^T C_2)$	1,2,3,5	8	38.15	36.82	36.12	37.03 ± 1.03

Table 8.4. Results of experiments with descriptors and scales. Each row presents classification rates obtained using one combination of descriptors for the most optimal scales and value of the γ kernel parameter. Additionally, the dimensionality of the resulting histogram is given. The marked rows correspond to the combinations of descriptors resulting in the highest average classification rate. The uncertainties are given as one standard deviation.

Descriptors	Scales	Bins	Class. rate / tr. set [%]			Average class. rate [%]
			Cloudy	Night	Sunny	
L_x, L_y	1,2,4	35	78.35	68.92	73.90	73.72 ± 4.72
L_{xx}, L_{xy}, L_{yy}	1,4	28	81.01	71.76	73.57	75.44 ± 4.90
L_x, L_y, L_{xy}	1,4	27	78.50	69.86	73.23	73.86 ± 4.35
L_x, L_y, L_{xx}, L_{yy}	1,5	15	77.20	69.61	70.40	72.40 ± 4.17
L_x, L_y, L_{xx}, L_{xy}, L_{yy}	2	39	75.70	65.87	70.86	70.81 ± 4.91
$\lvert\nabla L\rvert$, L_{xx}, L_{xy}, L_{yy}	1,5	18	77.84	73.39	71.08	74.10 ± 3.44
$\nabla^2 L$, L_x, L_y	1,4	21	79.04	68.12	72.19	73.12 ± 5.52

Table 8.5. Results of experiments with the number of quantization levels of the histograms. Each row presents classification rates obtained using one combination of descriptors for the most optimal scales, number of bins, and value of the γ kernel parameter. The marked row corresponds to the combination of descriptors resulting in the highest average classification rate. The uncertainties are given as one standard deviation.

based on chromatic cues results in poor performance. This can be explained by the fact that the environment consists of places for which color is not a distinctive feature (two offices, two parts of the corridor). The studies on human scene perception indicate that chromatic cues can facilitate recognition only in cases when color is diagnostic of a scene category. In general, the highest performance was achieved using a derivative-based descriptors applied to the illumination channel. However, it is apparent that histograms based on rotation invariant descriptors result in lower classification rates than those built from responses of single Gaussian derivative filters of the same order (compare Laplacian and second order derivatives as well as gradient magnitude and first order derivatives). This suggests that spatial orientation can be an important cue facilitating place recognition.

Several best performing combinations of descriptors and scales were selected (corresponding to the marked rows in Table 8.4), and further experiments were conducted only for these combinations. The second part of the experiments aimed to obtain the optimal number of histogram quantization levels (bins) per dimension. We built histograms with minimum 5 and maximum 40 quantization levels. Again, the experiments were repeated for 13 values of the γ parameter $(10^{-3}, 10^{-2.5}, \ldots, 10^{3})$, $C = 100$, and three splits into the training and test sets. The results are given in Table 8.5.

It can be observed that the histogram with 28 quantization levels per dimension, computed from normalized second order Gaussian derivatives at scales 1 and 4, performed best. Consequently, further experiments were conducted using only this combination of descriptors.

Training set			Cloudy	Night	Sunny
Parameters			$\gamma = 11.092$ $C = 100$	$\gamma = 16.788$ $C = 100$	$\gamma = 10.839$ $C = 100$
Test set	BO	Cloudy	—	64.82 %	49.07 %
		Night	82.41 %	—	44.44 %
		Sunny	44.44 %	48.15 %	—
	CR	Cloudy	—	94.79 %	95.31 %
		Night	94.53 %	—	87.50 %
		Sunny	95.05 %	89.84 %	—
	EO	Cloudy	—	87.50 %	90.48 %
		Night	91.67 %	—	84.52 %
		Sunny	85.26 %	82.05 %	—
	KT	Cloudy	—	71.76 %	92.13 %
		Night	80.56 %	—	83.80 %
		Sunny	93.52 %	64.82 %	—
	PR	Cloudy	—	62.04 %	67.13 %
		Night	59.72 %	—	43.06 %
		Sunny	83.80 %	56.02 %	—
Avg. classification rate			81.10 %	72.18 %	73.74 %

Table 8.6. Final results of the experiments with indoor place recognition and global descriptors.

8.3.2 Evaluation of the Performance of the System

The final performance evaluation in case of global descriptors was performed in an identical manner to the experiments with local features. The optimal kernel parameters as well as the value of the parameter C were estimated using the hold out cross-validation technique separately for each of the training sets: *cloudy*, *night*, and *sunny*. The histograms were computed from normalized second order Gaussian derivatives at scales 1 and 4 and contained 28 bins per dimension ($28^6 \approx 5 * 10^8$ bins in total). Detailed results together with the parameters used during training are presented in Table 8.2.

It is apparent that the system performs better than in case of local descriptors. In particular, the classification rates obtained for pictures acquired in Barbara's office are considerably higher. The system is thus more reliable. Is important to point out that all the pictures in the database were taken into account in the experiments. As a result, the system was forced to classify even non-informative pictures such as those imaging only blank walls (walls in different places have the same texture). Examples of non-informative pictures in the database are shown in Figure 3.6.

Perc. of init. class. rate [%]		Class. rate [%]	Red. rate [%]	No. of SVs
ORIGINAL		75.67 ± 4.76	—	1000 ± 29
REDUCED	100	75.67 ± 4.76	0.99 ± 0.57	990 ± 31
	99	74.92 ± 4.71	2.40 ± 0.74	976 ± 35
	98	74.16 ± 4.67	4.70 ± 1.09	953 ± 38
	97	73.40 ± 4.62	7.19 ± 0.60	928 ± 32
	96	72.65 ± 4.57	12.66 ± 0.65	873 ± 25
	95	71.89 ± 4.52	17.65 ± 2.44	823 ± 37
	90	68.11 ± 4.29	32.28 ± 3.95	677 ± 53
	85	64.32 ± 4.05	47.48 ± 3.59	526 ± 50
	80	60.54 ± 3.81	59.43 ± 2.35	406 ± 36

Table 8.7. Average results of the experiments with support vector reduction algorithm applied to the classifier trained on the global features extracted from the KTH-INDECS database. The uncertainties are given as one standard deviation.

8.3.3 Experiments with Support Vector Reduction

The standard experimental procedure was used in case of experiments with support vector reduction applied to the classifier trained on global features. Table 8.7 presents the obtained relationship between the reduction rate and the classification rate that was guaranteed to be preserved after the reduction. As in case of local features, the results were averaged over the three training sets.

It can be observed that the reduction rates are smaller than in case of local features. Still, they are similar in comparison to those obtained in Chapter 7. As a result, we can conclude that the amount of reduction is more problem dependent and the influence of the kernel type is smaller.

8.4 Summary

In this chapter we presented the results of experimental evaluation of our visual indoor place recognition system. We compared the performance of the system for both global and local descriptors. Additionally, we tested our support vector reduction algorithm on classifiers trained on both types of features extracted from the pictures of places.

The experiments were conducted on the KTH-INDECS database. The database can be regarded as demanding since the pictures were acquired under changing illumination conditions and capture the natural variability of the environment. Additionally, due to, inter alia, relatively narrow angle of view of the digital camera, it may happen that either the labeling is inconsistent with the contents of a picture or a picture contains little diagnostic information. In spite of that and the fact that training and testing were always performed on pictures acquired under different illumination conditions, the system provides the average classification rate of up

to 81%. Consequently, we achieved high robustness to variations that may occur in real-world environments.

The results indicate that global descriptors (CRFH) perform better for visual place recognition than local features. The performance of the latter might however improve if the affine invariant interest point detector was used. Additionally, the experiments revealed that in case of global descriptors, the most valuable features can be extracted using non-isotropic derivative-based descriptors applied to illumination channel, and that chromatic cues should not be used if they are not diagnostic of a scene category. It may be concluded that the results are consistent with the findings of studies on the human scene perception. On the whole, both Composed Receptive Field Histograms and local features extracted using Harris-Laplace detector and SIFT descriptor show great potential in the domain of place recognition.

This chapter also includes a report of experiments with support vector reduction. We showed that our algorithm may provide a substantial efficiency gain even for complex problems such as visual place recognition. Additionally, the results show that it can be successfully applied to classifiers employing non-Mercer's kernels. In conclusion, the experiments proved the usefulness of the algorithm in two different computer vision applications.

Chapter 9

Summary

Variability of the environment as well as presence of noise generated by changing illumination conditions make visual place recognition an extremely difficult task. In spite of that, humans are able to efficiently perform topological localization on the basis of exclusively visual perceptual information. Consequently, as designers of machines aiming to help people in performing everyday tasks, we need to provide similar capability.

In this thesis we proposed a visual indoor place recognition system built around the Support Vector Machine classifier. We reported the results of an extensive evaluation of the system conducted on the KTH-INDECS database. The database constitutes another contribution of the thesis and was acquired in a way allowing to capture the variability that may occur in a real-world indoor environments. During the design process as well as in the experiments, we placed the strongest emphasis on the robustness and efficiency of the system. Our experiments were conducted on both global and local features extracted from the pictures of places. In both cases the system performed very well, achieving higher classification rates for the global descriptors. The experimental procedure, as well as complexity of the database, ensure that the obtained results vouch for the high robustness of the system. In conclusion, the experiments proved that the excellent generalization abilities of the Support Vector Machines can be exploited also in the domain of place recognition.

As it was previously mentioned, efficiency was another important issue addressed in this thesis. We implemented and thoroughly tested an algorithm allowing for an exact simplification of the support vector solutions exploiting the linear dependence of the support vectors [19]. We showed how the algorithm can be extended in order to provide higher efficiency gain and the ability to trade the performance for the number of support vectors in the final solution. We tested the algorithm on visual data in two domains: material categorization and place recognition. In both cases our method provided substantial reduction in the number of support vectors proving its usefulness in a wide range of computer vision applications.

9.1 Future Work

The work presented in this thesis can be extended in a number of directions:

- The place recognition system can be implemented on a mobile robot platform. The system has been extensively tested under laboratory conditions on the KTH-INDECS database. The successful results suggest that it may be applied to the problem of global topological localization of mobile robots. This creates new challenges but also provides a flexible testing environment for future experiments.

- The ability of the system to generalize its knowledge to novel places should be tested. A system able to categorize places can be extremely useful in many applications. This task may however be difficult due to large differences in appearance between places belonging to the same category.

- A cue-integration scheme can be incorporated into the system. The experiments reported in this thesis were conducted separately for two types of features robust to different types of noise. Moreover, additional cues can be available for the system during recognition e.g. derived from the context or from the type of objects or actions recognized in the scene. For this reason, it becomes important to provide an ability to base the final decision on multiple cues. Such approach is also motivated by the results of studies on human perception showing that the great robustness of our visual system is partly due to the use of several cues.

- We said that the place recognition system may exploit information provided by other recognition systems. On the same basis, information about location can become a valuable cue and serve as a context for e.g. object recognition. Since the database was acquired with sufficient resolution, it is possible to use it for experiments with context-based object recognition.

- Incremental learning techniques can be used in order to update the internal knowledge representation of the system. The ability to learn from experience is particularly important in case of place recognition. This is due to the fact that places evolve over time (or the conditions change unpredictably), and it may be impossible to provide training data that will remain representative in the future.

- The support vector reduction algorithm can be coupled with incremental learning techniques in order to provide means for continuous learning with limited memory resources. Experiments in this field have already been performed (see [63]).

- Several extensions are planned to the KTH-INDECS database. Currently, the database contains pictures taken on one floor of a multi-floor laboratory under

three weather and illumination conditions. It would be of interest to extend the database to other similar floors and acquire more data in the locations used so far. Such an extended data set could be used in experiments with place categorization.

- Number of additional questions arose in designing the experiments: How many training data must be provided in order to achieve robustness to variations in viewpoint and illumination conditions. Is it possible to determine salient viewpoints within the rooms that should be always used for training? Will using affine invariant interest point detectors improve the performance of local features significantly?

All these issues demonstrate that although considerable amount of work has been done, still the majority of questions remain unanswered.

Bibliography

[1] V. Aginsky and M. J. Tarr. How are different properties of a scene encoded in visual memory? *Visual Cognition*, 7, 2000.

[2] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA05)*, Barcelona, Spain, 2005.

[3] M. Auckland, K. Cave, N. Donnelly, and F. Gomes-Pinto. Perceptual errors in object recognition are reduced by the presence of context objects, 2003. `http://people.umass.edu/kcave/context/nomics03.pdf`.

[4] M. Bar and S. Ullman. Spatial context in recognition. *Perception*, 25, 1996.

[5] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In *Proceedings of the International Conference on Image Processing*, pages 513–516, Barcelona, Spain, 2003.

[6] I. Biederman. On the semantics of a glance at a scene. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, pages 213–263. 1981.

[7] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergrouping relational violations. *Cognitive Psychology*, 14, 1982.

[8] P. Blaer and P. Allen. Topological mobile robot localization using fast vision techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA02)*, Washington, DC, USA, 2002.

[9] S. J. Boyce, A. Pollatsek, and K. Rayner. Effect of background information on object identification. *Journal of Experimental Psychology*, 15(3), 1989.

[10] H. Bulthoff and A. Yuille. Bayesian models for seeing shapes and depth. *Comments on Theoretical Biology*, 2(4), 1991.

[11] C. Burges. Simplified support vector decision rules. In *Proceedings of the 13th International Conference on Machine Learning*, pages 71–77, San Mateo, CA, USA, 1996.

[12] C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In *Advances in Neural Information Processing Systems 9 (NIPS)*, pages 375–381, Denver, CO, USA, 1996.

[13] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV05)*, Beijing, China, 2005.

[14] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[15] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5), May 1999.

[16] C. Christou and H. H. Bülthoff. View-direction specificity in scene recognition after active and passive learning. Technical Report 53, Max Planck Institut, 1997.

[17] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[18] Gyuri Dorkó and Cordelia Schmid. Object class recognition using discriminative local features. Accepted under major revisions to IEEE Transactions on Pattern Analysis and Machine Intelligence, updated 13 September, 2005.

[19] T. Downs, K. E. Gates, and A. Masters. Exact simplification of support vector solutions. *Journal of Machine Learning Research*, 2:293–297, 2001.

[20] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2nd edition, 2001.

[21] R. Epstein, K. S. Graham, and P. E. Downing. Viewpoint-specific scene representations in human parahippocampal cortex. *Neuron*, 37(5), 2003.

[22] R. Epstein, A. Harris, D. Stanley, and N. Kanwisher. The parahippocampal place area: Recognition, navigation, or encoding? *Neuron*, 23(1), 1999.

[23] R. Epstein, J. S. Higgins, and S. L. Thompson-Schill. Learning places from views: Variation in scene processing as a function of experience and navigational ability. *Journal of Cognitive Neuroscience*, 17, 2005.

[24] R. Epstein and N. Kanwisher. A cortical representation of the local visual environment. *Nature*, 392(6676), 1998.

[25] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2002.

[26] A. Friedman. Framing pictures: The role of knowledge in automatized encoding and memory for gist. *Journal of Experimental Psychology: General*, 108(3), 1979.

[27] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6), 2000.

[28] G. H. Golub, V. C. Klema, and G. W. Stewart. Rank degeneracy and least squares problems. Technical Report CS-TR-76-559, 1976.

[29] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[30] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[31] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice-Hall, 2nd edition, 2002.

[32] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[33] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 2nd edition, 1999.

[34] J. M. Henderson and F. Ferreira. Scene perception for psycholinguists. In J. M. Henderson and F. Ferreira, editors, *The Interface of Language, Vision, and Action: Eye Movements and the Visual World*, pages 1–58. Psychology Press, 2004.

[35] J. M. Henderson and A. Hollingworth. High-level scene perception. *Annual Review of Psychology*, 50, 1999.

[36] Andrew Hollingworth and John M. Henderson. Does consistent scene context fecilitate object perception? *Journal of Experimental Psychology: General*, 127(4), 1998.

[37] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, Mar 2002.

[38] O. Linde and T. Lindeberg. Object recognition using composed receptive field histograms of higher dimensionality. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR04)*, Cambridge, UK, 2004.

[39] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[40] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[41] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV99)*, Corfu, Greece, 1999.

[42] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[43] P. Mallikarjuna, M. Fritz, A. Tavakoli Targhi, E. Hayman, B. Caputo, and J.-O. Eklundh. The KTH-TIPS and KTH-TIPS2 databass. Available at `http://www.nada.kth.se/cvap/databases/kth-tips`.

[44] M. Mata, J. M. Armingol, A. de la Escalera, and Salichs M. A. A visual landmark recognition system for topological navigation of mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA01)*, Seoul, Korea, 2001.

[45] M. Mata, J. M. Armingol, A. de la Escalera, and Salichs M. A. Using learned visual landmarks for intelligent topological navigation of mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA03)*, Taipei, Taiwan, 2003.

[46] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro. Image-based monte-carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1), 2004.

[47] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. pages 525–531.

[48] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR03)*, Madison, WI, USA, 2003.

[49] K. P. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[50] D. Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3), 1977.

[51] S. K. Nayar. Catadioptric omnidirectional camera. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR97)*, Washington, DC, USA, 1997.

[52] M. E. Nilsback and B. Caputo. Cue integration through discriminative accumulation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR04)*, Washington, DC, USA, 2004.

[53] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, Jan 1996.

[54] A. Oliva and P. G. Schyns. Coarse blobs or fine edges? evidence that information diagnosticity changes the perception of complex visual stimuli. *Cognitive Psychology*, 34, 1997.

[55] A. Oliva and P. G. Schyns. Colored diagnostic blobs mediate scene recognition. *Cognitive Psychology*, 41, 2000.

[56] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3), 2001.

[57] E. Osuna and F. Girosi. Reducing the run-time complexity of support vector machines. In *Proceedings of the 14th International Conference on Pattern Recognition (ICPR04)*, Brisbane, Australia, 1998.

[58] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[59] M. C. Potter. Short-term conceptual memory for pictures. *Journal of Experimental Psychology: Human Learning and Memory*, 2(5), 1976.

[60] A. Pronobis. Detailed results of the evaluation of the support vector reduction algorithm, 2005. Available as an attachment to the thesis or upon request.

[61] A. Pronobis. Detailed results of the experiments with place recognition, 2005. Available as an attachment to the thesis or upon request.

[62] A. Pronobis and B. Caputo. The KTH-INDECS database. Technical Report 297, KTH, CVAP, 2005.

[63] A. Pronobis and B. Caputo. The more you learn, the less you store: Memory-controlled incremental SVM. In *Proceedings of the 9th European Conference on Computer Vision (ECCV06)*, Graz, Austria, 2006. (Submitted).

[64] L. Råde and B. Westergren. *Mathematics Handbook for Science and Engineering*. Studentlitteratur, 5th edition, 2004.

[65] L. W. Renninger and J. Malik. When is scene recognition just texture recognition? *Vision Research*, 44(19), 2004.

[66] R. A. Rensink, J. K. O'Regan, and J. J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8, 1997.

[67] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings of the 4th European Conference on Computer Vision (ECCV96)*, pages 610–619, Cambridge, UK, 1996.

[68] P. G. Schyns and A. Oliva. From blobs to boundary edges: Evidence for time and spatial scale dependent scene recognition. *Psychological Science*, 5, 1994.

[69] S. Se, D. G. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA01)*, Seoul, Korea, 2001.

[70] D. J. Simons and D. T. Levin. Change blindness. *Trends in Cognitive Sciences*, 1(7), 1997.

[71] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. Technical Report 2000-22, University of Amsterdam, ISIS, 2000.

[72] N. Syed, H. Liu, and K. Sung. Incremental learning with support vector machines. In *Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Articial Intelligence (IJCAI99)*, Stockholm, Sweden, 1999.

[73] H. Tamimi and A. Zell. Vision based localization of mobile robots using kernel approaches. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS04)*, Sendai, Japan, 2004.

[74] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2), 2003.

[75] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV03)*, Nice, France, 2003.

[76] A. Torralba and P. Sinha. Recognizing indoor scenes. Technical Report 2001-015, AI Memo, 2001.

[77] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA00)*, San Francisco, CA, USA, 2000.

[78] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1996.

[79] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of the 7th European Conference on Computer Vision (ECCV02)*, Copenhagen, Denmark, 2002.

[80] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2), 2004.

[81] J. Vogel. *Semantic Scene Modeling and Retrieval*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 2004.

[82] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV03)*, Nice, France, 2003.

[83] J. Weston and C. Watkins. Multi-class support vector machines. In *Proceedings of the 7th European Symposium on Artificial Neural Networks*, Bruges, Belgium, 1999.

[84] P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI83)*, pages 1019–1022, Karlsruhe, West Germany, 1983.

[85] R. A. Young. The gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial Vision*, 2:273–293, 1987.

List of Figures

List of Tables