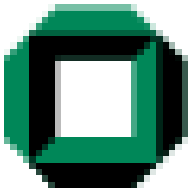


UNIVERSITÄT KARLSRUHE (TH)
FAKULTÄT FÜR INFORMATIK
INSTITUT FÜR ANTHROPOMATIK
Prof. Dr. Rainer Stiefelhagen

cv:hci

DIPLOMA THESIS

Local Feature-based Person Re-Identification in Video

SUBMITTED BY

Martin Bäuml

MAY 2009

ADVISORS

Prof. Dr. Rainer Stiefelhagen
Dipl. Inf. Keni Bernardin
Dr. Jie Yang

Computer Vision for Human-Computer Interaction Research Group
Institute for Anthropomatics
Universität Karlsruhe (TH)
Title: Local Feature-based Person Re-Identification in Video
Author: Martin Bäuml

Martin Bäuml
Hirschstraße 65
76133 Karlsruhe
email: baeuml@gmail.com

Statement of authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the work of others.

Karlsruhe, 8. Mai 2009

. . . . . .

(Martin Bäuml)

Abstract

Person identification in images and image sequences has been an important research topic in recent years. Research has focused mainly on biometric features such as face and gait. Lately, there is an increasing interest in real-world applications of visual person identification. It however comes with high demands on performance and robustness in realistic scenarios. In many situations, approaches that solely rely on biometric features cannot cope with challenges such as low resolution images and non-frontal faces.

In this thesis, the focus lies on using a person's overall appearance for person identification, i.e. his clothes, hair and skin color, accessories such as glasses, hats or bags. The approach relies on the assumption that usually a person does not change his clothes with short time frames. This is a reasonable assumption for many applications such as tracking a person in a distributed surveillance camera network. The property of biometric features to remain practically unchanged over months and years is often not required.

In recent years, local invariant features have been very successfully employed for object recognition and classification. Their success is based on invariance against transformations such as rotation, zoom, illumination and others. Furthermore, the local nature of the features provides robustness in the presence of occlusions and clutter. This thesis investigates how local features can be successfully employed in a challenging person re-identification task.

The mainly employed local features in this thesis are Speeded-up Robust Features (SURF). However, the SURF descriptor is only designed for intensity images. Since color is an important cue for distinguishing persons based on their clothing, I propose two variants of the SURF descriptor to include color information in the descriptor.

One problem of local interest point detectors is that they need structure in an image to repeatably select enough interest points. Since clothing often consists of regions of homogeneous color, the coverage with interest points can be low. A modified watershed segmentation is used to sub-segment an image into visually consistent parts. These parts are then used similar to local features to describe the person.

A person model is built in a bag-of-features manner. In order to reduce the number of features, two clustering approaches are employed. I show that it is important to perform the clustering of features separately for each person. This is in contrast to the usual approaches for object classification where feature from all object classes are clustered to build one common visual codebook. Both temporal decision fusion and fusion of different feature types (here SURF and watershed regions) improve recognition performance.

The evaluation of the approach is twofold. First, the proposed color SURF descriptors are evaluated on Mikolajczyk's data set for performance evaluations of local features. They are shown to outperform the original SURF descriptor especially under view point changes. Second, the proposed approach for person recognition is evaluated on a challenging derivation of

the CAVIAR data set in an open-set recognition task. The local features significantly outperform a baseline approach based on RGB histograms (77% vs. 47% correct classification rate at EER). The SURF color descriptors provide a further performance improvement over the original SURF descriptor, especially when only very little training data is available. Finally, a combination of SURF with watershed regions results in an even better classifier with 80% CCR at EER.

Kurzzusammenfassung

Die Identifikation von Personen in Bildern und Videos ist seit vielen Jahren ein wichtiges Forschungsthema. Der Fokus lag dabei lange auf biometrischen Methoden wie etwa Gesichts- oder Gangerkennung. Mit zunehmender Anwendbarkeit von bild-basierten Personerkennungsverfahren in realistischen Szenarien werden auch neue Anforderungen an die Effizienz und Robustheit dieser Verfahren gestellt. In vielen dieser Situationen sind rein biometrische Ansätze nicht mehr ausreichend, z.B. wenn die Auflösung der Videos gering ist oder Personen ihr Gesicht nur selten zur Kamera wenden.

Diese Arbeit behandelt das Thema der Personenwiedererkennung basierend auf dem kompletten Erscheinungsbild einer Person. Es werden also sowohl Kleidung, Haut- und Haarfarbe als auch Accessoires wie Brille, Mützen und Taschen berücksichtigt. Das vorgeschlagene Verfahren beruht auf der Annahme, dass eine Person über kürzere Zeiträume hinweg ihre Kleidung nicht wechselt und damit ihr wesentliches Erscheinungsbild beibehält. Für viele Anwendungen ist diese Annahme realistisch und ausreichend, etwa für das Verfolgen einer Person in einem verteilten Überwachungskamera-Netzwerk. Die Eigenschaft biometrischer Merkmale, über Tage und selbst Jahre hinweg erhalten zu bleiben, wird meist nicht benötigt.

Lokale invariante Merkmale wurden in den vergangenen Jahren sehr erfolgreich zur Objekterkennung und -klassifizierung eingesetzt. Ihr Erfolg beruht zum einen auf ihrer Invarianz gegenüber Transformationen wie Rotation, Skalierung, Beleuchtungsänderung und anderen. Die Lokalität der Merkmale schafft zum anderen aber auch Robustheit gegenüber Überdeckungen durch andere Gegenstände und macht eine Pixel-genaue Segmentierung der interessanten Regionen unnötig. Die Eignung lokaler invarianter Merkmale für ansichtsbasierte Personenidentifikation wird in dieser Arbeit untersucht.

Als grundlegende lokale invariante Merkmale werden Speeded-up Robust Features (SURF) verwendet. Der SURF-Deskriptor ist allerdings zunächst nur für Grauwertbilder definiert. Da Farbe jedoch eine wichtige Rolle bei der Unterscheidung von Personen spielen kann, werden zwei Varianten des SURF-Deskriptors vorgeschlagen, die auch Information über die Farbverteilung in der Region um einen Interest-Point beinhalten.

Lokale Interest-Point Detektoren benötigen ausreichend Struktur in einem Bild, um genügend Interest-Points für eine robuste Erkennung zu selektieren. Da Kleidung oft größere Regionen gleicher Farbe enthält, kann hier die Abdeckung durch lokale Merkmale gering sein. Um dem entgegen zu wirken, wird eine modifizierte Watershed Segmentierung verwendet, die das komplette Eingabebild in zusammenhängende, visuell konsistente Bereiche segmentiert. Von jedem dieser Segmente werden dann Merkmale zur Beschreibung der Person berechnet.

Die Personenmodelle werden nach einem Bag-of-Features Ansatz aufgebaut. Um die Anzahl der gewonnenen Merkmale zu reduzieren, werden zwei Clustering-Verfahren eingesetzt. Es wird dabei gezeigt, dass es wichtig ist, das Clustern separat für jede Person durchzuführen – im Gegensatz zu Ansätzen in der Objektklassifizierung, wo Merkmale von vielen verschiedenen Objekten zu einem gemeinsamen visuellen Wörterbuch geclustert werden. Sowohl zeitliche Entscheidungs-Fusion als auch Fusion von verschiedenen Merkmalen (hier SURF und Watershed Regionen) führen zu einer Verbesserung der Erkennungsrate.

Die Evaluation der entwickelten Identifikationsmethode ist in zwei Teile gegliedert. Zunächst werden die vorgeschlagenen Farbvarianten des SURF-Deskriptors auf Mikolajczyks Datensatz für die Evaluation von lokalen Merkmalen getestet. Dabei zeigen beide Farbdeskriptoren eine gegenüber dem originalen SURF-Deskriptor verbesserte Performance, insbesondere bei Veränderungen des Blickwinkels. Schließlich wird der Ansatz zur ansichtsbasierten Personenidentifikation auf einem Teil des CAVIAR-Datensatzes in einer Open-Set-Klassifizierungsaufgabe evaluiert. Dabei zeigen die verwendeten lokalen Merkmale ein deutlich besseres Ergebnis als der auf RGB-Histogrammen basierende Baseline-Ansatz (77% gegenüber 47% korrekte Klassifikationen am Punkt der EER). Auch hier sorgen die SURF Farbdeskriptoren für eine Verbesserung, besonders wenn nur wenige Trainingsdaten zur Verfügung stehen. Die Fusion von Watershed Regionen bzw. RGB-Histogrammen mit SURF-Merkmalen liefert schließlich die besten Ergebnisse mit 80% korrekte Klassifikationen am Punkt der EER.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Related Work 3
1.3 Contributions 9
1.4 Thesis Outline 10

2 Local Features **11**
2.1 Interest Point Detection 12
2.2 Feature Description 15
2.3 Object Recognition 16
2.4 SURF - Speeded Up Robust Features 18
2.5 Summary of this Chapter 20

3 Person Recognition Using Local Features **23**
3.1 Local Features 24
3.2 Semi-local Features 29
3.3 Person Recognition 32

4 Experiments **41**
4.1 Color SURF Descriptor Evaluation 41
4.2 Person Recognition Evaluation 47

5 Conclusion **69**
5.1 Future Work 69

Bibliography **73**

1 Introduction

The problem of finding and recognizing people in images and image sequences has been studied in depth for a long time. One can distinguish between the problems of person *detection* and person *recognition*. Person detection refers to the task of deciding if and where a person is present in the image. Person recognition on the other hand determines the identity of the person. In many cases it is not important to actually affiliate the person in the image with a unique identity but only to determine further occurrences of this individual in other images or image sequences. This is also known as the person *re-identification* problem.

1.1 Motivation

In the past, research in visual person identification has focused mainly on biometric features such as face [1] and gait [2]. Biometric features comprise an individual's unique characteristics and are therefore highly discriminative. They remain sufficiently invariant over time, which permits a large time difference between training and recognition. Of course even biometric features underly changes, the appearance of the face for example changes with age. In practice however, long term changes of biometric features are not the main challenges in visual person identification. Especially in surveillance scenarios one has to deal with varying viewpoints, frequent illumination changes and low-resolution images from far-view cameras. In the case of face recognition, additional challenges arise from varying facial expressions and occlusions, e.g. from glasses or hair.

In this thesis the focus lies on using a person's overall appearance for person re-identification. A person's overall appearance is a viable source of information that is very well exploited by humans. Consider the upper row of cropped faces in figure 1.1. Even for a human it is difficult to decide how many different children are shown in these images based on the faces only. If however presented with the full-body appearance, the task becomes a lot easier.

The appearance of a person highly depends on the clothes he is wearing. Although people change clothes regularly and therefore their appearance can vary drastically between sightings, one can assume that a person does not change his clothes completely within time frames of a couple of minutes up to a few hours. This is a reasonable assumption and constitutes a sufficiently long period of stable appearance for many applications such as shopping mall surveillance and track-linking. For increased robustness small local changes should be tolerated, such as the bag of the person in figure 1.2, a baseball cap or sunglasses.

Figure 1.1: The task of deciding how many different children are shown in these images is non-trivial even for humans, if only cropped faces are presented (top row). The full-body appearance however provides more than enough discriminative power (bottom row) (images from Gallagher et al. [3]).

1.1.1 Applications

There are many applications that can benefit from a robust ability to re-identify individuals in images and image sequences. A person's full-body appearance can complement the feature set available to biometric-based systems [4, 5, 3], e.g. by compensating for uncertainties of biometric features when such features are not present or their confidence is low. The other way around, each system that mainly employs features from full-body appearance can be complemented by biometric features. Although I will consider only full-body appearance in this thesis, I strongly suspect that in many real-world scenarios only a combination of both approaches will provide sufficient overall performance. In all of the following real-world applications, current state-of-the-art biometric features alone usually do not suffice because of the aforementioned challenges.

Visual surveillance. Large camera networks are employed to monitor facilities such as airports, subways and shopping malls. A human operator might notice suspicious activity of a person and desires to investigate the whereabouts and actions of this person in the recent past. This can be translated into a content-based information retrieval problem: identify all other image sequences and camera views that contain the same individual as in the example sequence. The sheer amount of available data from a large camera network makes this task infeasible for human operators only and calls for an automated approach.

Semi-automatic multimedia annotation. Television shows, news and sport broadcasts, movies and private photo collections become easily browsable and searchable if they are annotated with sound and fine grained metadata, for example the names of the persons shown in an image. A possible search would be to look for all pictures of one's

Figure 1.2: Small local changes in a person's appearance, here due to a bag, are common and should be tolerated.

grandmother, or news shots containing Barack Obama. Unfortunately, manual annotation can be tedious and is often not worth the effort. However, if the annotator assigns labels to a few pictures of a person, an automated system should be able to label the rest of the collection by re-identifying the individuals from the sample images. Similarly, instead of text based retrieval, the user might also search for images containing the persons or objects from an example image, without explicitly naming them.

Person tracking refers to the task of estimating a person's path in an image sequence. A common problem are disconnected tracks due to occlusions. Person re-identification can help linking tracks of the same individual to bridge occlusions. In scenarios with multiple cameras a similar problem occurs when a person leaves the field of view of one camera and enters the field of view of another camera. The tracks in both cameras should be associated to the same person. This problem is also called the *re-acquisition* problem.

1.2 Related Work

Appearance-based person recognition is still a new field in computer vision. To the best of my knowledge, it was Nakajima in 2003 who proposed the first system explicitly designed to identify people based solely on their full-body appearance. Since then, appearance-based person recognition has gained research interest in mainly two areas. Firstly, in multimedia annotation scenarios, clothing models support face identification systems to increase the confidence of a correct match [4, 3, 6]. Secondly, in surveillance applications, where face identification often fails, approaches rely solely on appearance features [7, 8, 9, 10].

Full-body models of persons have also been employed in tracking systems, such as Pfinder [11]. In Pfinder, a person is represented using a number of 'blobs' modeled as Gaussians with space and color components. This representation is however only used to track the person within

the field-of-view of one camera, not to identify the person. In another tracking application, Krumm et al. [12] used color histograms to disambiguate tracks when they are close together.

In the remainder of this section, I will give an overview of relevant work in the field of full-body appearance based person recognition.

1.2.1 Clothing Segmentation

In monocular vision one can define the appearance of a person as the visible portion of his body which is dominated by his clothes. Among the approaches based on global features, an important aspect is the detection of the person's body and its segmentation from the background. The person's body region can be further subdivided into smaller regions corresponding to (sub-)parts of his clothing. Intuitively, a better segmentation leads to a better model since background, clutter and intra-clothing-variance usually introduce unwanted noise. However, there is a tradeoff between the quality of the segmentation and computational effort. Depending on the application, a crude segmentation might suffice if it at least captures the relevant portions of the clothes and in return is computationally efficient.

Face-relative Regions

A simple approximation of a person's clothing region can be a region relative to some other body part, e.g. the head. A common approach in still images is to employ a face detector first and then estimate the clothing as a rectangular box relative to the position and scale of the detected face [4, 13, 14, 15]. This approach can lead to overlaps between persons if they are too close to each other in the image. To overcome this problem, Song et al. [13] refine the rectangular clothing regions until they do not intersect with other persons' regions.

A face detector also can provide an efficient way to detect persons in videos. Everingham et al. [4] use a frontal face detector on every frame of a video. A Kanade-Lukas-Tomasi tracker is employed to track faces over multiple frames. Similar to the approaches in still images, the clothing region is estimated from a box relative to the face's position. In order to reduce false detections of the face detector, Jaffre et al. [16] employed a temporal smoothing scheme over several adjacent frames.

Video-based Methods

If the input modality is video instead of single images, the whole body of a person can be extracted efficiently under the assumption of fixed cameras and a static background [17, 18, 8, 7, 19, 20].

The main difference between static background subtraction methods is the complexity of the background model. Nakajima et al. [18] compute the average of the k latest images, if no person has been present. Annesley et al. [8] model each background pixel with two Gaussians, one modeling the color in Hue-Saturation space and the other the intensity. The method of Horprasert et al. [21] also detects shadows and highlights so that they do not cause spurious foreground pixels. Their method is adopted by Goldmann et al. [7] with additional

filtering on the resulting foreground maps. Gheissari et al. [9] compute a *frequency image*, denoting for each pixel in a certain time window how often it deviated more than a threshold δ from the current frame's pixel value. Pixels with high deviation count are classified as foreground.

Graph-based Methods

An approach for clothing segmentation in consumer image collections based on graph-cuts is proposed by Gallagher et al. [3]. In the first step, normalized cuts [22] are used to group similar pixels to superpixels. In the second step, another graph cut is computed to segment clothing superpixels from non-clothing superpixels. Additionally, if multiple images of the same person are available, the clothing regions in all images are *co*segmented based on a joint underlying model to improve segmentation accuracy. The authors could show that in a person recognition task a clothing model built from a superior segmentation outperformed models that were extracted from a rectangular box relative to the position of the face.

Gheissari et al. [9] obtain a stable sub-segmentation of the foreground, i.e. the person's appearance, by considering all frames in a time window simultaneously. Their *spatio-temporal segmentation* uses a graph clustering algorithm to group regions of similar appearance. First, an over-segmentation of the foreground is produced using the *watershed algorithm*. Then, neighboring regions in space *and* time are merged to larger clusters as long as the cluster's intra-cluster variance is smaller than the inter-cluster variance.

Segmentation Discussion

Segmentation is a very challenging problem by itself, not only in the context of person recognition. It can probably only be solved sufficiently in conjunction with higher-level reasoning, which presents a chicken-egg-problem in current approaches.

In the case of video, efficient segmentation can be achieved if the background is static and there is a sufficient difference in the appearance of foreground objects against the background. In still images, where we usually do not have a static background model, a face detector is a quite efficient way to find people. Since face detection methods are quite advanced, the presence of a person and its position can be deduced from the output of a face detector. However, face detection only works reliably if a significant portion of the face is visible. It is unsuitable under circumstances where appearance based person recognition is needed most: when the face is *not* visible. Furthermore, the output of a face detector does not provide a segmentation of the person's body. Approximations such as a rectangular region beneath the face have to be employed, which has been shown to be inferior to an actual upper body segmentation [3].

More sophisticated approaches, e.g. based on graph cuts, proved to provide a better segmentation, which in turn leads to better performance of following steps. However, graph-cuts are expensive to calculate and thus not suitable for real-time applications.

Nevertheless, good segmentation is crucial to good performance. Annesley et al. [8] showed that a further segmentation of the clothing region into different parts and fusion of the

Figure 1.3: Left: Simple segmentation in upper and lower half. Right: Sophisticated segmentation of the clothing in its single garments (here manually segmented). Only the better segmentation could improve person recognition performance (image from [8]).

individual retrieval results could improve recognition over using a single full-body descriptor. This is however only true for a good intra-clothing segmentation (in their experiments they used a manual segmentation). Simply splitting the person's body in a top and a bottom half and fusing the two results deteriorated performance.

1.2.2 Modeling Color and Texture

Important cues for describing outer appearance and in particular clothes are undoubtedly color, texture and shape. A large variety of descriptors such as histograms or Gaussian mixture models has been proposed to describe the distribution of these cues on a global level. Usually, the feature distributions are extracted from the full foreground region as classified by the segmentation algorithm.

Several authors evaluated color histograms in different color spaces (e.g. YCbCr [4], RGB [20], HSV [16], LCC [3]) to describe a person's appearance. Commonly employed distance measures between histograms are the χ^2-distance [20, 3] and the Bhattacharyya coefficient [16]. No single color space or distance measure proved to be significantly superior to all others.

Yang et al. [17] compute a smoothed probability density function from a color histogram in tint-saturation space in order to use it as generative model in a multi-modal fusion framework.

Song and Leung [13] use code-word frequency vectors of representative patches to encode clothing pieces. The similarity between two clothes is given by a weighted scalar product. Patches that occur less frequently on a training set are sought to be more discriminative and therefore receive a higher weight. They showed that their method outperforms color histograms.

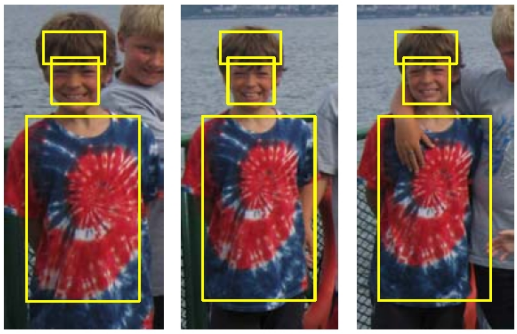


Figure 1.4: Rectangular parts in three images corresponding to hair, face and torso regions. The position of each part is obtained as a maximum a posteriori configuration considering both a data model and a spatial model.

Annesley et al. [8] evaluated MPEG7 color descriptors for visual surveillance retrieval tasks. In their experiments the *Color Structure descriptor* performed best among several other MPEG7 descriptors. Haehnel et al. [23] also compared the performance of a large variety of color and texture descriptors such as *Oriented Gaussian Derivatives*, *Homogeneous Texture Descriptor*, *Edge Histogram Descriptor* and also the MPEG7 Color Structure Descriptor. They found the MPEG7 Color Structure Descriptor in combination with a RBF neural network classifier to be the most stable across all experiments, though not the best in all cases.

In [14] colors are quantized by an adaptive binning technique using the standard k-means algorithm which leads to a bag-of-colors model. Similarly, a k-means quantization is used to cluster texture responses to Gabor filters. A clothing similarity score is learned to account for the different discriminative potentials of different texture clusters.

1.2.3 Local Features

Local features (see chapter 2 for an introduction) have been widely employed for object and face recognition tasks. However, to the best of my knowledge, there are only two papers in which the authors employ local features for appearance-based person recognition.

Gheissari et al. [9] use the Hessian Affine invariant interest operator to nominate interest points. The local region around an interest point is described by a histogram in Hue-Saturation color space. Two interest points between two images are matched if the one is the nearest neighbour under all interest points in the other's image and vice versa. A final validation step is applied to further prune false correspondences.

With focus on real time performance, Hamdoun et al. [10] extract Speeded Up Robust Features ([24], see also section 2.4) from video frames in distances of 0.5 seconds. Features are matched efficiently using the Best-Bin-First approximation in a kd-tree. A simple voting model is employed for recognition.

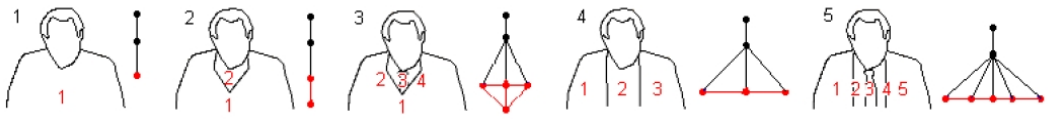


Figure 1.5: Five possible clothing configurations modeled as graphs. The clothing configuration of a person is determined as the best possible match of interconnected image partitions with one of the five models (image from [27]).

1.2.4 Incorporating Spatial Information

A globally computed distribution of cues, such as color or texture, is intuitively not enough to distinguish persons in realistic scenarios. It is also important *where* on the body the colors appear. There are some approaches that address the problem by combining cues with their location.

Yoon et al. [19] model the appearance of a person using a color/path-length profile. The path-length is the length of the shortest path between head and feet of the person's silhouette. The color/path-length distribution is estimated using Gaussian mixtures. The authors showed that the incorporation of the spatial information into the model improved the discrimination between persons.

Sivic et al. [6] build on the general framework of pictorial structures [25]. A person is modeled with three rectangular regions corresponding to hair, face and torso regions (see figure 1.4). Spring-like potentials between model parts assure that they retain a reasonable configuration. The appearance of each part is modeled as Gaussian mixture in RGB color space. A maximum a posteriori configuration of the model parts is computed to find known people in images even when a face detector did not find their face.

Wang et al. [26] incorporate co-occurrence information of appearance features relative to object parts, in this case human body parts. The co-occurrence matrices are computed based on a generalization of integral images which makes their computation fast. The incorporation of the spatial information of the features lead to a significant improvement in recognition performance.

Gheissari et al. [9] fit a decomposable triangulated graph to salient edges in order to crudely estimate the body pose. The graph is divided into several regions, i.e head, upper torso, lower torso and legs, from each of which a separate description is computed. For the matching of persons, only the descriptors of corresponding regions are compared. This approach is shown to outperform a combination of Hessian Affine region detector and a histogram descriptor for the local regions. However, the employed local descriptor is not state-of-the-art. Furthermore, the authors did not publish any information about the runtime of the graph-based approach.

1.2.5 Semantic Description

Borras et al. [27] propose a method to automatically derive a high-level interpretation of a person's clothes. An input image is segmented into different regions of homogeneous color

or texture. The spatial configuration of the segments is modeled as an attributed graph, which is matched against five pre-defined graph structures (see figure 1.5). Each structure represents a common configuration of clothes (e.g. tie and jacket).

Zhang et al. [20] determine the type of a person's upper body garment, depending on the presence of a collar (C) and/or long sleeves (S). The four possible garments are business shirt (C+S), polo-shirt (C), sweater (S) and t-shirt (neither). The presence of the collar is determined from the number of Harris corner points within a region beneath the face. The sleeve length is deduced from the number of skin-colored pixels outside the face.

1.2.6 Summary of Related Work

Before any recognition can take place, a person of interest has to be segmented from the background. In current work, there are three approaches: Firstly, static background subtraction in videos. Secondly, employing a face detector first and then defining the body of a person relative to the position and scale of the face. And thirdly, circumventing the need for segmentation by employing local features extracted at interest points. However, also further sub-segmentation of a person's full-body appearance into smaller parts is of interest. A good sub-segmentation into clothing parts can improve recognition performance [8] or be ground for higher level description of the clothes [27].

Much of the previous research is concerned with the question of which is the right descriptor for describing clothes. Many color and texture descriptors have been tested. In more than one experiment the MPEG7 Color Structure Descriptor performed very well, supporting the intuition that both color and texture, and their combination, play an important role in describing clothing. Shape however has not yet been investigated as a viable source of information. Local features, playing an important role in recent object recognition approaches, have also not gained much research attention. Two approaches however successfully employed different kinds of local features and reported promising results. This motivates this thesis.

Ultimately, one will also be interested in high level descriptions of a person's clothes. The interesting approach of Borras et al. [27] classifies an input image as one from five different upper body clothing configurations. A bottom-up approach has been proposed in [20], using clothing part detectors such as a collar detector and then reasoning about the type of garment based on the presence of different parts.

1.3 Contributions

While most of the previous research in non-biometric person recognition mainly concentrated on global features, the focus of this thesis lies on local features. The SURF interest point detector and descriptor [24] are employed in a realistic surveillance scenario, where one has to deal with low resolution images, illumination changes and articulated body movements. This part of this thesis is inspired by the work of Hamdoun et al. [10], who used a simple voting approach in combination with SURF. In contrast to Hamdoun, for each person one separate model is built. The distance between two features is used as confidence measure for

their matching. In addition, I propose two variation of the SURF descriptor to also encode color information instead of just intensity patterns. These color SURF descriptors are shown to provide a performance improvement in both the person recognition task as well as in finding one-to-one interest point correspondences between two images.

Since clothes often contain large regions of homogeneous color, usual interest point detectors can fail to find enough interest points within these regions. In order to use such regions, an image is sub-segmented in visually alike regions. Similar to the approach used for local features, descriptors of these semi-local regions are used to train one model for each person.

A clustering approach is used to select the most salient features. Instead of using features in a nearest neighbour manner, cluster centers are used to represent strong features learned from multiple frames. It is shown, that using such clustered features, similar or even better performance can be achieved than with nearest neighbour matching with all training features. Furthermore, the recognition stage based on clustered features is faster.

1.4 Thesis Outline

The remainder of this thesis is organized as follows.

I begin by introducing local features in chapter 2. The general concept of interest point detectors and descriptors is explained and illustrated. Two approaches to object recognition using local features are briefly outlined. The chapter concludes with an in-depth explanation of Speeded-up Robust Features (SURF) in section 2.4, which are the main type of local features used in this thesis.

In chapter 3 the person recognition approach is explained. I propose two variations of the SURF descriptor to encode color information in the descriptor (section 3.1.3). In the following section 3.2, semi-local region features based on the watershed segmentation are explained. Finally, I introduce the training and classification approach for person recognition (section 3.3) and depict how temporal decision fusion and fusion of feature type can increase recognition performance (sections 3.3.3 and 3.3.4).

An extensive evaluation can be found in chapter 4. In section 4.1, the two proposed color SURF descriptors are characterized on a Mikolajczyk's evaluation data set for local features. In the remainder of chapter 4, the person recognition approach is evaluated. In section 4.2.1 I describe a challenging data set for person recognition which I derived from the CAVIAR data set. The evaluation criteria are explained in detail in section 4.2.2. Finally, in section 4.2.3, the results of the evaluation are presented.

2 Local Features

In the last few years, local features have regained a lot of popularity because of their successful application in a broad variety of computer vision problems such as wide baseline stereo [28], object class recognition [29] and specific object instance recognition [30]. Basically in all tasks which involve finding correspondences between two or more images, local features provide a valuable tool.

The basic principle for finding correspondences between images by using local features goes as follows. First, so-called *keypoints* or *interest points* are chosen at prominent points in the image in such a way that they can be also found in other images with high reliability, even under different viewing conditions. One can broadly distinguish between corner, blob and region detectors. This does however not imply that these detectors actually find semantically relevant object parts. Their underlying localization criterion is usually based on more abstract notions such as maxima of the Hessian Matrix. A common goal of all interest point detectors is a high repeatability, in the best case allowing to find the same interest points under illumination changes, rotations and even arbitrary affine transformations.

In the second step, a local region around the found interest point is extracted as the feature *descriptor*. Locality and distinctiveness of the descriptor have to be traded off carefully, as increasing the one decreases the other. Many local descriptors have been proposed in the literature, e.g. [24, 30, 31].

Finally, the descriptors are matched against descriptors from another image or a descriptor database, depending on the application. For comparing two individual descriptors, commonly used distance measures are the Euclidian or Mahalanobis distance. A larger descriptor requires more computational effort, which is again traded off against its distinctiveness. Efficient data structures for nearest neighbour matching such as kd-trees [32] help keeping the matching computationally tractable.

In the following sections we discuss these steps in more detail. First, interest point detection is generally introduced and two popular interest point detectors are explained (section 2.1). A brief overview over feature descriptors is given in section 2.2. The principal approach to object recognition and classification using local features is explained in section 2.3. For on-line applications all three steps should be fast. In section 2.4 *SURF* [24] - Speeded Up Robust Features - are introduced which were especially designed with low computational complexity in mind. The chapter is concluded with a short summary (section 2.5).

2.1 Interest Point Detection

The term *interest point* has been introduced by Moravec in 1979 [33]. It can be defined as a well localized anchor point. An ideal interest point has several desirable properties such as repeatability, accurate localization, distinctiveness, efficient computation and availability in large enough quantities across many image contents. For the purposes of person recognition a reliable and accurate re-detection in other images is probably the most important property.

Interest point *detection* refers to the task of finding such well localized points in an image. Although desirable, in general it is today intractable to extract feature points with a higher semantic meaning such as 'wheel', 'leg' or 'head' without losing the detector's general applicability to a wide variety of domains. Rather, current approaches are directly based on the intensity patterns of the image, and select interesting structures such as points with high curvature (corners, T-junctions), blob-like structures or small homogeneous image regions. One possible distinction between the different approaches is the kind of detected underlying structure.

In the following, we introduce two popular interest point detectors in more detail, the *Harris Corner Detector* and the *Hessian Detector*. An approximation of the Hessian Detector is later employed in the SURF detector in section 2.4. This section is concluded by a brief discussion of automatic scale selection for *scale invariant* interest point detectors.

2.1.1 Harris Corner Detector

A very popular approach is the *Harris Corner Detector* [34]. It detects image structures that contain a high curvature such as corners and T-junctions. It is based on the second moment matrix M,

$$M(\sigma_D, \sigma_I) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_y I_x(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \tag{2.1}$$

describing the gradient distribution around an image point, where σ_D is the *derivative scale* and σ_I the *integration scale*, i.e. the local image derivatives I_x and I_y are calculated using a Gaussian kernel of scale σ_D and then convoluted with a Gaussian kernel of scale σ_I. An interest point is detected, if both eigenvalues of M are large, corresponding to a high *cornerness* at that point. Harris proposed the following cornerness measure c to capture simultaneously large eigenvectors

$$c = det(M) - \lambda \cdot trace(M) \tag{2.2}$$

Without going too much into detail, note that the Harris detector is invariant with respect to translation and rotation but not to scale changes.

2.1.2 Hessian Matrix Based Detectors

Hessian detectors respond well to blob-like structures, i.e. round or ellipse-shaped intensity patterns. In a local region around image point $\boldsymbol{x} = (x, y)$, the local Taylor expansion

$$I(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx I(\boldsymbol{x}) + \nabla I(\boldsymbol{x})\Delta\boldsymbol{x} + \Delta\boldsymbol{x}^T \mathcal{H}(\boldsymbol{x})\Delta\boldsymbol{x} \tag{2.3}$$

approximates the intensity function up to the second-order term. The *Hessian matrix* $\mathcal{H}$ is the square matrix of second-order partial derivatives. Its eigenvalues capture the strength of the quadratic term. In first approximation, blobs are two-dimensional quadratic functions. Large eigenvalues of the Hessian matrix therefore indicate the presence of a blob.

For a point $\boldsymbol{x} = (x, y)$ in image I, the Hessian matrix at scale σ_D is defined as

$$\mathcal{H}(\boldsymbol{x}; \sigma_D) = \begin{bmatrix} L_{xx}(\boldsymbol{x}; \sigma_D) & L_{xy}(\boldsymbol{x}; \sigma_D) \\ L_{yx}(\boldsymbol{x}; \sigma_D) & L_{yy}(\boldsymbol{x}; \sigma_D) \end{bmatrix} \tag{2.4}$$

where L is the convolution of image I with a Gaussian kernel $g(\sigma_D)$ of scale σ_D. $L = L(\boldsymbol{x}; \sigma_D)$ is also called *scale-space representation* of I (see also section 2.1.3). Based on the Hessian matrix one can build two flavors of blob detectors. Both select interest points for which the eigenvalues of the Hessian matrix are large. One is based on the trace of $\mathcal{H}$, the other one on the determinant of $\mathcal{H}$. Note that in neither approach the eigenvalues have to be explicitly computed.

Laplacian of Gaussian

The trace of the Hessian matrix is equal to the *Laplacian of Gaussian*

$$\nabla^2 g(\sigma_D) * I = \nabla^2 L = L_{xx} + Lyy = trace(\mathcal{H}) \quad . \tag{2.5}$$

Recall that the trace of $\mathcal{H}$ is the sum of the eigenvalues of $\mathcal{H}$. Thus, a large trace corresponds to at least one large eigenvalue of $\mathcal{H}$ and we can select interest points at local minima and maxima of the trace according to

$$\begin{aligned} \boldsymbol{x}^* &= \text{argmax}_{\text{local}} \left| trace\left(\mathcal{H}(\boldsymbol{x})\right) \right| \\ &= \text{argmax}_{\text{local}} \left| L_{xx}(\boldsymbol{x}) + L_{yy}(\boldsymbol{x}) \right| \end{aligned} \tag{2.6}$$

Selecting interest points based on the trace of the Hessian matrix has one major drawback. The trace can be large if only one of the eigenvalues of $\mathcal{H}$ is large. This is equivalent to a long-shaped blob with most of its extension in one direction. The Laplacian therefore also responds to line segments and might as such deliver many poorly localized interest points.

Hessian Determinant

Instead of using the trace of the Hessian matrix, one can just as well use its determinant for selecting interest points. The determinant of the 2×2 Hessian matrix can be computed as

$$det(\mathcal{H}) = L_{xx} L_{yy} - L_{xy}^2 \tag{2.7}$$

which is equivalent to the product of the two eigenvalues of $\mathcal{H}$. Both the presence of a light and a dark blob are indicated by large positive values of the determinant, whereas a large negative determinant arises from saddle points. Since we do not really look for a semantic

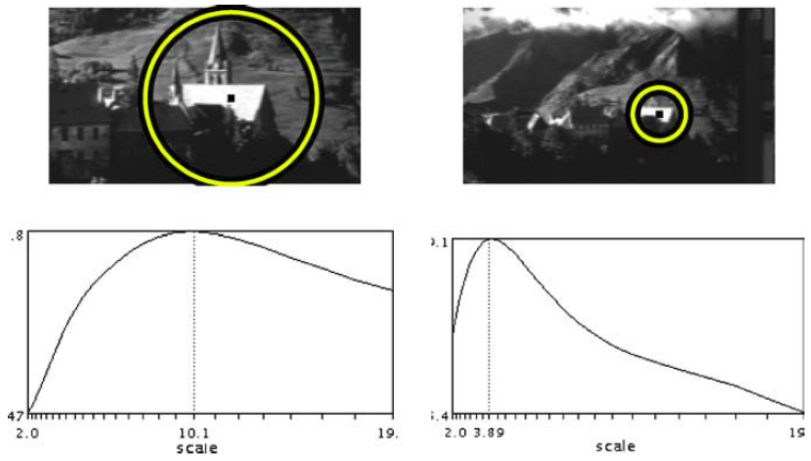


Figure 2.1: Scale selection. The maximum of an appropriate scale-dependent function is chosen as characteristic scale s. By sampling the feature's descriptor from a region of size relative to s around the interest point, the feature can be matched at arbitrary scales (image from [35]).

association with the interest points, we do not reject the saddle points and select the interest points as

$$\begin{aligned} \boldsymbol{x}^* &= \text{argmax}_{\text{local}} \left| det\left(\mathcal{H}(\boldsymbol{x})\right) \right| \\ &= \text{argmax}_{\text{local}} \left| L_{xx}(\boldsymbol{x}) L_{yy}(\boldsymbol{x}) - \left(L_{xy}(\boldsymbol{x})\right)^2 \right| \end{aligned} \tag{2.8}$$

This detector suppresses responses from long-shaped blobs and thus avoids respective problems in contrast to the Laplacian.

2.1.3 Automatic Scale Selection

In practice we expect the same image structures to appear in different sizes in different images. However, basic interest point detectors only work well for features of a certain scale. One possible solution to overcome this problem is to apply the detectors successively at different scales. This however produces many responses at the same spatial location albeit at different scales.

Automatic scale selection deals with this problem by finding a characteristic scale for each interest point independent of the size of the input image. This is usually accomplished by finding local minima or maxima of some function over different scales (cf. figure 2.1). An interest point detector is called *scale invariant* if it not only finds a distinct interest point in space but also selects its inherent scale.

Both the Laplacian and the Hessian have been employed for automatic scale selection [36], for example for scale invariant extensions to both Harris- and Hessian-based interest point detectors [37]. The approximation of the Laplacian

$$\begin{aligned} \nabla^2 L(\sigma_D) &= L_{xx}(\sigma_D) + L_{yy}(\sigma_D) \\ &\approx \frac{1}{2\Delta\sigma} \left(L(\sigma_D + \Delta\sigma) - L(\sigma_D - \Delta\sigma) \right) \end{aligned} \tag{2.9}$$

is used for an efficient implementation for finding scale-space extrema in [30]. It avoids computing the second order derivatives in both directions. This approximation is often referred to as *Laplacian of Gaussian*.

2.2 Feature Description

A *feature descriptor* captures the distinct intensity patterns around an interest point. The goal is to match the descriptor against descriptors from other images and thus establish correspondences between points from different images.

There are several properties that are of interest for a good descriptor. Obviously, a descriptor should be as distinctive as possible to avoid false correspondences. Yet, it is also desirable that the descriptor is tolerant with respect to variations in illumination, rotation, scale and viewpoint. Finally, the computation of the descriptor should be efficient, as a single image can easily contain 1000 or more interest points.

In its simplest form, a feature descriptor is formed by sampling intensity values around the interest point. However, such *patches* usually have an unnecessarily high dimensionality and are also easily susceptible to small intensity variations. Many feature descriptors have been proposed that are superior to patches in both dimensionality and tolerance against variations, e.g. invariant moments [31], geometric blur [38], shape contexts [39] and many more. Inspired from insights in biological vision, Lowe proposed to use histograms of oriented gradients [30] as descriptor which achieves to a great deal robustness against small intensity variations. His descriptor, coined *SIFT* for Scale Invariant Feature Transform, was shown to outperform others [40].

In general, tolerance against a transformation can be achieved by normalizing the descriptor with respect to the transformation. After such a normalization we can say that the descriptor is *invariant* under the transformation. For example, let us rotate the feature region around an interest point always in such a way, that is aligned with the image's *content* in always the same way. Then, the extracted descriptor will always be the same, regardless of the original rotation of the input image. We say, the descriptor is *rotation invariant*.

For rotational invariance of the SIFT descriptor, gradient directions are sampled and the most frequent direction is selected as feature direction. The descriptor is then normalized with respect to this feature direction. Similar approaches have been proposed to achieve invariance against arbitrary affine transformations [37, 35].

In section 2.1.3 we already addressed the problem of finding a characteristic scale for the interest point. This information can be directly employed to compute a scale invariant

descriptor. A region around the interest point with size relative to the scale of the interest point is rescaled to a fixed patch size. The feature descriptor is then always extracted from this rescaled patch which in theory makes the descriptor's content independent of the scale of the original image.

2.3 Object Recognition

In the tasks of object class and object instance recognition, local features build the basis for higher level reasoning. Local features are particularly appealing in that they basically eliminate the need for semantic-level segmentation, which is up to now an intractable problem in computer vision. Instead of a huge quantity of pixels, local features provide only a few (compared to the number of pixels) distinct yet robust characteristics of an image. Some are representing (interesting) foreground objects and some background. Compared to pixel-by-pixel segmentation of foreground objects from background, the filtering of the respective local features is a by far easier task because of their increased distinctiveness over single pixels.

Object recognition based on local features builds on the insight from Schmid and Mohr [41] that it is not necessary to obtain a complete correspondence between all extracted features and the object model (e.g. local features extracted from training images). It is rather sufficient to find enough correspondences to (i) discriminate the object from background, and to (ii) distinguish between different objects. Due to the local nature of the features, a significant portion of an object can be occluded and still be correctly recognized.

Both object class and object instance recognition can be viewed as feature matching problems. The recognition procedure can be as simple as counting correspondences and selecting the object class with the most correspondences. In the following I will briefly introduce two approaches for object recognition using local features.

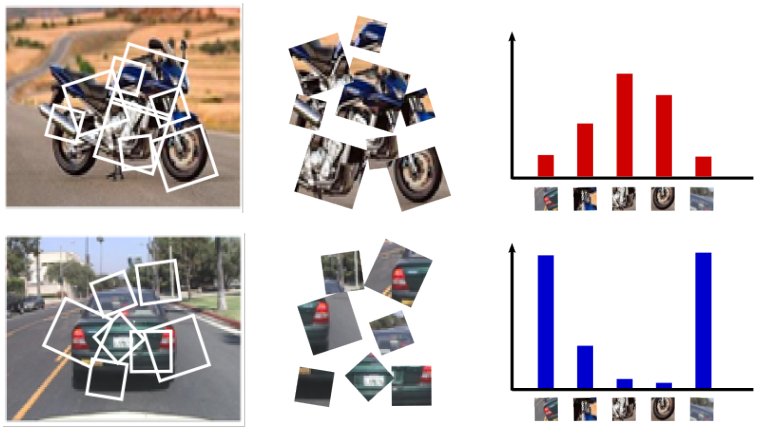
2.3.1 Voting Model

The basic idea behind the *voting model* is that similar objects have more correspondences between local features than dissimilar objects. Consider the task of telling cars, airplanes and bicycles apart. Given a bunch of training images, we extract local features from them and store their descriptors in a database together with their respective object class. The database thus consists of many descriptor-object pairs.

For a new picture, for which the task is to decide which of the possible objects is shown, we again extract local feature descriptors. For each of the descriptors we find its nearest neighbour in the database (i.e. the closest descriptor according to some distance metric, e.g. L2) and cast a vote for the respective object class. Finally, the object class with the most votes is selected.

The voting model has the disadvantage that it does not deal well with similar features across different objects (e.g. a corner). Due to the hard voting scheme, the vote would only be cast for exactly one model which might cause a lot of incorrect votes.

(a)



(b)

Figure 2.2: Object class recognition using histograms of visual words. (a) Clustering local features in order to build a visual codebook. (b) Computing occurrence histograms of visual words as object models.

2.3.2 Histograms of Visual Words

Instead of using local features directly, one can also find common patterns among the features first, grouping those that are alike. Using clustering techniques such as agglomerative clustering, one can build a *visual codebook*. The codebook consists of *visual words*, common patterns that can be used to describe objects (see figure 2.2a).

During training of an object class, the extracted local features are matched against the codebook. It is recorded, which of the visual words are matched how often, leading to an occurrence histogram. These histograms are saved as object models (see figure 2.2b).

For recognition, an occurrence histogram for the test image is computed as in the training stage. This histogram is now compared to all trained object histograms, e.g. using the χ^2-distance. The closest matching histogram denotes the found object class.

This model is advantageous over the simple voting approach since visual patterns that occur in more than one object class can contribute to the recognition of all of them. On the other hand, building a visual codebook can be computationally expensive and needs careful adjustment of the clustering parameters since usually the underlying distribution of visual patterns is unknown.

2.4 SURF - Speeded Up Robust Features

Speeded Up Robust Features, proposed by Bay et al. [24], is a combination of a scale-invariant interest point detector and a feature descriptor. Both detector and descriptor have been especially designed with computational efficiency in mind. This is achieved by the extensive use of box-type filters which can be computed fast at arbitrary scales using integral images. SURF works on single channel images only, no color information is exploited.

2.4.1 Integral Images

An *integral image* is a data structure for efficiently computing box-type filters at arbitrary scales. In an integral image $I_{\sum}$ the value at point $\boldsymbol{x} = (x, y)$ is the sum of all pixel values from the rectangular region between $\boldsymbol{x}$ and the origin in an input image I.

$$I_{\sum}(\boldsymbol{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{2.10}$$

The integral image can be computed efficiently in a single pass and subsequently allows to compute the sum of pixel values over arbitrary rectangular regions of the input image I in constant time according to the following equation,

$$\begin{aligned} \sum_{i=x_1}^{i \leq x_2} \sum_{j=y_1}^{j \leq y_2} I(i, j) \; = \; & I_{\sum}(x_2, y_2) + I_{\sum}(x_1 - 1, y_1 - 1) \\ & - I_{\sum}(x_1 - 1, y_2) - I_{\sum}(x_2, y_1 - 1) \quad . \end{aligned} \tag{2.11}$$

2.4.2 Interest Points

Unlike other scale-invariant interest point detectors, SURF uses the determinant of the Hessian matrix for both interest point and scale selection. The Hessian matrix is approximated using box-type filters which can be computed very fast using the integral image. For the smallest scale $\sigma = 1.2$ the filters are of size 9×9. Both the discretized original filters and their approximations can be seen in figure 2.3.

If we denote the filter responses with D_{xx}, D_{yy} and D_{xy} we can approximate the Hessian determinant as (cf. equation 2.7)

$$det\left(\mathcal{H}_{\text{approx}}\right) = D_{xx} D_{yy} - (\omega D_{xy})^2 \quad , \tag{2.12}$$

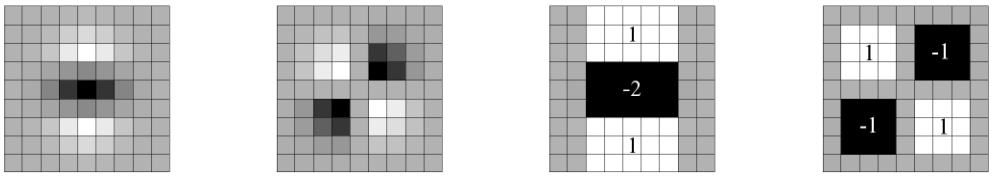


Figure 2.3: Discretized and approximated filters for calculating the determinant of the Hessian matrix. The two filters on the left are second-order partial derivatives of a Gaussian filter at scale $\sigma = 1.2$ in y- and xy-direction. Their approximated counterparts are depicted on the right. (image from [24])

where ω adjusts the relative weight of the filters, which would otherwise be shifted due to the approximation. In practice a constant value of $\omega \approx 0.9$ can be used for filters at all scales [24].

In contrast to the usual approach, scale space analysis is performed by up-scaling the filters instead of down-scaling the image. The filter responses are normalized by the size of the filter. This is where the approximation of the filters and the usage of integral images pays off. No Gaussian scale space pyramid has to be explicitly computed.

The Hessian determinant is recorded over image and scale space. Interest points are selected by employing $3 \times 3 \times 3$ non-maximum suppression, i.e. a point is only selected as interest point if its Hessian determinant is larger than any of its neighbours' in both scale and image space. Furthermore, the *strength* of the interest point $h = \left|det\left(\mathcal{H}_{\text{approx}}\right)\right|$ must lie above a threshold t.

The location of each local maximum is further refined by interpolation in both scale and image space. The interpolation is achieved by fitting a quadratic function to the intensity values [42] and then shifting the maximum's location according to the gradient of the fitted function (effectively performing one step of gradient descent). Especially the scale space interpolation is important because of large differences between two scale steps.

2.4.3 Descriptor

The SURF descriptor captures the information about the intensity patterns in the region around the interest point. It is built from local intensity differences. These are calculated as responses of first order Haar-Wavelets which again can be sped up using the integral image.

Orientation Assignment

The extraction of the SURF descriptor consists of two steps. First, a characteristic orientation of the descriptor is estimated in order to achieve invariance towards image rotation. For

that matter, Haar wavelet responses around the interest point are calculated within a circular neighbourhood with size relative to the scale of the interest point. Again, the integral image can be used to speed up the computation of the Haar wavelet responses.

The filter's responses are interpreted as x- and y-coordinates of vectors originating at the interest point. Within a sliding orientation window all vectors are summed up. The maximum resulting vector over all sliding windows determines the characteristic orientation of the descriptor.

Descriptor Extraction

In the second step, the actual descriptor extraction, we consider a square region around the interest point oriented according to the descriptor assigned orientation. The size of the region is chosen as $20s$, where s is the characteristic scale of the interest point. The region is further divided into 16 sub-regions by imposing a 4×4 grid. From each of the subregions, four descriptor entries are calculated, resulting in total in a 64-dimensional feature descriptor.

The four descriptor entries of each sub-region are calculated as follows. Within the subregion, Haar-wavelet responses in x- and y-direction are calculated at 5×5 equally spaced points within the sub-region. To increase the contribution of responses closer to the interest point, the responses are weighted with a Gaussian window of scale $\sigma = 3.3$ originating at the interest point. The weighted responses are summed up to form two of the descriptor entries. The other two entries are formed from the sum of the *absolute values* of the responses. If d_x denotes the weighted responses in x-direction and d_y the weighted responses in y-direction, the four descriptor entries for the subregion then are

$$v_1 = \sum d_x \qquad v_3 = \sum |d_x|$$
$$v_2 = \sum d_y \qquad v_4 = \sum |d_y| \quad .$$

Illumination and Contrast Invariance

We assume a simple model of image intensity variations under the influence of different illumination or camera calibrations. The model consists of an offset and a linear scale factor for the intensity values.

$$I = I_{\text{offset}} + s \cdot I \tag{2.13}$$

Illumination invariance of the descriptor is therefore already achieved, since only differences of intensity values are considered in the descriptor. The offset parameter I_{offset} becomes irrelevant. The contrast, or scale factor s of the intensity values, remains a common factor in all descriptor entries. It is removed by normalizing the descriptor to unit length.

2.5 Summary of this Chapter

In this chapter the concept of local invariant features was introduced.

First, I explained two popular methods for selecting interest points, the Harris Corner Detector and the Hessian Blob Detector. By considering not only local extrema in image space, but also over multiple scales, a characteristic scale of each interest point can be computed. I presented two methods to do this scale space analysis, one based on the trace of the Hessian matrix and another on its determinant.

The local region around the interest point can be described in a local feature descriptor. Invariance against various transformations can be achieved by normalizing the descriptor with regard to the transformation.

For this thesis, local features are employed in a similar manner as for object recognition. For that reason, I gave a short introduction on object recognition based on local features. Due to their trade-off between locality and distinctiveness, local features are a robust representation of an image's content. They eliminate to a great extend the need for good foreground-background segmentation. Furthermore, they perform well even in the presence of clutter and occlusions.

Finally, I introduced SURF, which constitutes both an interest point detector and a feature descriptor. SURF is computationally efficient due to the extensive usage of integral images and fast box-type filters. SURF performs on-par with current state-of-the-art local feature interest point detectors and descriptors.

3 Person Recognition Using Local Features

In this chapter, the main contributions of this thesis towards non-biometric person identification are explained. The focus lies on using local and semi-local features and how they can be efficiently exploited to be useful in a challenging scenario.

The basic recognition procedure employed in this thesis can be divided into three steps.

- Features are extracted from image sequences. An image sequence consists of a number of frames showing one person. Such a sequence can be for example the output of a generic person tracker. The result of the feature extraction stage is a set of features representing the frames' content.

- The second step is a supervised training step where models are trained for all individuals that are to be re-identified in other image sequences. Features from one or more image sequences (extracted in the first step) are employed as training data. These models can then be used to identify the trained persons in other image sequences.

- In the third step, the recognition or classification, a person in a new image sequence is identified using the models from step 2. A score is computed for each of the trained models, indicating how well the person in the input sequence matches the model. In the simplest version, the classifier chooses the person which achieves the highest score over the whole sequence.

In the remainder of this chapter all three steps are described in detail. In sections 3.1 and 3.2, the features that are used for training the models are introduced and explained. Section 3.1 begins with a recap of SURF's most important properties. Then, two variations of SURF's descriptor for color images are motivated and proposed in section 3.1.3. Semi-local features based on the watershed segmentation are explained in section 3.2. In section 3.3 the classification procedure is described. Clustering is introduced as a means to both speed up the classification and increase the recognition performance.

The classifier itself operates on single images. However, it can be leveraged that there are multiple consecutive frames in an input sequence that all show the same person. Temporal filtering allows to compensate for wrong decisions of the classifier in some frames of the sequence by considering the whole sequence (section 3.3.3). Finally, different features are combined to complement each other.

3.1 Local Features

In unconstrained environments where persons freely move around, two major challenges are (i) non-rigid deformations of the human body and (ii) view point changes. Varying view points can be for example a result from the different positions of multiple cameras, but also from small movements of a person's body, e.g. during walking.

An overview and introduction to local features was given in chapter 2. In this work, local features are used for person recognition. They have been shown to be able to deal with various challenges such as view point and illumination changes in many object recognition tasks, due to their (partial) invariance against certain types of transformations (depending on the exact interest point detector and descriptor). The local nature of these features reduces the influences of non-rigid deformations of the human body.

In this work, the employed local features are *Speeded-up Robust Features* or *SURF* [24]. SURF consist of both an interest point detector and a descriptor. The interest point detector is a fast approximation of a Hessian blob detector. The interest point descriptor extracts intensity gradient patterns around the interest point. I selected SURF for this work for two reasons. First, SURF performs on par with state-of-the art interest point detectors and descriptors [24, 40]. Second, SURF has been designed with efficiency in mind, resulting in both a fast detector and descriptor. This is especially important for real time applications. Although this work does not explicitly address run time performance, the applicability of this approach within a real time system has always been kept in mind.

In the following, the most important parts of SURF will first be repeated briefly. For a detailed description the reader is referred to section 2.4. After the introduction of SURF, two variations of the SURF descriptor are proposed to include color information, which makes the descriptor more discriminative in color images.

3.1.1 Fast-Hessian Interest Points

In a first step, for each image in an input sequence interest points are selected. This selection is based on an approximate version of the Hessian matrix. The approximation is calculated from integral images (see section 2.4.1). The determinant of the approximated Hessian matrix

$$det\left(\mathcal{H}_{\text{approx}}\right) = D_{xx}D_{yy} - (\omega D_{xy})^2 \quad , \tag{3.1}$$

is used for both interest point and characteristic scale selection. An interest point is selected if it is a local maximum in both space and scale and its strength $h = \left|det\left(\mathcal{H}_{\text{approx}}\right)\right|$ lies above a threshold (which is a parameter to the algorithm). The resulting interest point is refined further to sub-pixel and sub-scale accuracy by interpolation.

3.1.2 SURF Descriptor

The SURF descriptor describes a local region around the interest point. Responses of first order Haar-Wavelets are computed using the same integral image as in the interest point detection step.

Orientation Assignment. In order to achieve invariance against image rotation, a dominant orientation at the interest point is calculated. Within a circular region around the interest point, responses to first-order Haar-Wavelets in x- and y-direction are interpreted as abscissa and ordinate of vectors originating at the interest point center. The dominant direction is computed as the sum of these vectors within a sliding orientation window. The maximum resulting vector is selected as dominant direction.

Descriptor Computation. The actual descriptor is computed from a square region around the interest point oriented in direction of the dominant orientation. The size of the region is relative to the characteristic scale of the interest point. The region is subdivided in $4 \times 4 = 16$ sub-regions from which each four feature entries are calculated, resulting in a 64-dimensional feature vector.

The four descriptor entries for each sub-region are calculated from Haar-wavelet responses in x- and y-direction. To increase the contribution of responses closer to the interest point, the responses are weighted with a Gaussian window originating at the interest point. The weighted responses are summed up to form two of the descriptor entries. The other two are formed from the sum of the absolute values of the responses. If d_x denote the weighted responses in x-direction and d_y the weighted responses in y-direction, then we have the four descriptor entries for the subregion

$$v_1 = \sum d_x \qquad v_3 = \sum |d_x|$$
$$v_2 = \sum d_y \qquad v_4 = \sum |d_y| \quad .$$

An important aspect is the descriptor's invariance against illumination and contrast changes. Following the simple model in equation 2.13, SURF implicitly achieves illumination invariance by utilizing only differences of intensity values. Contrast invariance is accomplished by normalizing the descriptor to unit length.

3.1.3 Color SURF Descriptor

The SURF descriptor as described in the previous section was designed for intensity images only. In this section I propose two variants of the SURF descriptor that leverage additional information about the interest regions that can be gained from color images.

Color has not played much a role in local feature research since one of the major target applications is object class recognition. For object class recognition it is rather harmful to include color in the descriptor, because usually an object class is more defined by its shape than its color. For example, an airplane is an airplane because it has wings that make it fly. The airplane's color is irrelevant. In fact, by including color in a feature descriptor, one effectively increases the intra-class variance. This unnecessarily raises the number of training images required to learn the characteristics of the object class.

For specific person re-identification however, color is an important cue. It can be essential for distinguishing persons with clothes of similar shape and texture. Only considering intensity images unnecessarily discards information. The encoding of color in a local feature descriptor is an open research problem. Weijer and Schmid [43] proposed histograms

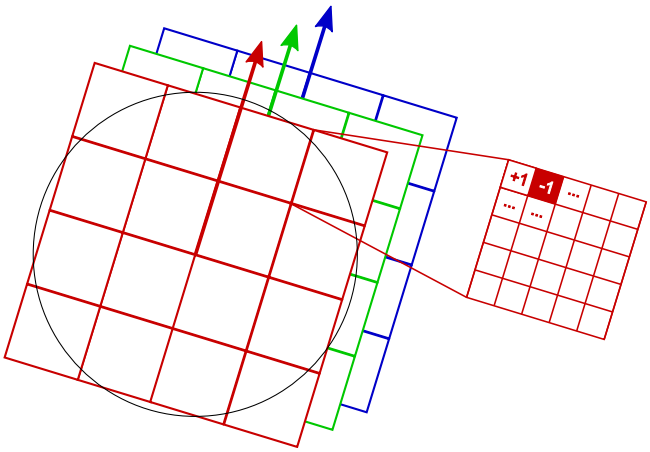


Figure 3.1: Multi-channel descriptor. The descriptor is formed by concatenation of three SURF descriptors, one for each color channel.

of color invariants as possible extensions to any local feature descriptor. However, these histograms are in the style of the SIFT descriptor [30], thwarting the speed of calculation of the SURF descriptor. The two color descriptors proposed in this chapter rather follow the the simple valuation of SURF that an approximation often performs as good as a more complicated descriptor, yet is far more efficient to compute.

Note that only the descriptor is modified. The interest point detection step and orientation assignment remain unchanged.

Multi-Channel SURF

In regions with colored texture, it is prudent to assume that not all information about the (colored) structure of the texture is contained in its respective intensity counterpart. This is especially the case if the structure elements of the texture are small compared to the considered feature region. The basic idea of the Multi-Channel SURF descriptor is to treat red, green and blue color channels independently and compute one independent component of the descriptor from each channel.

A color image can be divided into its red, green and blue channel. In all three color channels, pixel values lie within a fixed interval [a, b] (commonly used interval limits are 0 and 1, or 0 and 255). This is similar to intensity images. Since the original SURF descriptor has proven to successfully capture the local structure in an interest region, I propose to compute the SURF descriptor for each color channel independently. Each of these sub-descriptors is computed as if the respective color channel were an intensity image (see figure 3.1). The three sub-descriptors are then concatenated to form the Multi-Channel descriptor. This results in a descriptor that has three times the size of the original SURF descriptor ($64 \times 3 = 192$ entries).

Although the intensity image is a linear combination of the three color channels according to

$$\mathrm{Y} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{3.2}$$

the three parts of the Multi-Channel descriptor cannot be linearly combined to form the original SURF (intensity-based) descriptor. Recall that in the original SURF descriptor responses to Haar-Wavelets *and their absolute values* are integrated over sub-regions. So while the first two descriptor entries for each sub-region $\sum dx$ and $\sum dy$ are linear combinations of their respective color counterparts, this is not the case for the integral of absolute values:

$$|dx_{\mathrm{Y}}| \neq 0.299 \cdot |dx_{\mathrm{R}}| + 0.587 \cdot |dx_{\mathrm{G}}| + 0.114 \cdot |dx_{\mathrm{B}}| \tag{3.3}$$

$$\Rightarrow \sum |dx_{\mathrm{Y}}| \neq 0.299 \cdot \sum |dx_{\mathrm{R}}| + 0.587 \cdot \sum |dx_{\mathrm{G}}| + 0.114 \cdot \sum |dx_{\mathrm{B}}| \tag{3.4}$$

Hence, the original SURF descriptor is not implicitly contained in the Multi-Channel descriptor.

In order to keep the computational effort as low as possible, the interest point detection as well as the direction assignment of the descriptor are still computed from only one channel, for example the intensity channel. However, since the integral image of the intensity channel is not needed for the descriptor computation, this would waste computational effort. Instead, both interest point detection and orientation assignment should be computed from one of the color channels. However, I did not specifically conduct repeatability experiments to quantify the channel-dependence of the interest point detection and orientation assignment.

The Multi-Channel SURF descriptor has a serious drawback. It triples the computational effort of the feature description stage and, probably as important, triples also the size of the descriptor. Although a large descriptor with independent entries usually increases the discriminative power of the feature, it also increases the cost of matching two descriptors. Furthermore, in scenarios where memory is a limited resource, such as in smart cameras, it effectively reduces the number of storable descriptors by two thirds.

Channel-Difference SURF

The second color variant of the SURF descriptor that I propose, the so-called Channel-Difference descriptor, has less dimensions, therefore less memory requirements, and is also faster to compute. It consists of two parts. The first part is computed on the intensity image and is identical to the original SURF descriptor. The second part encodes color information. Thus, in contrast to the Multi-Channel descriptor, the original SURF descriptor is explicitly contained. Both parts have 64 dimensions each, resulting in a 128-dimensional descriptor. The color information is encoded in the following way.

Instead of computing the gradient in x- and y-direction, the difference between two color channels is computed. This *color gradient* implicitly encodes the color information. Independence of the overall luminance is achieved in the same manner as with the original SURF descriptor since the luminance offset (cf. equation 2.13) is canceled out when taking the difference between color channels.

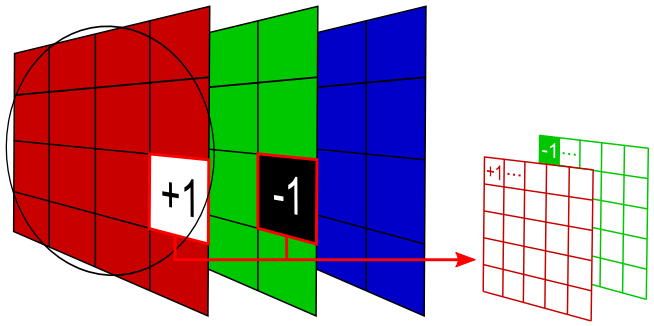


Figure 3.2: Channel-Difference extension. The descriptor extension is built from the sum of differences between the three color channels. The full descriptor is a combination with the original intensity based descriptor.

As feature region, a square region of size $20s$ around the interest point is taken, where s is the scale of the interest point. The feature region is oriented according to the interest point's dominant direction. This is the same region that is also employed for calculating the intensity part of the descriptor.

In order to capture the locality of the color differences, the feature region is divided into 16 sub-regions each of which is described by four descriptor entries. In each of the sub-regions, the color gradients are computed for 5×5 equally spaced square regions of size s which are weighted with a Gaussian ($\sigma = 3.3$) centered at the interest point (see figure 3.2). The values of the parameters are motivated by their respective counterparts in the original SURF descriptor.

The sum of gradients between the red and green channels as well as between the green and blue channels form the first two entries of the sub-region's descriptor entry. The color gradient between the red and blue channels is omitted since it is a linear combination of the other two color gradients and does not contain any new information. In order to also capture high frequency changes within a sub-region, the absolute values of the gradients are summed up as well and form the remaining two entries of the sub-region's descriptor. Thus, for sub-region S_i the four descriptor elements are calculated as

$$d_1^{(i)} = \sum_{(x,y)\in S_i} red(x,y) - green(x,y) \cdot g(\sigma, x, y) \tag{3.5}$$

$$d_2^{(i)} = \sum_{(x,y)\in S_i} |red(x,y) - green(x,y)| \cdot g(\sigma, x, y) \tag{3.6}$$

$$d_3^{(i)} = \sum_{(x,y)\in S_i} green(x,y) - blue(x,y) \cdot g(\sigma, x, y) \tag{3.7}$$

$$d_4^{(i)} = \sum_{(x,y)\in S_i} |green(x,y) - blue(x,y)| \cdot g(\sigma, x, y) \tag{3.8}$$

For the Channel-Difference descriptor, both interest point detection and orientation assignment can be performed on the intensity channel, since the integral image has to be computed for the intensity-based part of the descriptor anyway. The calculation of color gradients cannot generally gain from a transformation of the respective color channels to integral images.

3.2 Semi-local Features

Clothes often contain large homogeneous regions of similar color. Especially from a distance, clothes may appear to have no, or only a coarse-grained structure. Local interest point detectors however rely on underlying structures such as corners or blobs being present to be able to repeatably select interest points. On a plain red sweater for example the coverage with interest points from a standard interest point detector would be low.

There are a few interest *regions* detectors that select regions instead of interest points. However, in my preliminary investigations it turned out, that current state-of-the-art region detectors such as Maximally Stable Extremal Regions [28] or Maximally Stable Color Regions [44] only found very few interest regions on the frames of the evaluation data set of this thesis (see section 4.2.1). These detectors were designed with the intention of establishing pixel-accurate correspondences between two images for wide-baseline stereo. In order to avoid many false correspondences, interest regions are only selected if they fulfill strict criteria. I believe that the low resolution and challenging lighting conditions of the dataset prevent the selection of enough stable interest regions.

Rather than finding a few repeatable interest regions, I start by sub-segmenting the whole image into smaller parts. Mori et al. [45] followed a similar approach using normalized cuts [22], but their goal was only to find and segment the person in the image from the background. I however propose to use the extracted regions directly for recognition in the same manner as local features (see 3.3). It is not necessary that the shape and position of the segments are determined with pixel-accuracy. Far more important is that we achieve a division of a person's appearance into its relevant regions, which then can be described by a fitting descriptor. In contrast to Mori, I segment the image using the watershed segmentation algorithm [46] instead of normalized cuts for performance reasons. Furthermore, I adopt an agglomerative clustering algorithm to merge adjacent segments that are visually alike.

The resulting regions are described using color histograms. The description with a color histogram is of course not the only possible way. Other descriptors such as Hu Moments, histograms of texture filter responses or the MPEG7 Color Structure Descriptor, which performed promising in other recognition systems [23, 8], are viable options. Regardless of the employed region descriptor, the basic principles remain unchanged.

In the remainder of this section I will first briefly introduce the watershed segmentation algorithm and then explain the actual segmentation and clustering procedure. An overview of the segmentation and clustering procedure is depicted in figure 3.3.

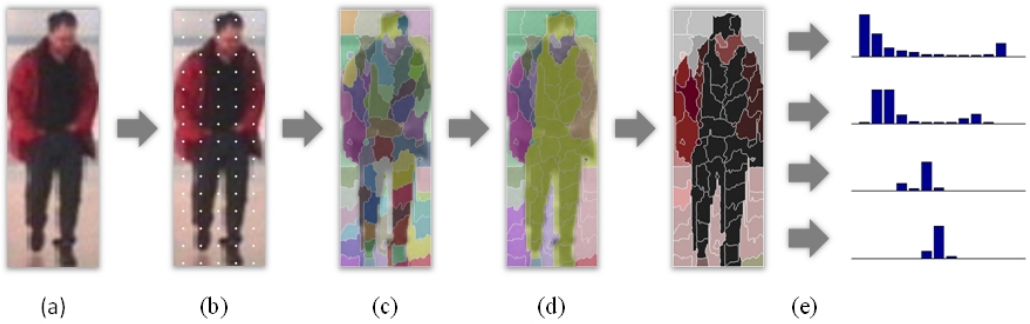


Figure 3.3: Overview over the extraction of semi-local features. (a) Input image. (b) Seed placement. (c) Watershed segmentation. (d) Clustering of similar regions. (e) Computation of color histograms as descriptors.

3.2.1 Watershed Segmentation

The segmentation procedure is based on a morphological region growing scheme known as *watershed transformation*. The underlying idea is to regard an image as a topographic surface where pixel intensities define the height of the corresponding point on that surface. Imagine that it rains and water pours onto the surface. Water will start accumulating in local minima, filling up basins around them. After a while, the water level rises high enough that water from a basin would flow over to neighboring basins. Dams are built at these lines to prevent leakage from one basin to its neighbors. These dams are also called *watershed lines*. They later mark the borders between segmented regions. The process is stopped when the water reaches the highest point of the surface.

In practice, the watershed transform starting at local minima produces a considerable over-segmentation of the image due to noise and irregularities in the input image. For segmentation purposes, the results can be improved if the watershed transform is started at arbitrary markers instead of local minima. This *marker-based watershed transform* can easily be fine-tuned to the number of resulting regions. Furthermore, multiple markers can be defined to end up in the same basin, i.e. no watershed line will be established between the regions resulting from these markers.

Segmenting Color Images

In a color image, different color values might map to a similar intensity. In this case neighboring regions are not sufficiently separable which can lead to a bad segmentation. The similarity measure between pixels has to be adapted to effectively capture color differences that would otherwise map to a similar intensity. A similarity measure that works well is the maximum of differences from all three color channels [46],

$$d(p_i, p_j) = \max(|R_i - R_j|, |G_i - G_j|, |B_i - B_j|) \tag{3.9}$$

where the C_i and C_j, $(C \in R, G, B)$ are the color values of pixels p_i and p_j of the respective color channel.

This color difference is suitable for clothing that is not overly textured. Small texture, if at all visible at low resolutions such as from a surveillance camera, might result in a fine-grained over-segmentation.

Marker Placement

If we do not consider minima as initial markers, the question remains how many markers are needed and where to place them. Without further information about the image structure equally spaced markers are a viable option. The remaining parameter is the step size between markers.

There is a tradeoff between granularity and accuracy as the watershed algorithm does not leave unsegmented regions. Every pixel of the original image is assigned to one region originating from the markers. If the number of markers is too low, the result would be regions with undesired high intra-region variance, negating our goal of finding regions of similar color and texture. It is therefore preferable to use many initial markers and accept in a first step even a subdivision of regions of similar appearance. These resulting smaller regions can then be re-merged based on a higher level distance metric (other than just pixel value differences) if they are alike enough.

The algorithm for merging similar regions is explained in the following section.

3.2.2 Region Clustering

In order to reduce a too fine-grained division of a visually otherwise compact region, neighboring regions that are sufficiently alike are merged to larger regions. The similarity of two regions is computed as χ^2-difference between the color histograms of the two regions. Here again, any other region descriptor instead of color histograms is possible. However, since our goal is to find regions that are of similar color or texture, the descriptor should of course capture some of those cues.

I adopt an agglomerative clustering algorithm in such a way that only neighboring regions are merged. The algorithm is depicted in table Algoritm 1.

Steps 8-14 of this algorithm can be efficiently implemented using a priority queue.

The advantages of clustering already at frame-level (in contrast to training-set-wide clustering) are twofold. First, on frame-level the neighborhood relationship is retained and only regions are clustered that actually belong together. Second, frame-level clustering keeps the number of features resulting from a frame somewhat independent of the marker step size. This is important as a resulting higher number of features increases the computational effort of subsequent training and matching steps.

Algorithm 1 Merging Similar Regions

1. **for** each watershed region i **do**
2. compute descriptor $\mathcal{H}_i$
3. **end for**
4. compute neighborhood matrix $\mathcal{N}(i,j)$
5. **for** each neighboring pair (i,j) of watershed regions ($\mathcal{N}(i,j) = 1$) **do**
6. compute pair-wise difference $d(i,j) = \chi^2(\mathcal{H}_i, \mathcal{H}_j)$
7. **end for**
8. **while** $d(i,j)$ between two neighboring regions is smaller than a threshold **do**
9. merge regions i and $j \rightarrow R_{new}$
10. compute the descriptor for R_{new}
11. remove regions i and j
12. update $\mathcal{N}$, including R_{new}
13. compute $d(k, R_{new}) \quad \forall k$
14. **end while**

3.3 Person Recognition

In this section I will describe how the local and semi-local features are used for person recognition. The approach builds on the insights from object recognition approaches using local features (see section 2.3) that it is not necessary to establish exact correspondences between all features in training and test images. Already a subset can provide enough information. For both training and classification the input are short image sequences showing mainly one person.

The principal recognition procedure is depicted in figure 3.4. First, a feature database is built for each person from training images. Then, for recognition, the distances of a test image to each of the trained person models are calculated. The distance between the test image and a person model is computed as sum of nearest neighbour distances between all the test image's features and the person's feature database.

In the following, both the training and the recognition step are explained in detail.

3.3.1 Training

In the training stage, models are trained in a supervised way for each of the persons that are to be recognized. It is assumed that each training sequence mainly shows only one person. Such a sequence might be the output of a generic person tracker, producing subsequences from larger video sequences and cropping the whole scene to an approximate bounding box around one person. Figure 3.5 shows one such training sequence.

Each sequence in the training data is annotated with the identity of the person. Note that only the whole sequence itself is labeled. The training stage does not know which of the features extracted from the frames actually correspond to the person and which come from

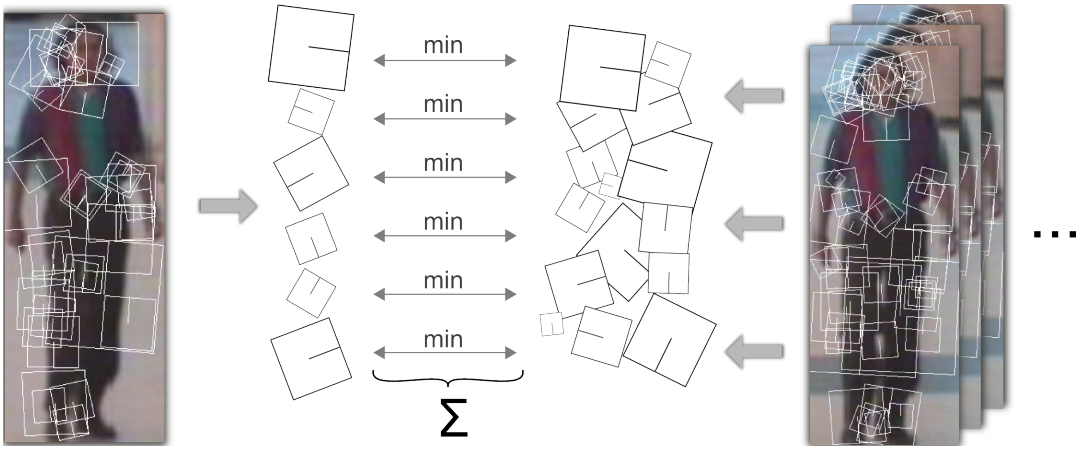


Figure 3.4: Matching descriptors. Descriptors are extracted from a test image (left). These descriptors are matched against a feature database which was built from several training images (right). The distances to the respective nearest neighbors in the database are summed up to compute the distance of the test image to the model.

Figure 3.5: Example training sequence. The person's pose and posture already varies substantially in this short sequence. For more training sequences see figure 4.6.

background or occluding other persons. In order to test the limits of this approach, explicit background segmentation is omitted.

Local and semi-local features are extracted from these training sequences as described in sections 3.1 and 3.2. In the experiments in chapter 4, the length of a training sequence is around 40 frames with around 50 features each. This results in approximately 2000 features per sequence.

Bag-of-Features Model

A person is modeled using a *bag-of-features* representation. This model was chosen for its simplicity and reported good performance in object recognition tasks (e.g. [47]). The basic idea is to characterize a person based on a collection of features without taking into account their spatial distribution. As for object recognition tasks, the underlying assumption is that the local features by themselves already provide enough information about the person.

Features are collected individually for each person. The whole set of features (or a reduced set obtained from clustering as explained in the next section) constitutes the representation of the person. This is in contrast to the usual approach for object recognition, where either feature-object-class pairs or histograms of visual words are used as model (see section 2.3).

Clustering

In general, clustering attempts to find groups of similar features given an unordered set of noisy samples. In *hard clustering* each feature vector is assigned to exactly one cluster (in contrast to *fuzzy clustering* where each feature belongs to each cluster with some weight between 0 and 1). In this thesis, only hard clustering is considered. Clustering of features can improve the person models in two ways. First, it can be used to find the "true" features, removing unwanted features resulting from background and clutter. Second, it can speed up the recognition stage by reducing a large number of single features to a few representative clusters.

Clustering can be seen as a means to build a *visual codebook*. A visual codebook contains a set of features that are sought to be a visual alphabet for describing the objects or persons in an image. For object recognition tasks one often builds one generic visual codebook for describing all object classes. An object is then modeled as histogram of visual word occurrences (see section 2.3.2). However, in the case of specific person recognition this is more harmful than helpful. A visual codebook across different persons is expected to rather learn the general characteristics of a human (e.g. head, body, legs etc.) than specifics of one individual. For that reason, the clustering is performed on each person's features individually, resulting in a visual codebook for each person.

The intra-cluster variance can serve as a normalization measure for distances from the cluster. If one considers the full covariance matrix, one such distance is the *Mahalanobis Distance*. The Mahalanobis distance from a cluster with mean $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_N)$ and covariance matrix $\boldsymbol{\Sigma}$ is defined as

$$d_{\text{Mah}}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})} \tag{3.10}$$

For computational efficiency only the diagonal entries of $\mathbf{\Sigma}$ are considered, reducing the Mahalanobis distance to a normalized version of the Euclidian distance

$$d_{\text{NormEuc}}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T {\mathbf{\Sigma}_{\text{diag}}}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \quad . \tag{3.11}$$

Pruning Small Clusters

Common clustering techniques take as parameter the number of clusters or a maximum distance threshold for merging clusters, although the underlying distribution is usually unknown. The number of features that end up in a clusters is usually undetermined beforehand. It therefore indicates the support of the sampled features for this cluster and can be used to filter the results from the clustering to increase robustness in the later matching step. The number of features forming a cluster is a natural weight for the respective feature vector. The underlying assumption is that features describing the actual persons occur more frequently than features from background or clutter. However, I do not explicitly compute a weight for each cluster. Instead clusters that are not supported by enough features (i.e. only consist of fewer than t features) are removed, assuming they result from noise, clutter or background. Another possibility is to keep only the k largest clusters. This has the advantage that each person model contains the same number of clusters, hence no model is preferred during recognition only because of a higher number of features.

Limiting Cluster Variance

Due to the low resolution of the images we expect a lot a variance even in features that do not result from background or noise. This is especially a problem if the number of features is generally small ($<< 10000$), but there are large relative differences between the number of extracted features (e.g. for person 1 we have 2000 features whereas for person 2 we have 4000). As a result from the higher number of features, variances around cluster centers will be larger. This in turn results in smaller distances between test features and the model, if the distance is normalized by the cluster variance. Hence, a nearest neighbor classifier more likely would pick the model with the most training features.

The impact of large variability around the cluster center can be mitigated by restricting the variance components to a narrow interval. Although this does not completely remove the influence of outliers on the cluster variance, it effectively reduces its impact. I found this especially necessary for person identification, since we have an instance recognition problem instead of the usual object-class recognition problem. Another possibility is to not use the cluster variance for distance normalization at all.

Clustering Algorithms

I used two different clustering algorithms for feature clustering: *k-means clustering* and *hierarchical agglomerative clustering*.

The k-means algorithm [48] iteratively computes the means of k clusters. Each mean denotes the center of one cluster. The algorithm works as follows. First, the k cluster centers

are placed randomly in the feature space. Second, all features are assigned to their respective clusters, given by their nearest cluster center. The nearest cluster center is determined according to a pair-wise distance measure $d(\cdot, \cdot)$. Third, the new cluster centers are computed as means of all features belonging to the same cluster. Step two and three are repeated either for a fixed number of times or until the intra-cluster variance reaches a threshold. The algorithm can also be stopped if the cluster assignment of features does not change significantly between two steps. In practice, the algorithm converges after a few steps. Its advantage lies in its simplicity and speed. The number of clusters k is a parameter that has to be chosen beforehand. However, in general the true number of clusters is unknown.

Agglomerative clustering does not need a predefined number of clusters. The basic principle of agglomerative clustering is quite simple. In the beginning, every feature forms its own cluster. In each step, the two clusters with the smallest inter-cluster distance are merged to a new cluster. This results in a hierarchy of clusters with one less cluster in each hierarchy step.

One of many possible distance measures between clusters for agglomerative clustering is the *group average distance*. It is based on an underlying pair-wise distance metric $d(\cdot, \cdot)$ between cluster elements. The distance between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is defined as the average distance of all cluster elements

$$d(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{|\mathcal{C}_i| \cdot |\mathcal{C}_j|} \sum_{\boldsymbol{x} \in \mathcal{C}_i} \sum_{\boldsymbol{y} \in \mathcal{C}_j} d(\boldsymbol{x}, \boldsymbol{y}) \tag{3.12}$$

Other distance measures are the *maximum distance* or *minimum distance* between cluster elements. The group average distance usually leads to compact clusters with reasonable cluster centers and sizes.

A *reduction property* defines how the distance between clusters is affected by the merging of two clusters. Consider three cluster $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. The group average distance has the following reduction property,

$$d(\mathcal{A}, (\mathcal{B} \cup \mathcal{C})) = \frac{|\mathcal{B}|}{|\mathcal{B} + \mathcal{C}|} d(\mathcal{A}, \mathcal{B}) + \frac{|\mathcal{C}|}{|\mathcal{B} + \mathcal{C}|} d(\mathcal{A}, \mathcal{C}) \tag{3.13}$$

The weighting factors compensate for different cluster sizes, so that the resulting reduction is *not* influenced by the cluster sizes. That is why this reduction property is also called *Unweighted* Pair Grouping Method with Arithmetic mean (UPGMA). A reduction property is *convex* if the distance of cluster $\mathcal{A}$ to the merged cluster $(\mathcal{B} \cup \mathcal{C}))$ always lies between the distances of $\mathcal{A}$ to $\mathcal{B}$ and $\mathcal{C}$. The group average distance has a convex reduction property regardless of the underlying distance metric.

A convex reduction property allows for a time efficient implementation of agglomerative clustering [49]. Depending on the underlying distance measure, it can also be implemented *space* efficiently. Space efficiency becomes important when dealing with thousands of features, for which the distance between each pair of features has to be computed. If the number of features is in the order of 10000, the pair-wise distance matrix cannot be held in the system memory of current end user PCs anymore. A space efficient implementation is possible for example for the $L2$ metric [50].

3.3.2 Classification

In the matching or classification stage it is determined which of the known persons is shown in a new test sequence. The output of the classification stage is a score for each known person corresponding to how well a test image or frame matches the trained models. As in the training stage, features are first extracted from each of the frames in the sequence. The number of features is unknown in advance.

After the features are extracted, they are matched against the person models. For each feature a distance to each person model is computed. Recall that the person model consists of a set of features or feature clusters (represented by their cluster center and variance). The distance between a feature and a person model is the minimum distance between the feature and any feature from the model

$$d(\boldsymbol{x}, \mathcal{P}^{(i)}) = \min_k d_f(\boldsymbol{x}, \mathcal{P}_k^{(i)}) \tag{3.14}$$

where $\mathcal{P}^{(i)}$ is set of features modeling person i, and $\mathcal{P}_k^{(i)}$ is the kth feature within this set. The distance metric $d_f(\cdot, \cdot)$ between the individual features can be chosen arbitrarily depending on the feature type, for example $L2$ or Euclidian distance for SURF features and χ^2 distance for histograms. This step can be sped up by using efficient data structures such as kd-trees [32] for storing the features.

If the model consists of clusters instead of individual features, the variance of the cluster can be taken into account by normalizing the distance from the cluster center $\boldsymbol{\mu}$ in each feature dimension by the respective cluster variance σ_i in that dimension by using the normalized Euclidian distance (cf. section 3.3.1)

$$d_f(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T {\boldsymbol{\Sigma}_{\text{diag}}}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})} \tag{3.15}$$

However, for the recognition of individual persons in low resolution images, I found that using the variance for normalization harmed the recognition performance.

The distance between the test image and the person model is now obtained by adding up the individual distances of all features to that model

$$d(\boldsymbol{\omega}, \mathcal{P}^{(i)}) = \sum_k d(\omega_k, \mathcal{P}^{(i)}) \tag{3.16}$$

A smaller distance indicates better agreement between test image and model. Hence, the model with the smallest distance is chosen as correct match.

This classification procedure works on single frames. The distances are computed for each frame in a video sequence individually. This neglects information that could be gained from tracking for instance individual local features and thus finding more stable features. It does however avoid problems when the camera itself is in motion, e.g. when the camera is mounted on a personal robot which is moving freely around. It also allows the approach to work on high resolution still images, e.g. from a personal photo album. This is however not further pursued in this thesis.

3.3.3 Temporal Fusion for Video

Given model distances for each frame, it is possible to fuse these over several frames. In this way even those frames can be classified correctly for which the frame-based classification could not determine the correct identity.

Sequence-level Decision Fusion

All distances within a sequence are fused so that unified classification of the whole sequence is achieved. Outliers are averaged out. The fusion is based on the sum-rule with additional normalization. A score for a sequence is calculated from the individual frame distances as follows:

1. *N-best Selection*. The n smallest distances (e.g. $n = 3$) from each frame are selected. All other distances are discarded.

2. *MinMax-Normalization*. The n smallest distances from step (1) are re-normalized to the interval $[0, 1]$. The smallest distance d_{min} of the n distances is mapped to 1, the maximum d_{max} is mapped to 0 and all remaining distances are mapped linearly between 1 and 0 according to

$$s_i^* = \frac{d_{\text{max}} - d_i}{d_{\text{max}} - d_{\text{min}}} \tag{3.17}$$

   This effectively turns distances into scores, where a higher score indicates a better match. The normalization is necessary to compensate differences between frames resulting e.g. from varying numbers of features.

3. *Norming to Unit Length*. The resulting minmax-normalized scores s_i from the previous step are normed so that their sum $\sum_i s_i$ equals 1. This achieves an implicit higher weighting of scores that have a large distance to their respective next best score.

4. *Fusion*. The resulting normalized scores from all frames in the sequence are added up (sum-rule fusion) and normalized by the number of frames N of the sequence. This results in an average score per frame,

$$s_{avg} = \frac{1}{N} \sum_{i}^{N} s_i^* \tag{3.18}$$

   The normalization by the number of frames is necessary for open-set recognition scenarios. The decision if the person is known or unknown will be based on whether the score is higher or lower than a threshold θ. Sequences with only a few frames would otherwise always be biased towards the impostor class.

I also experimented with non-normed scores for the frames and further minmax-normalization on sequence score level, but did not find any significant improvement over the described fusion method.

3.3.4 Fusion of Feature Types

It is prudent to assume that no single (simple) descriptor scheme is sufficient to reliably and repeatably capture the necessary information to achieve perfect recognition performance. Especially contrasting features such as SURF and watershed regions promise to complement each other well.

The fusion of feature types is performed at frame level. For each frame a combined score resulting from both SURF and watershed regions is computed. The combination method is similar to the temporal fusion from the previous section.

First, for both feature sets the respective n smallest distances are individually minmax--normalized to an interval between 0 and 1, and then again normalized so that their sum equals 1 (cf. equation 3.17). This is important as the scores of the two features can be in different value ranges. Without normalization, the fusion might always be unintentionally biased towards one feature. The combination of both feature scores is done by calculating the weighted sum

$$s_i^* = \alpha \cdot s_i^{\text{SURF}} + \beta \cdot s_i^{\text{watershed}} \tag{3.19}$$

The weights α and β allow to give more importance to one feature than the other. This scheme can further be combined with temporal decision fusion and also easily be generalized to more than two features.

4 Experiments

In this chapter I present experiments conducted to evaluate the performance of the proposed approach. In the first section, the two proposed color SURF descriptors are investigated using a standard data set for local features. The second section contains the results of the person recognition approach in a surveillance scenario.

4.1 Color SURF Descriptor Evaluation

In this section the performance of the color SURF descriptors from section 3.1.3 is evaluated against the standard SURF descriptor. The performance criterion is *recall* versus *precision* in a one-to-one matching task of descriptors between two images.

4.1.1 Local Feature Evaluation Data Set

Mikolajczyk's data set[1] for the performance evaluation of local features is well established. It has been used for an extensive comparison of affine region detectors and local descriptors [35, 40] and is practically *the* standard data set on which new detectors and descriptors are evaluated (e.g. [44, 24]). The data set consists of several image series each exhibiting one of several common changes in image conditions that an interest point detector and descriptor should be able to cope with (cf. figure 4.1):

- *Viewpoint Changes*. The *graffiti* and *wall* series each show a mostly planar wall from different viewpoints. The viewpoint changes from frontal (perpendicular to the wall) up to a viewpoint angle of approximately 70 degrees.
- *Scale Changes*. The *bark* series depicts a bark in various grades of scale and rotation. The scale changes up to a factor of four. The *boat* series from the original data set was not considered for this evaluation. It is a second series for evaluating the influence of scale and rotational changes, but consists only of gray scale images which makes it unsuitable for the evaluation of the color surf descriptors.
- *Image Blur*. The *trees* and *bikes* series show an ally of trees and some motorbikes, respectively, with different amount of blur resulting from varying zoom and focus of the camera.
- *JPEG Compression*. The *ubc* series shows a house. Each image in the series is JPEG compressed with a gradually worse compression quality ranging from 40 to 2 %.

[1] The data set is available at http://www.robots.ox.ac.uk/~vgg/research/affine

- *Illumination Changes*. The *leuven* series shows cars in front of a house. Illumination is gradually reduced by varying the camera aperture.

In figure 4.1 the first and fifth image from each of the series are shown. The first image always is the reference image of the respective series and the fifth already contains considerable changes in the respective image condition. Unless otherwise noted, these are the two respective images used for the descriptor evaluation.

For each image series, ground truth homographies describe the spatial relation of the two images. They were computed using automatically detected and matched interest points. This was done by the publishers of the data set and used as is.

4.1.2 Evaluation Criterion

The performance of the descriptor is evaluated by setting the number of correct matches between the local features of two images in relation to the number of false matches. This relation is presented as recall-vs-precision curve. This approach is similar to the evaluation in [40].

As ground truth, a matching of interest points has to be established without actually using the descriptors. Correspondences of interest points can be computed by transforming the interest point locations in the test image according to the ground truth homographies that were provided with the data set. Of course one cannot expect a perfect matching between interest points. Therefore, two interest points are considered as true correspondence if their feature region overlaps to a certain amount. More precisely, the overlap error ϵ_O must be smaller than 50%. Hence, the number of ground truth correspondences is the total count of matched interest points with $\epsilon_O < 0.5$.

Correspondences between descriptors are established using two different matching strategies. First, for *nearest-neighbor matching* (NN) a descriptor from the reference image is matched to its closest descriptor from the test image (closest according to the L2 metric), if the distance is below a threshold. Second, for *nearest-neighbor distance ratio matching* (NNDR) a point in the test image is only matched if its distance to the nearest neighbor is smaller than t times the distance to the second nearest neighbor, i.e. $d_{1st}/d_{2nd} < t$ with $t < 1$. Using the ground truth correspondences (computed as described in the previous paragraph), one can now calculate the number of correct and false matches.

The performance of the matching is presented as recall-vs-precision curve. *Recall* is the number of correct matches in relation to the number of total possible correspondences

$$recall = \frac{\#correct\ matches}{\#total\ correspondences} \tag{4.1}$$

and *precision* is the number of correct matches in relation to the sum of false and correct matches.

$$precision = \frac{\#correct\ matches}{\#correct\ matches + \#false\ matches} \tag{4.2}$$

Note that in the evaluation the $recall$ is plotted over $1 - precision$.

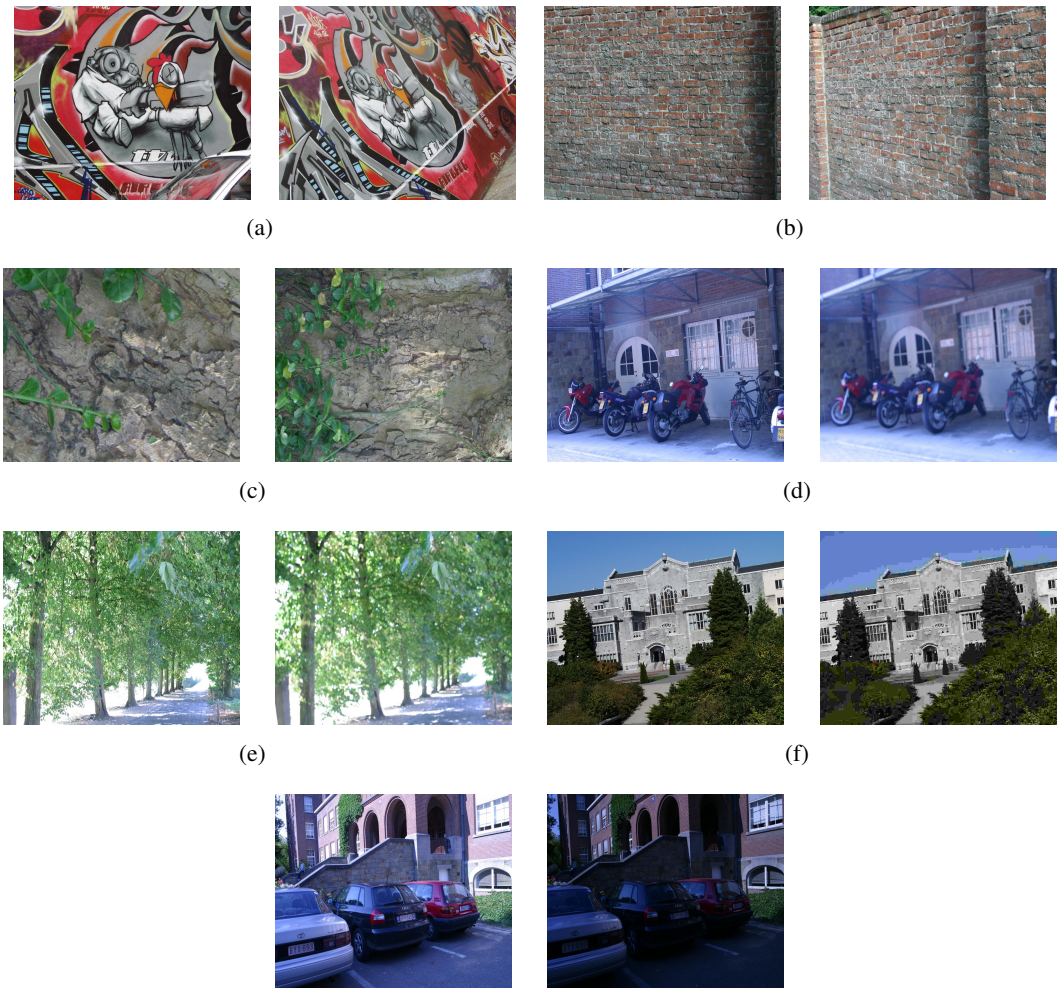


Figure 4.1: Local Feature Evaluation Data Set. (a), (b) Changes in viewpoint. (c) Changes in scale and rotation. (d), (e) Image blur. (f) JPEG compression. (g) Illumination. The image pairs are the ones used for the evaluation. The first (left) is the reference image, the second (right) image contains considerable changes in the respective image condition.

4.1.3 Results

In this section the results for the evaluation of the color SURF descriptors are presented. The performance of the Multi-Channel SURF descriptor and the Channel-Difference SURF descriptor is compared with the performance of the original SURF descriptor. The comparison is performed for viewpoint changes, zoom and rotation, illumination, blur and JPEG compression changes.

However, before this, the influence of the matching strategy is investigated. The two matching strategies, nearest-neighbor matching and nearest-neighbor distance ratio matching, were already introduced in the previous section. Figures 4.2a-d show the results of the comparison. The matching strategy only has a small influence on the performance of the descriptors. More importantly, the relative performance difference of the three descriptor schemes is unaffected. The NNDR matching strategy generally has a higher precision. This can be explained by NNDR's additional pruning of descriptors for which several potential matches have a similar distance. By adjusting the ratio threshold one can make this pruning more or less strict.

Since the ranking of the descriptor performances does not seem to be affected by the matching strategy, for the following experiments only results for the NNDR matching strategy will be reported.

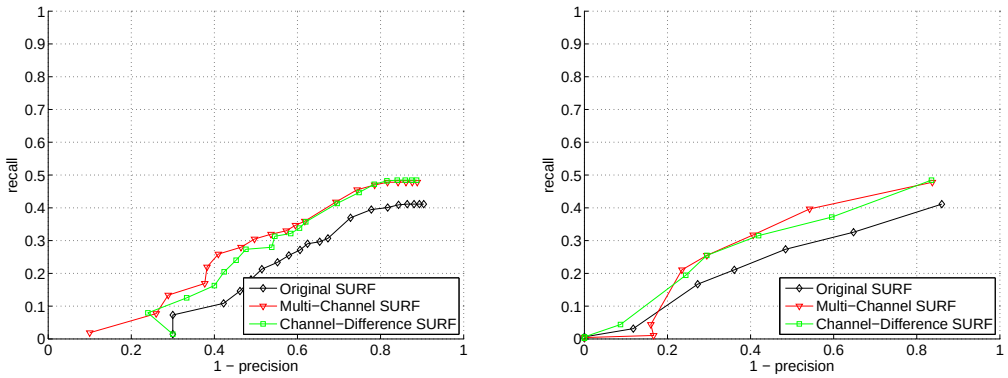
Viewpoint Changes

Changes in viewpoint are commonly observed changes in person recognition scenarios, for example in surveillance applications where we expect to see persons from different cameras. In addition, the viewpoint of local parts of a person effectively changes constantly due to small local pose changes of the person, e.g. during walking.

The performance of the descriptors is evaluated on two different scene types. The first series (graffiti) shows a structured image on a wall, whereas the second series (wall) shows a textured wall with repeating bricks (see figures 4.1a and 4.1b). The viewing angle is changed between 40 and 60 degrees relative to the reference image. These changes pose the biggest challenge to the SURF descriptor since the interest point *detector* does not normalize the feature region with respect to affine transformations.

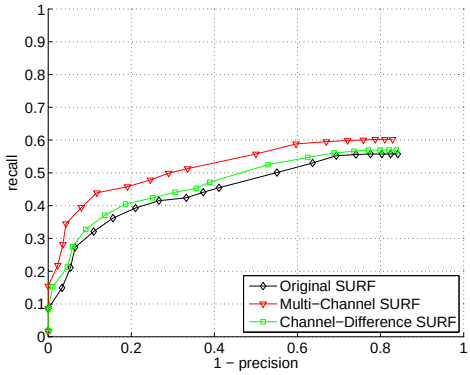
The results of the evaluation can be seen in figure 4.3. For both the structured and the textured scene the color descriptors provide an improvement over the original SURF descriptor. For the textured scene the relative improvement is higher for larger view angles (cf. figure 4.3d). In the structured scene the Channel-Difference descriptor performs as good as the slower Multi-Channel descriptor, whereas in the texture scene the Multi-Channel descriptor outperforms both the original and the Channel-Difference descriptor. This indicates that the Multi-Channel descriptor is able to use small color variations of the wall texture, which cannot be captured by the Channel-Difference descriptor. In the structured scene however, where the color is rather homogeneous over larger regions, there are much less such small variations and both descriptors perform equally.

It is important to notice that the performance improvement does not only result from the increased dimensionality of the color descriptors. An extended shape descriptor for SURF

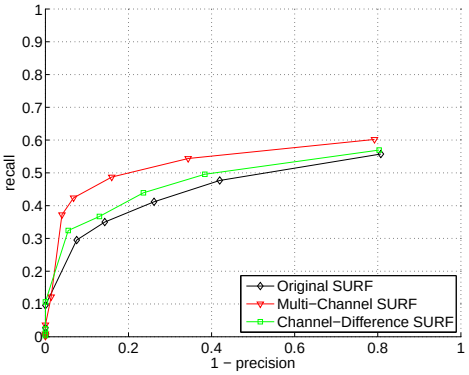


(a)

(b)



(c)



(d)

Figure 4.2: Comparison of matching strategies. (a), (c) Nearest Neighbour Matching. (b),(d) Nearest Neighbour Ratio Matching. The descriptors in the upper row are computed on images 1 and 4 of the graffiti series (fig. 4.1a). The descriptors in the bottom row are computed on images 1 and 5 of the wall series (fig. 4.1b). The relative performance of the color extensions in comparison to the original SURF descriptor is independent of the matching strategy.

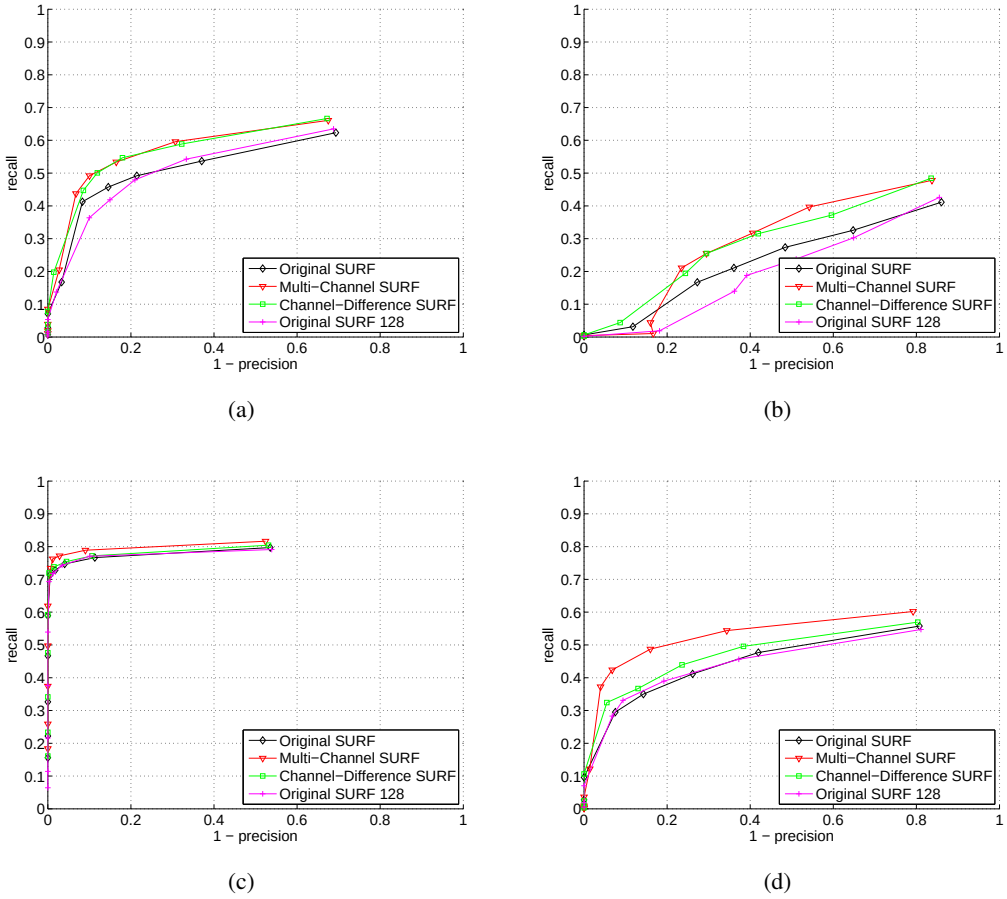


Figure 4.3: Evaluation for viewpoint changes of approximately 40 and 60 degrees. (a), (b) Results for a structured scene (graffiti series). (c), (d) Results for a textured scene (wall series). In all cases the color extensions outperform the original SURF descriptors. For the structured scene the faster Channel-Difference SURF even performs on-par with the Multi-Channel SURF extension.

was proposed by the original authors of SURF [24]. The number of entries was doubled (to 128) by including more details about the intensity patterns in the local feature region. This extended descriptor has the same number of dimensions as the Channel-Difference descriptor. The performance of the SURF128 descriptor is also given in figures 4.3a-d for comparison. In contrast to the color extensions, there is no performance improvement. This indicates that the increased distinctiveness of the color descriptors actually comes from including color in the descriptor and not just from increasing the dimensionality.

Zoom & Rotation, Illumination, Blur, JPEG Compression

Figures 4.4a-e display the results for the evaluation of the descriptor extensions for zoom and rotation, illumination, blur and JPEG compression. Since there is no large change in viewpoint, the feature regions are not subject to perspective transformations.

For all changes in image conditions there are no significant performance improvements over the original SURF descriptor.

Only for changes in blur (figures 4.4c-d), the Multi-Channel SURF descriptor seems to provide a small advantage over both the original SURF descriptor and the Channel-Difference extension. Both test images contain a lot of texture which favors again the Multi-Channel descriptor (as already seen for viewpoint changes).

In figure 4.4a both the original SURF descriptor and the Channel-Difference descriptor perform equally, while the Multi-Channel descriptor shows a small performance degradation. Recall from section 3.1.3 that the Multi-Channel does neither explicitly nor implicitly include the intensity SURF descriptor, while the Channel-Difference descriptor does. This indicates that here the relevant information comes from the intensity patterns alone. However, this is not surprising since the image does not exhibit many local color changes. The same is probably true for the illumination changes (figure 4.4b). Since there are less colors (combinations of color values) when luminance is low, color does not play an important role.

4.2 Person Recognition Evaluation

In this section, the performance of the proposed person recognition approach is evaluated. The scenario is a surveillance application. The task for the system is to identify the person in an image sequence based on his appearance. Only a subset of the presented persons are known to the system and only these known persons are to be identified. All other persons are to be classified as ‘unknown’. This is usually called *open-set classification*.

Open-set classification is a typical task in surveillance and security applications. For example, consider a person identification system at a front desk of a building. It is trained with appearances of all persons that work in the building and are allowed to enter it. However, it does not know what persons look like that are *not* allowed to enter the building. Only trained with the known persons, the system must decide whether the persons is allowed to enter the building or not. Furthermore, *if* the person is known to the system, it is desirable to identify

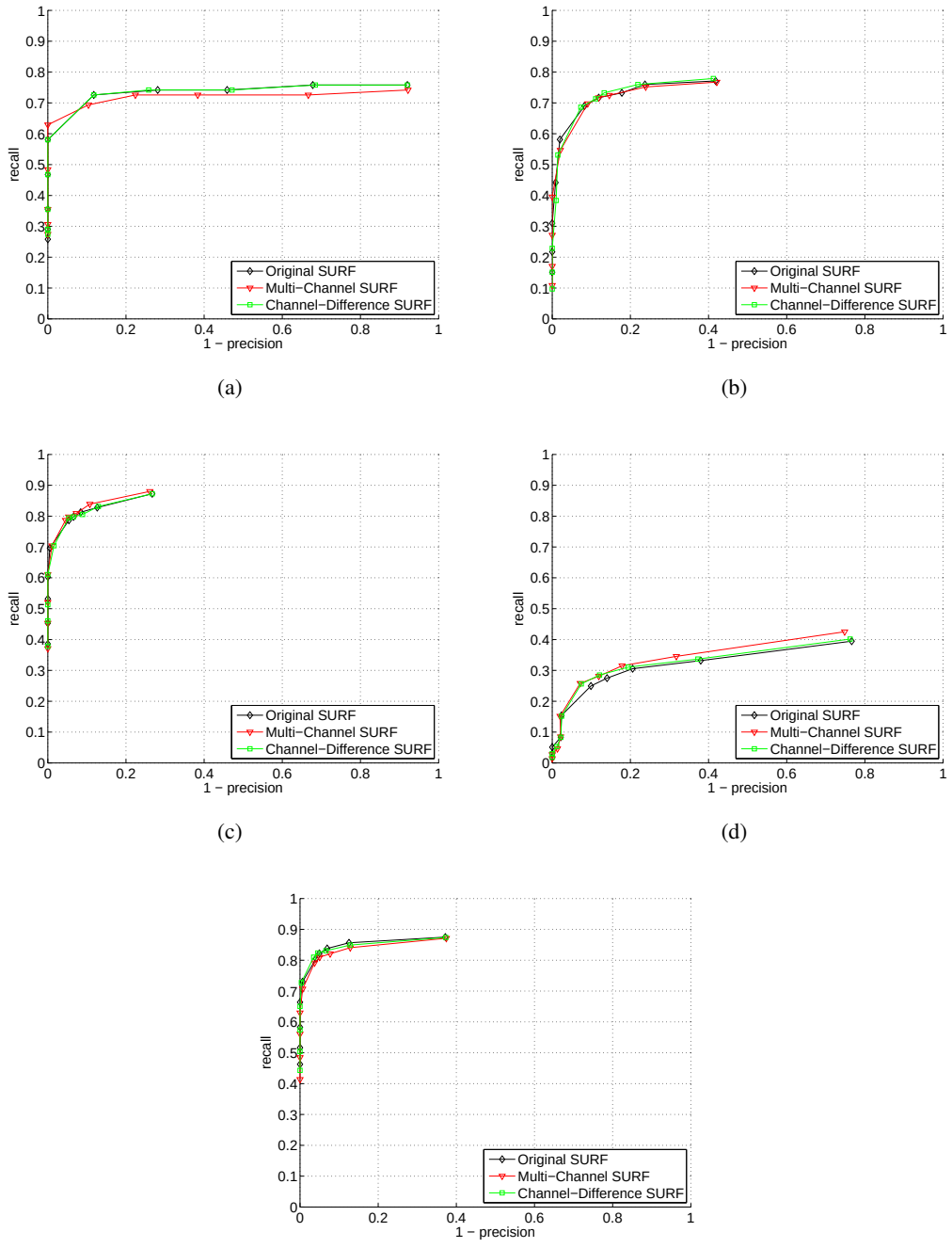


Figure 4.4: Evaluation for various changes in image conditions. (a) Zoom and rotation (bark series). (b) Illumination (leuven series). (c), (d) Blur (bikes and trees series). (e) JPEG compression (ubc series). There are no significant changes in performance in comparison to the original SURF descriptor.

him, for example to record his time of arrival or to establish a list of the persons that are in the building at the time.

The performance criterion is the number of correctly identified persons in comparison to the number of mis-identified persons and persons that were falsely accepted as 'known'. The results are presented as ROC curves extended for open-set classification as described in section 4.2.2.

4.2.1 Shopping Center Data Set

The proposed method for appearance based people recognition is evaluated on a realistic data set of short video sequences taken in a Lisbon shopping center. It consists of a series of short image sequences (on average 41 frames) in low resolution (on average around 47×118 pixels, cf. figure 4.5).

The data set is a subset of the publicly available CAVIAR data set[2] which was used for instance for the PETS04 workshop competition [51]. The original CAVIAR data set shows people walking down a corridor in a shopping center in Lisbon and their interaction with each other. It consists of 26 clips with lengths of around 1500 frames each. The resolution of the clips is 384x288 pixels and the frame rate is 25 frames per second. Due to synchronization issues with another camera occasionally two consecutive frames are identical. Bounding boxes of the tracks of the individual persons were already annotated in the original data set.

I extracted 281 short sequences from these tracks using the annotated bounding boxes (see figure 4.6 for some exemplary sequences). The tracks simulate the output of a person tracker.

Most of the extracted short sequences have approximately 40 frames and show exactly one individual. Subsequently, I assigned a unique identifier to all tracks that show the same individual. As the goal is person identification from full-body appearance under the assumption that people do not change their clothes significantly between training and recognition, individuals that appear with significantly different clothing have been assigned multiple identifiers, e.g. the person in rows 1 and 6 of figure 4.6. This resulted in a total of 53 different person, of which 10 have more than 6 sequences.

Because of articulated body movements and different distances of the persons from the camera, the sizes of the extracted frames are not fixed. See figure 4.5b for the distribution of frame heights and width. The large variance results from varying distances of the persons to the camera, articulated body movements and varying poses. The frames in the lower right of the plot are due to bounding boxes that clipped the persons' upper body when he or she was very near to the camera so that the legs were not visible anymore (see figure 4.5c).

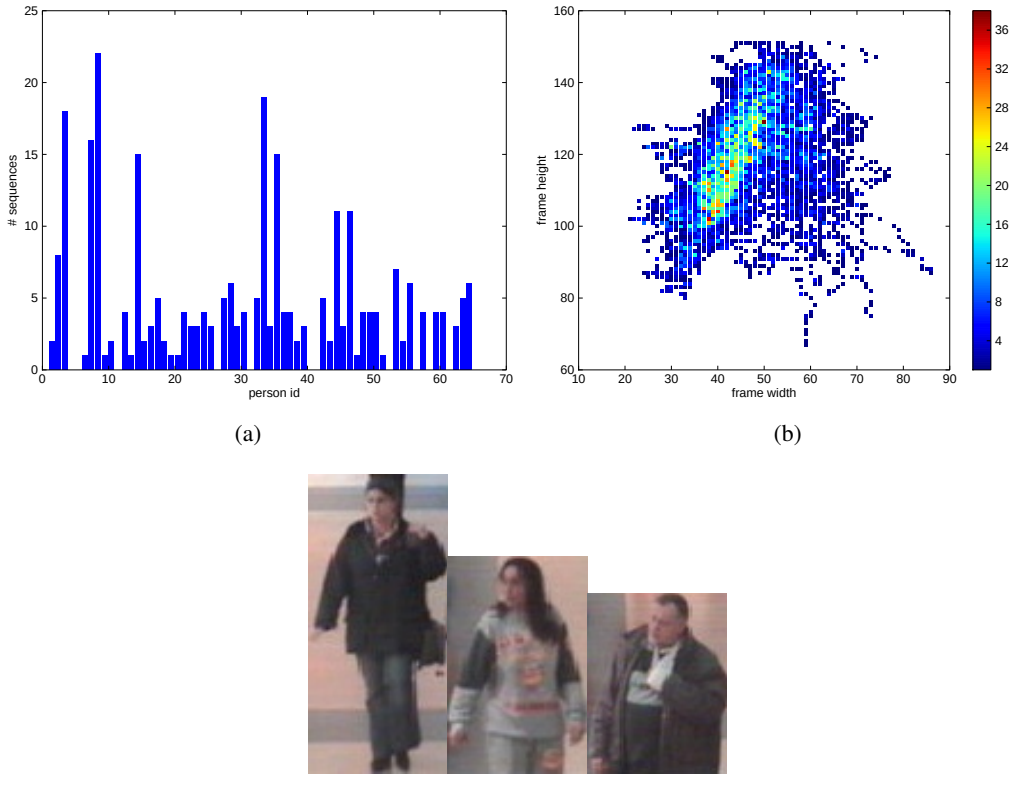
4.2.2 Evaluation Criteria

The proposed features and methods are evaluated in an open-set classification task. For a formal description of the used performance metrics, I will shortly introduce the following notation.

[2]The original data set can be downloaded from http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

Person Recognition Evaluation Data Set	
number of sequences	281
average sequence length	41 frames
number of persons	53
number of persons with more than 6 sequences	10
average frame width	47 px
average frame height	118 px

Table 4.1: Data Set Quick Fact Sheet.



(a) (b)

(c)

Figure 4.5: (a) Number of sequences of each of the 53 persons. For 10 persons there are more than 6 sequences available. (b) Frame width vs. height. Most frames have a height between 100 and 140 pixels and a width between 35 and 50 pixels. (c) Examples of different aspect ratios in the data set. When a person is too close to the camera, only his or her upper body can be used to perform the recognition.

Figure 4.6: Subset of the data set. There is considerable amount of variance regarding the persons' posture, size and point of view. The actual sequences are longer, containing around 40 frames each.

Let us denote a set of n known persons as S,

$$S = \{S_1, S_2, ..., S_n\} \tag{4.3}$$

The classifier will be presented with test sequences x_k^i, where k denotes the number of the sequence and i is the ground truth, i.e. person S_i is shown in the sequence. A test sequence that shows one of the known persons, i.e. $i \in 1, \ldots, n$, is denoted as *genuine test sequence*. The ground truth of all persons treated as unknown is set to $i = -1$. A sequence that shows such an unknown person is referred to as *impostor test sequence*.

Hence we can denote the sets of genuine and impostor test sequences as

$$X_{\text{genuine}} = \left\{x_k^i | i \in 1, \ldots, n\right\} \tag{4.4}$$

$$X_{\text{impostor}} = \left\{x_k^i | i = -1\right\} \quad . \tag{4.5}$$

Closed-Set Classification

Closed-set classification refers to the task of classifying a feature vector into one of a finite set of known classes. In the case of person recognition this means, that given a frame or sequence showing a person, the task is to decide which individual from a finite group of people is shown. A closed-set classifier can be regarded as a function

$$\mathcal{C}(\boldsymbol{x}) = S_i, \quad i \in \{1, \ldots, n\} \tag{4.6}$$

that maps an input vector to one of the known persons.

Because we present only genuine test sequences to the classifier, there is only one type of error the classifier can make in the closed-set classification task, namely a *misclassification*. The number of misclassifications for a test set serves as performance metric for the classifier. Normalizing by the number of test samples gives us the misclassification rate

$$MCR = \frac{\#misclassifications}{\#samples} = \frac{\left|\left\{\mathcal{C}(x_k^i) \neq S_i\right\}\right|}{\left|X_{\text{genuine}}\right|} \tag{4.7}$$

Open-Set Classification

For open-set classification the classifier operates under the assumption that not all persons in the test samples have been seen in the training set. The classifier first decides whether the individual in the test sequence is genuine or not. If he is, the classifier also tries to determine the individual's identity. The desired output class for an impostor is $S_{\text{unknown}} = S_{-1}$. Hence we can regard an open-set classifier as a function similar to the closed-set case for which the possible output classes are extended by the unknown class:

$$\mathcal{C}(\boldsymbol{x}) = S_i, \quad i \in \{1, \ldots, n\} \cup \{-1\} \tag{4.8}$$

In open-set classification three types of errors have to be distinguished:

- A *false rejection* occurs for a genuine test sequence if it is classified as impostor sequence:
$$\mathcal{C}\left(x_k^i\right) = S_{-1} \quad i \in \{1, \ldots, n\}$$
- A *false acceptance* occurs for an impostor test sequence if it is classified as genuine test sequence:
$$\mathcal{C}\left(x_k^{-1}\right) = S_i \quad i \in \{1, \ldots, n\}$$
- And thirdly, a *misclassification* denotes the case, where although a genuine test sequence is rightly accepted as genuine, the assigned person identity is wrong:
$$\mathcal{C}\left(x_k^i\right) = S_j, \quad i \neq j, \quad i, j \in \{1, \ldots, n\}$$

We can derive the three corresponding error rates *false rejection rate (FRR)*, *false acceptance rate (FAR)* and *misclassification rate (MCR)*, which serve as performance metrics for the open-set classification task, by normalizing by the number of respective samples:

$$FRR = \frac{\#false\ rejections}{\#genuine\ samples} = \frac{\left|\left\{\mathcal{C}\left(x_k^i\right) = S_{-1}\right\}\right|}{\left|X_{\text{genuine}}\right|} \tag{4.9}$$

$$FAR = \frac{\#false\ acceptances}{\#impostor\ samples} = \frac{\left|\left\{\mathcal{C}\left(x_k^{-1}\right) = S_i\right\}\right|}{\left|X_{\text{impostor}}\right|} \tag{4.10}$$

$$MCR = \frac{\#false\ classifications}{\#genuine\ samples} = \frac{\left|\left\{\mathcal{C}\left(x_k^i\right) = S_j\ :\ i \neq j\right\}\right|}{\left|X_{\text{genuine}}\right|} \tag{4.11}$$

Note that from these equations follows that

$$CCR = 1 - MCR - FRR \tag{4.12}$$

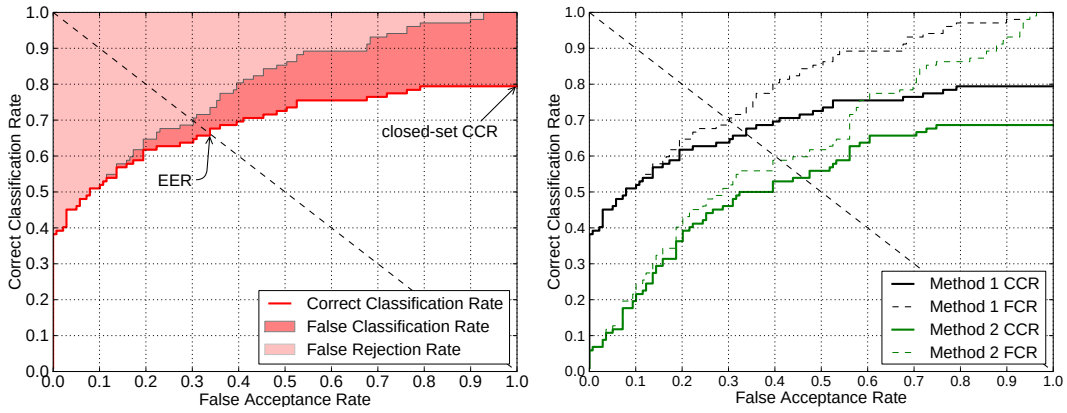
where CCR is the *correct classification rate*

$$CCR = \frac{\#correct\ classifications}{\#genuine\ samples} = \frac{\left|\left\{\mathcal{C}\left(x_k^i\right) = S_i\ :\ x_k^i \in X_{\text{genuine}}\right\}\right|}{\left|X_{\text{genuine}}\right|} \tag{4.13}$$

Equation 4.12 will help understanding the visualization plots of the receiver operation characteristics in section 4.2.2.

Open-set classification as Multi-Class Verification Problem

Verification refers to the task of determining whether the identity of a person is the one he claims it to be. A common approach to verification usually takes two steps: First, a score $s(\boldsymbol{x})$ is computed from the test sequence $\boldsymbol{x}$ for the model corresponding to the claimed identity. Second, the verification decision is made based on the relation of s to a threshold θ:



(a) (b)

Figure 4.7: Example ROC curves for open-set classification performance visualization. (a) The correct classification rate is plotted as bold red line. The dark red area over the CCR denotes the false classification rate, whereas the light red area denotes the false rejection rate. (b) Exemplary plot for a comparison of two methods. Here the CCR is plotted as bold line and the respective FCR as dashed line.

$$\mathcal{V}_i(\boldsymbol{x}) = \begin{cases} 1 & s \geq \theta \\ 0 & s < \theta \end{cases} \tag{4.14}$$

This can be straightforwardly generalized to multi-class verification in two ways. Either we define a global threshold θ_g that is the same for all classes, or one separate threshold θ_i for each class. In this thesis I use a global threshold.

Receiver Operating Characteristic

The performance measures defined in equations 4.9-4.11 are dependent on the threshold parameter θ. More specifically, by choosing θ one can trade off different kind of errors, as it is not possible to minimize all three of them at once. However, it usually does not make sense to report performance measure for a single threshold only, because there is no reasonable justification for any specific threshold, and also the system's overall characteristics cannot be deduced from a single operating point only.

The *Receiver Operating Characteristic* or ROC curve, a tool originally used in signal detection theory, visualizes the system's overall performance. Although usually employed for assessing the performance of binary classification and single-class verification tasks, we can generalize it to the open-set classification task.

The misclassification, false rejection and false acceptance rates are functions of the threshold parameter θ:

$$MCR = MCR(\theta) \tag{4.15}$$
$$FRR = FRR(\theta) \tag{4.16}$$
$$FAR = FAR(\theta) \tag{4.17}$$

The three functions describe a three-dimensional parametric curve. By elimination of θ we can consolidate them into two functions depending on the false acceptance rate. They are essentially two dimensional projections of the three-dimensional curve.

$$MCR = MCR(FAR) \tag{4.18}$$
$$FRR = FRR(FAR) \tag{4.19}$$

A cumulative plot of these two functions over the correct classification rate results in a typical open-set ROC curve such as in figure 4.7. Recall from equation 4.12 that at any given operating point, i.e. any FAR, the correct classification rate, the misclassification rate and the false rejection rate always sum up to 1.

Note that we can derive directly the performance of a classifier for the related closed-set recognition task from the open-set ROC curve. The related closed-set task results from considering only the genuine persons in the test set. The correct classification rate for the closed-set task is the CCR at $FAR = 1$ (see figure 4.7)[3].

Equal Error Rate

Although any attempt to reduce the ROC curve to a single number loses information, it is sometimes desirable to report single values for a specific operating point. A common choice is the *equal error rate* or *EER*, defined as the error at the point, where the false acceptance rate is equal to the sum of false rejection rate and misclassification rate. To obtain it, the function

$$FRR(\theta) + MCR(\theta) = FAR(\theta) \tag{4.20}$$

has to be solved. In the evaluation plots the equal error line is plotted as dashed black line (see for example figure 4.7). The EER is at the intersection of the equal error line with the CCR.

Discussion

A perfect classification system would achieve a CCR of 1 and FRR of 0 for any FAR. In practice, the CCR increases and the FRR decreases with increasing FAR. If the CCR rises steeply for small FAR, the classifier usually works well and can correctly classify a large number of genuine persons with only low acceptance of impostors. On the other hand,

[3] more precisely at $\theta = 0$, but in practice the difference is usually marginal

if the CCR is only slowly rising, with a high FRR over a large range of FAR, the classifier cannot distinguish well genuine persons from impostors. One reason can for example be, that for many genuine persons there is at least one visually alike impostor according to the extracted features, e.g. both wear a red shirt.

4.2.3 Results

In this section the results of the experiments are presented. I will begin by describing how I partitioned the shopping center data set into training and test data and investigate the influence of the training set size on the recognition performance.

In a second set of experiments, the performance of the features from chapter 3 is evaluated. First, the performance of the original SURF descriptor is compared with the performance of watershed regions. Then, both SURF descriptor and watershed regions are evaluated in more detail. The two color extensions for SURF are compared with the original SURF descriptor.

An *RGB color histogram model* is used as baseline. It is calculated directly from the bounding boxes of the persons' tracks. For each frame in a sequence, one color histogram is calculated. In the recognition stage, the test histogram is compared to all histograms from the training stage in a nearest-neighbour manner. This is similar to the basic recognition procedure employed for both SURF and watershed regions.

Finally, the influence of the normalization method for temporal fusion is investigated and some feature fusion results are presented, resulting in the best performing classifier in this evaluation.

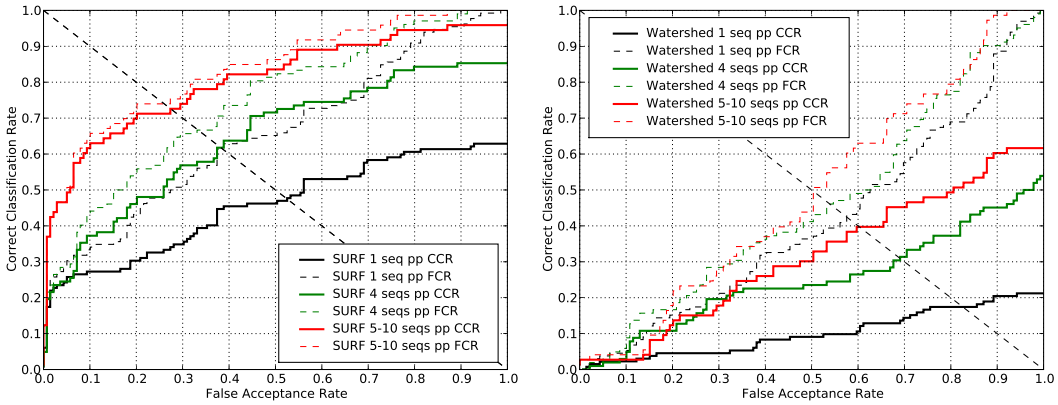
Training Set

The data set was partitioned in training and test data as follows. First, I selected 10 persons as 'known' persons. The selection criterion was that there were more than 6 image sequences for each of the persons, which was only true for the selected 10 (cf. figure 4.5a). Second, I chose randomly three training sets for each of the persons: a small, a medium and a large one.

- The small training set consists of only one sequence for each of the known persons. Although chosen randomly, I made sure that all persons are shown from the front, since this is by far the dominant in the data set and I did not want to penalize persons for which the training sequence showed a side or back view.

- The medium training set consists of four sequences for each of the known persons. Each of the four sequences shows the person from a different side: front, back, left and right. If one side was not available, another front view was selected instead.

- The large training set consists of 5 to 10 sequences per person. The number of sequences used for a person depended on their total number of sequences in the data set. For example, for one person there are only 7 sequences. As I wanted to use more training data than for the medium data set, but also retain more than one test sequence,



(a) (b)

Figure 4.8: Influence of training set size. (a) SURF. (b) Watershed regions. For both SURF and watershed regions an increase of the number of sequences in the training set increases the performance of the recognition. (seqs pp $\widehat{=}$ sequences per person)

I selected 5 sequences for this person. For other persons I selected around 50% of their sequences for training, but 10 at maximum.

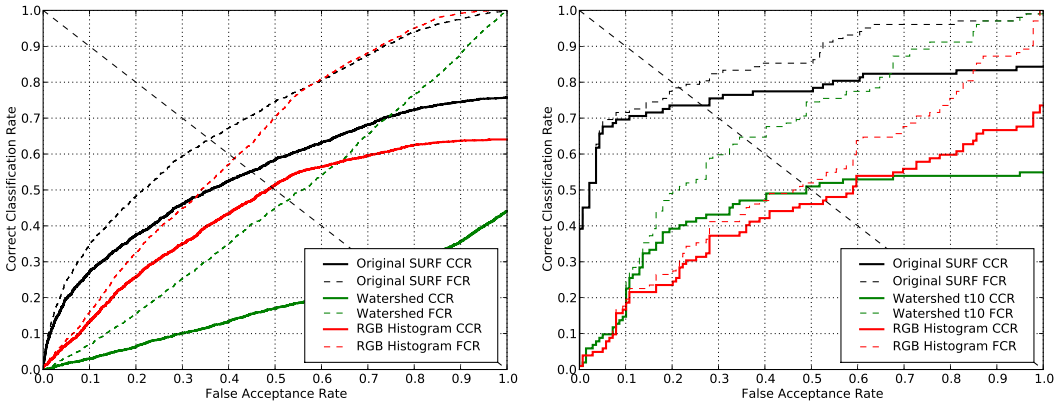
The influence of the training set size can be seen in figure 4.8. The plots show the results for SURF and watershed regions, respectively. Increasing the training set size increases recognition performance in three ways. The correct classification rate increases, while the false classification rate and the false rejection rate are reduced. The increased number of features extracted from a larger training set clearly helps the recognition.

In the remainder of this chapter, all experiments were conducted with the medium training set unless otherwise noted.

Features

Both watershed regions and RGB histogram are described with 8 bins per color channel, i.e. $8^3 = 512$ feature entries in total. For SURF the original descriptor with a threshold $t = 20000$ is used. The performance of SURF, watershed regions and RGB histograms is depicted in figure 4.9. On frame-level classification SURF performs best while the watershed regions perform worse than RGB histograms. RGB histograms only show a slight performance degradation compared to SURF. However, sequence level fusion (normed minmax-normalization) boosts both SURF and watershed, while the RGB histograms' performance is almost unaffected.

The performance of watershed regions is disappointing, although this is somewhat mitigated after temporal fusion. However, the underperformance on frame level actually indicates that the segmentation procedure works rather well. It seems to effectively be able to



(a) (b)

Figure 4.9: Comparison of features for person recognition. (a) Independent classification of each frame. (b) Fusion of frame scores for sequence-level classification. Although the RGB histogram outperforms the watershed regions on a frame-by-frame basis, it cannot benefit from temporal fusion. SURF outperforms both.

partition the image into regions of alike colors. In contrast to full image histograms, a well segmented e.g. black region then does not differ much from other black regions from other persons' frames, which explains the loss of some of the discriminative power compared to simple histograms. Such well segmented regions are a good basis for a model that also takes into consideration the relative positioning of the regions. This should be able to outperform histograms, as they are not capable of capturing such information about the spatial distribution of the color regions. How to efficiently incorporate such information into a model will be an important part of future research.

Color SURF Descriptors

We already saw in section 4.1.3 that the two color extensions to the SURF descriptor can improve the performance of the descriptor. While that evaluation was carried out on high resolution still images with several hundreds of features, the shopping data set presents the other extreme. The average size of a frame is quite small (see figure 4.5) and the interest point detector only selects $5 - 30$ features per frame.

Nevertheless, also on this challenging data set the color SURF descriptors provide a performance improvement. We can see in figure 4.10 that the Channel-Difference descriptor outperforms both original and Multi-Channel descriptors. The performance increase of the Channel-Difference descriptor can again be explained by the presence of rather large homogeneous regions of color. There are only small amounts of intra-region color structure present so that the Multi-Channel descriptor cannot benefit from the additional color information

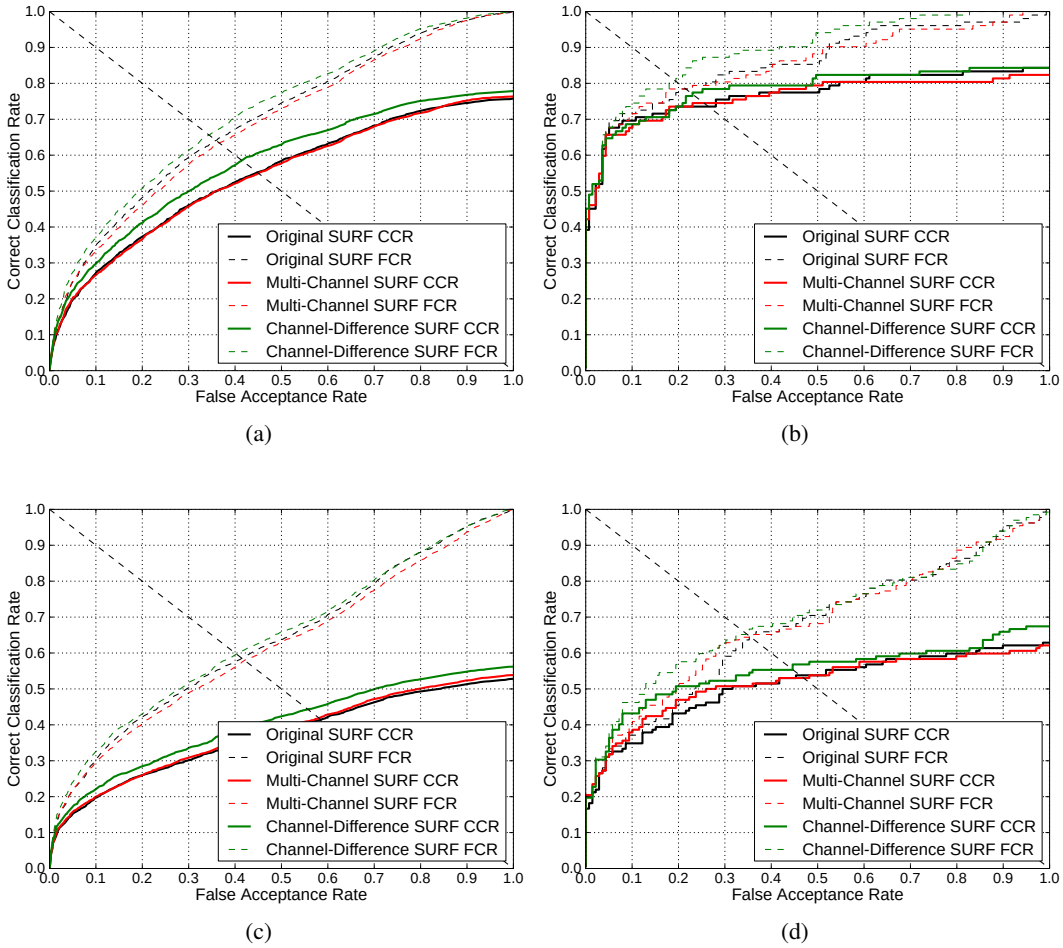


Figure 4.10: Comparison of different SURF descriptors. (a) Single frame classification. (b) Normed minmax normalized temporal fusion. (c),(d) Results from the small data set. The Channel-Difference descriptor provides a slight performance improvement over both original and Multi-Channel descriptors. When using only the small training set (bottom row) the relative performance improvement of the Channel-Difference descriptor after temporal fusion is larger.

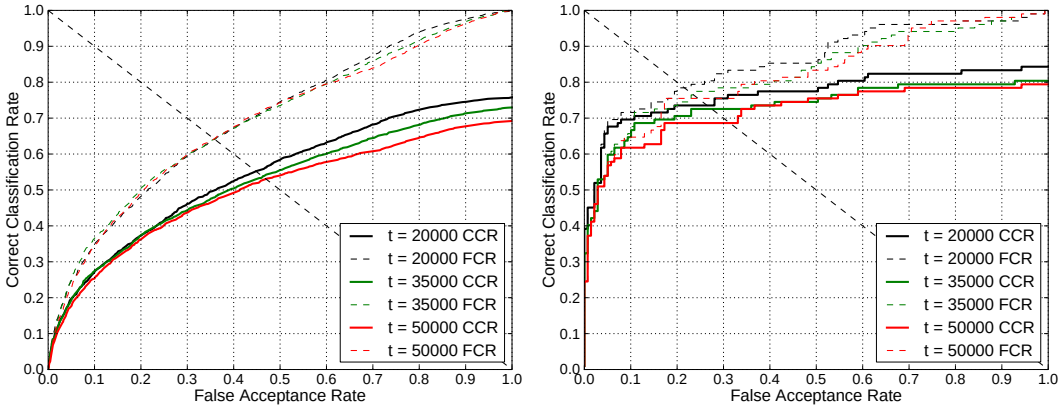


Figure 4.11: Influence of the SURF detector threshold. (a) Single frame classification (b) After normed minmax-normalized fusion. Lowering the detector threshold increases the recognition performance

compared to the original SURF descriptor. These findings are in agreement with the ones from section 4.1.3.

Fast-Hessian Threshold

SURF's fast-hessian detector takes a threshold as parameter. It denotes the minimum strength that an interest point must have to be selected. In this experiment, the influence of the threshold parameter is investigated. The experiments are performed with thresholds $t_1 = 20000$, $t_2 = 35000$ and $t_3 = 50000$.

Figure 4.11 shows that a lower threshold increases the recognition performance. The most likely reason for the performance increase is the larger quantity of features that result from a lower threshold. The lower the threshold is, the more interest points are selected and the better the image is covered. This result is consistent with other findings, reporting the absolute number of features has an impact on recognition performance (e.g. [52]).

Adding Border

In the original data set the bounding boxes around the persons are labeled tightly. At each side, the bounding box touches at least one part of the person's body (see figure 4.13a). It was reported that e.g. for the HOG detector it helped to leave a border around the person's body [53]. Although the task and approach in this thesis are quite different from Dalal's and Triggs' HOG, there are also similarities. The SURF descriptor also captures local intensity differences, i.e. gradients. This motivates this experiment. However, as can be seen in figure 4.13, the usage of a larger instead of a tight bounding box is not helpful. For the RGB

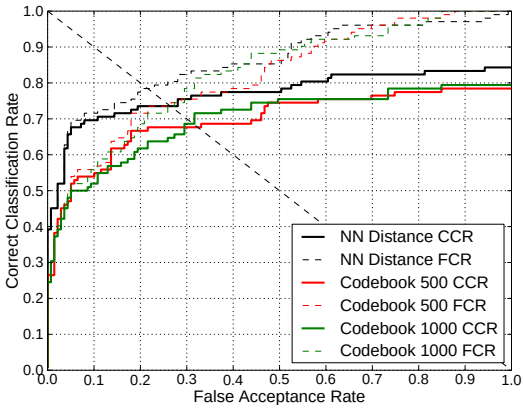


Figure 4.12: NN feature matching for each person vs. codebook based histograms of visual words. The common codebook looses some of the individuals' features' discriminative power.

histogram, the result can be easily understood as more irrelevant background is captured with a larger bounding box. For SURF, it indicates that a tighter bounding box keeps the features well 'inside' the person's body. A large bounding box permits feature regions to cover large parts of the person's body at once. This causes the classifier to rather learn the shape of a person in general than the specifics of the individual.

Watershed Markers

For watershed regions, one important parameter is the number of markers from which the segmentation starts. A higher number of markers promises a higher segmentation accuracy. If carefully implemented, the number of markers has only little influence on the runtime of the segmentation. However, the following clustering of similar regions is dependent on the number of initial regions. Hence a too fine grid should be avoided, too.

For this experiment the segmentation procedure was initialized with equally spaced markers with distances of 5, 10 and 15 pixels, corresponding to around 10% to 30% of the average frame width. Figure 4.14a indicates that the number of initial markers indeed influences the segmentation accuracy. However, the impact on the recognition performance is minimal, practically none after temporal fusion. A possible explanation is that only few frames actually can benefit from the increased number of markers, it can however not solve the inherent problem of the watershed regions as discussed above. I suspect that the increased accuracy would pay off more if spatial relations between regions were considered.

Clustering

In this section, the influence of clustering the features during training is investigated. Recognition performance using kmeans-clustered and agglomeratively clustered training data is

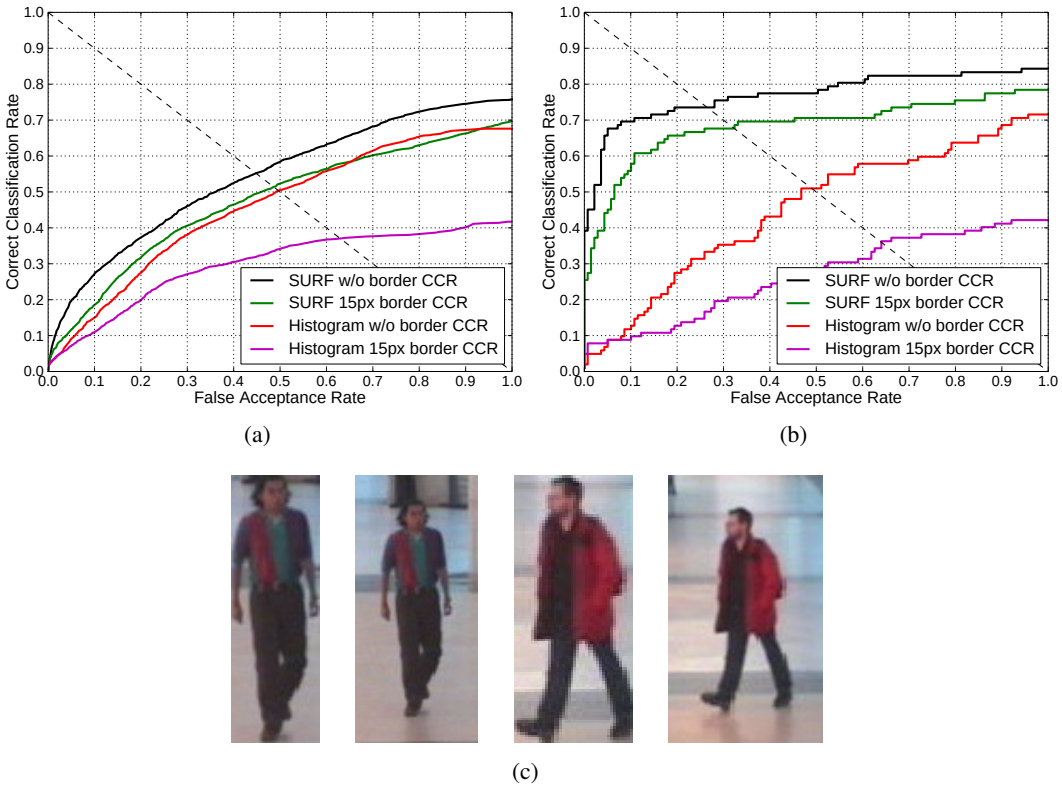


Figure 4.13: Increasing the size of the bounding box. (a) Single frame recognition performance (b) After normed minmax-normalized fusion. (c) Example images from the shopping data set with and without an additional border of 15 pixels. For both SURF and RGB histograms it is beneficial to use a tight bounding box.

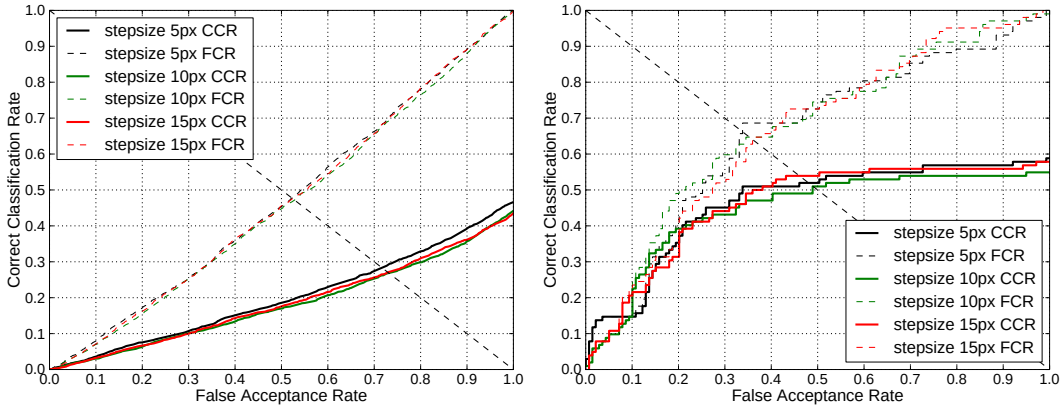


Figure 4.14: Evaluation of watershed marker stepsize. (a) Single frame classification. (b) After normed minmax-normalized temporal fusion. For single frames the larger number of initial watershed markers shows a minimal performance improvement.

compared with nearest neighbor matching on all training features. The evaluation is performed using the original SURF descriptor with $t = 20000$. As can be seen in figure 4.15 both agglomerative and kmeans clustering show similar recognition performance compared to using all training features. However, the number of features is significantly reduced, resulting in a speed-up at the recognition stage.

Agglomerative clustering leads to an uneven distribution of cluster sizes (see fig. 4.15c). If only the 750 largest clusters are kept (to be comparable with a 750 kmeans) the performance at EER is improved from 70% to 75%. Especially the false classification rate is reduced, for the closed-set recognition task by almost 10%.

Recall that clustering can also be used to build a common visual codebook from all persons, which is a popular approach for object classification using local features (see 2.3). However, in the case of individual person recognition, using such a common codebook harms performance (see figure 4.12). The results are given for codebook sizes of 500 and 1000 features. The performance loss can be explained by a loss of discriminative power of the features by clustering across individuals. The resulting cluster centers rather describe a human in general than one specific individual. What helps object classification approaches to generalize well to an object class, is something we would rather like to avoid.

Temporal Decision Fusion

In this section, the three methods of normalization before sum-rule temporal fusion are investigated: (i) no normalization, (ii) minmax-normalization of the n smallest distances and (iii) normalizing the n minmax-normalized scores, so that they sum up to 1 (*normed-minmax*) (cf.

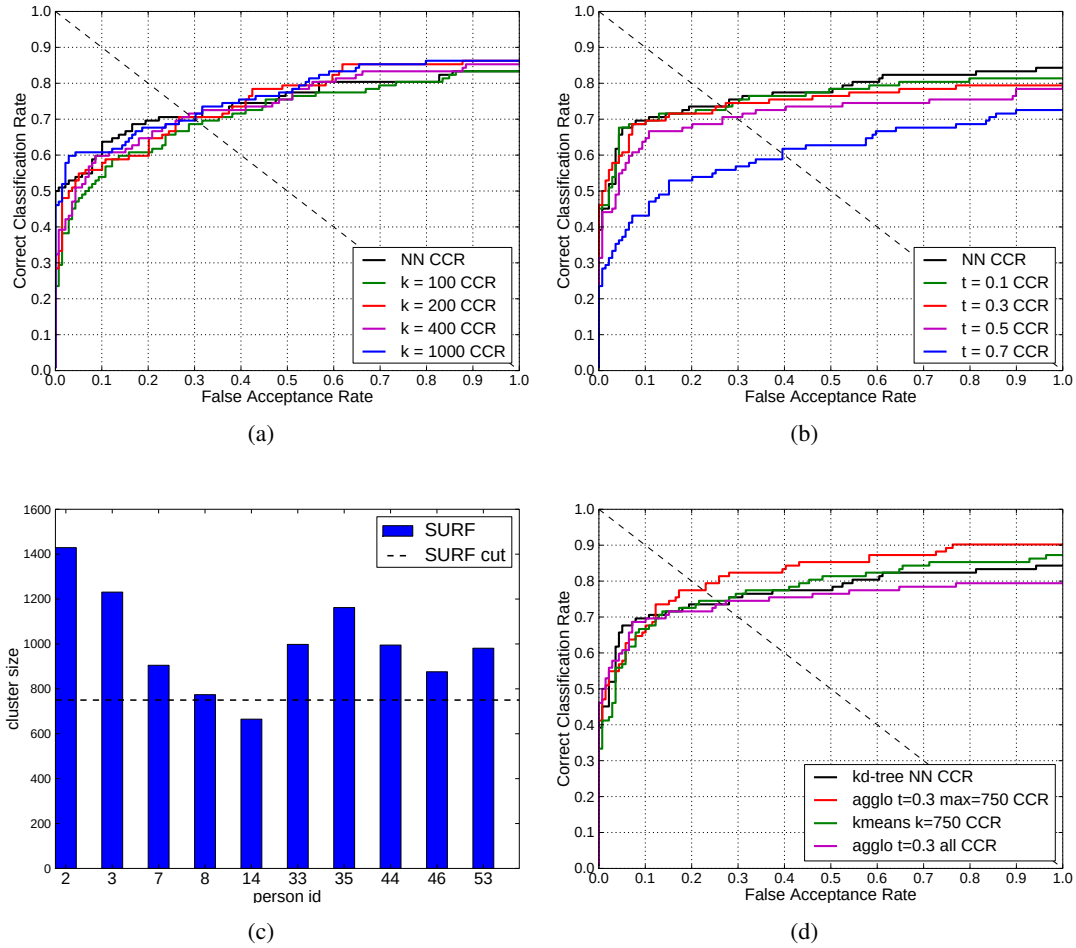
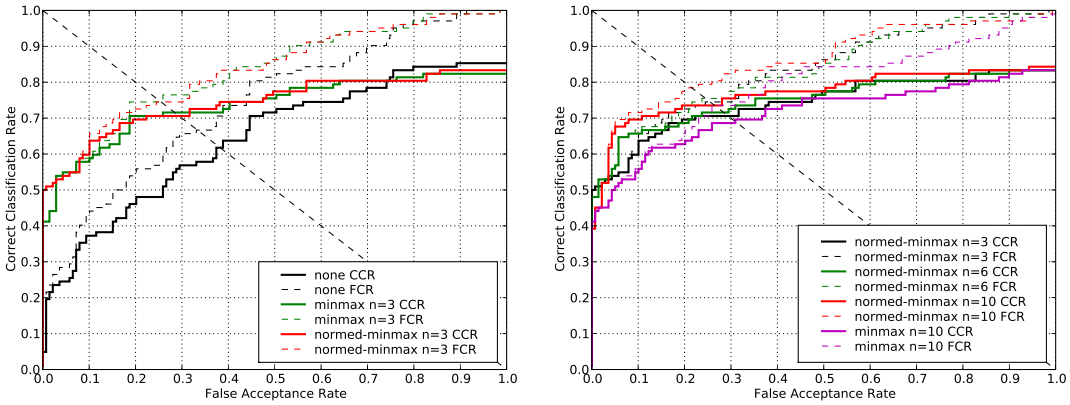


Figure 4.15: Clustering training features. (a) Kmeans clustering with different numbers of clusters k. (b) Agglomerative clustering with different merging thresholds t. (c) Number of clusters after agglomerative clustering. (d) Pruning small clusters. The recognition performance using clustered training features is comparable to using all training features. If small clusters are pruned, the performance can be considerably increased. The FCR has been omitted from the plots for reasons of clarity.



(a) (b)

Figure 4.16: Evaluation of score normalization for temporal fusion. (a) Minmax and normed-minmax perform equally for $n = 3$. (b) Including more than just the three smallest distances increases performance slightly. The best performance is obtained when normalizing using all available distances ($n = 10$) with normed-minmax.

section 3.3.3). Figure 4.16a depicts the performance increases gained from score normalization.

Clearly, sequence level fusion increases recognition performance over single frame classification. By classifying at the sequence level, 'bad' frames can be compensated for by a majority of correctly classified frames. This observation is consistent with all experiments conducted in this thesis, with the sole exception of RGB histograms (see figure 4.9).

Normalization of distances before the fusion increases the performance further. It effectively gives higher weight to frames for which the classifier is more confident that it chose the correct person. That is, the normalization increases the relative difference between the highest-ranked distance and the second- and third-ranked (etc.) distances, depending on the original difference.

In figure 4.16b the influence of the parameter n is investigated. n determines how many smallest distances are used in the normalization, e.g. for $n = 3$ only the three smallest distances per frame are minmax-normalized, the rest of the resulting scores is set to 0. It turns out that it is most beneficial to *not* discard any distances, but to take the distances from all 10 known persons into consideration. This indicates, that even when the correct person is only ranked fourth, fifth or even 9th[4] for one frame, the frame level score can still contribute to a correct sequence level classification. Furthermore, all distances are involved in the 'confidence estimation'.

[4]The 10th score never contributes to the sequence level score, since the 10th score is always 0 due to minmax-normalization.

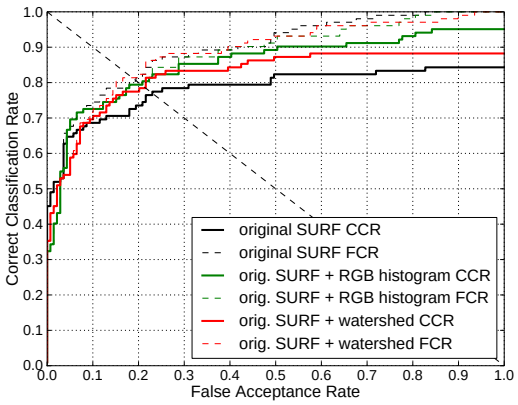


Figure 4.17: Fusion of feature types. The fusion of the original SURF descriptor with either watershed regions or RGB histograms shows further performance improvement.

Fusion of Feature Types

Finally, SURF features are combined with watershed regions and RGB histograms. The feature fusion is performed as described in section 3.3.4 with weights $w_1 = 0.66$ for the SURF features and $w_2 = 0.33$ for the respective other feature to account for the individual performance difference.

The results in figure 4.17 show that local features can be complemented successfully with other features. Fusion with either watershed regions or RGB histograms provides a comparable performance increase. The performance at EER of around 79% is even better than for the Channel-Difference descriptor (77%, not shown in the plot). The fusion with RGB histograms especially improves the performance for the related closed-set task. This is most likely due to an already good distinguishabilty of the genuine persons using color (cf. figure 4.9).

Example Identifications

I conclude the experiments with some exemplary identifications and misclassifications from the combined classifier using both SURF features and watershed regions. Figures 4.18, 4.19 and 4.20 show the results of the classifier ranked by score. In the respective figures, the three leftmost pictures depict three frames from the sequence to be recognized. The pictures on the right represent the output of the classifier sorted by score. The leftmost person on the right achieved the highest score and is selected by the classifier. The ground truth is marked with a red frame. Note that the approach can deal with substantial pose and view point variations.

Figure 4.18: Examples for correct classifications. Although the input sequences exhibit considerable variations in illumination and pose, the approach selects the correct person.

Figure 4.19: Examples for impostor test sequences. Even for impostors the first-ranked results are visually alike.

Figure 4.20: Examples for false classifications. Both correct answers are ranked second. In the bottom row, the first and second ranked person are actually the same person, just dressed differently, which again illustrates the challenging nature of this data set.

5 Conclusion

In this work, a non-biometric person recognition approach based on local features was presented. A bag-of-features model was adapted to the problem of instance recognition by building one model per individual and including distances between features as confidence measure. Two variations of the SURF descriptor for color images were proposed, the Multi-Channel and the Channel-Difference descriptors. Both were separately evaluated on Mikolajczyk's local feature evaluation data set. In order to describe homogeneous color regions, where the SURF interest point detector only achieves low coverage, the watershed segmentation was extended by clustering similar regions based on a higher level distance metric than pixel differences. It sub-segments images into regions of similar color which can then be used to describe the person.

The developed person recognition approach was evaluated on a subset of the publicly available CAVIAR data set. The data set is especially demanding due to the person's unconstrained movements and the low resolution of the sequences. The combination of SURF descriptor and person-specific models showed promising performance on the evaluation data set. The watershed regions however exhibited only equal or even worse performance than a baseline recognition approach based on simple RGB color histograms. On the other hand, it was shown that fusing frame scores for a sequence level classification improves the performance of SURF and watershed regions significantly, while RGB histograms cannot benefit well from fusion. This indicates that the watershed features allowed for a good classification in some frames, compensating for many 'bad' frames. A promising advancement would be to include spatial information about the position of the watershed regions into the model. This would provide a significant advantage over histograms, since it effectively could distinguish between different clothing configurations of the same colors.

The best classifier based on only local features achieved 77% correct classification rate at EER. In combination with watershed regions, 80% correct classification rate at EER could be achieved.

It can be concluded, that local features provide a viable approach to non-biometric person recognition. Nevertheless, this work can only be one further step towards robust, efficient and reliable appearance-based person recognition. In the following I will briefly outline four possible future research directions.

5.1 Future Work

Most existing local features were designed and evaluated in object recognition scenarios or for matching of points between two images. It is unclear which properties of local features

are favorable for individual person recognition. Considering the results of the performed tests, color promises to be an effective addition. It has so far been omitted from most state-of-the-art local features.

The bag-of-features model is a simplification of the real world as one way to deal with non-rigid body movements, occlusions and clutter. However, all information embedded in the spatial configuration of features is lost. Especially watershed regions would benefit from a model that incorporates spatial information since it would resolve many ambiguities that are responsible for the current poor performance.

Although I experimented with different training set sizes, it was not a goal of this thesis to specifically investigate methods to deal with a small training set. Nevertheless, this is an important issue for person recognition in several applications such as the ones discussed in section 1.1.1. For example, in surveillance applications it would be helpful if a human operator only had to manually label one frame (a few frames at most) instead of several hundred to search for a person in other video sequences. One possible research question would be, how a generic person model can be learned in advance, which then serves as basis for the individual models.

The confidence of a recognition can be improved if a person can be seen from several different cameras. Distributed camera networks become more and more a reality due to cheaper setup costs and an increased desire for security. On the one hand, temporal reasoning and information about the positions of the cameras could be used to prune possible correspondences between persons. On the other hand, a robust person re-identification approach can improve tracking persons within such camera networks. This is one example for an interesting research area where the shopping center data set from this thesis is insufficient.

The subset of the CAVIAR data set was prepared to serve as foundation for both the experiments in this thesis as well as further person recognition research. However, it lacks several desirable properties, which inhibits investigating some interesting research problems. There is for example just one camera view, which results in an improper balance of person view points (most views are frontal). Furthermore, the number of persons who are shown in more than three or four sequences is limited. A standard data set, especially designed for the purpose of advancing appearance-based person recognition, as is common for example for face recognition research, would certainly help to further the field.

Acknowledgements

I would like to express my gratitude to all those who made this work possible.

First of all, I would like to thank Prof. Rainer Stiefelhagen for giving me the opportunity to write my thesis in the interesting area of computer vision and become part of his team. I am very grateful to Keni Bernardin for suggesting this topic and his supervision with all his experience. Thanks go also to Dr. Jie Yang who supervised me during the first three months of my thesis while I stayed at Carnegie Mellon University.

I would also like to thank all members of the CVHCI group for the pleasant and inspiring working atmosphere. Thank you for letting me become part of the group and always safely guarding my laptop during lunch hours.

I want to thank Fabian, Franziska, Fred, Freya, JP, Kristie, Sabine, Anthony, Lisa, Nils, Alex and Patrick for a wonderful time in Pittsburgh. I very much enjoyed hanging out with you guys after long hours in the lab. I want to thank also Margit Rödder, Anja Knauer and InterACT who have made my stay at CMU possible in the first place.

Special thanks go to Konrad Miller for our weekly meetings, where we discussed the progress and problems of our respective theses. This was always something to look forward to throughout the week.

Thanks go also to Keni Bernardin, Ben Fuchs, Konrad Miller and Rainer Stiefelhagen for proofreading my draft and providing helpful suggestions for final improvements.

I wish to express my deepest gratitude to my parents for their continuous support and encouragements throughout my life. This work is dedicated to them. Finally, I want to thank my best friend and love Julia for her love and support.

Bibliography

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.

[2] Jeffrey E. Boyd and James J. Little. Biometric Gait Recognition. *Advanced Studies in Biometrics*, pages 19–42, 2005.

[3] A. C. Gallagher and Tsuhan Chen. Clothing cosegmentation for recognizing people. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[4] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *Proceedings of the British Machine Vision Conference*, volume 2, 2006.

[5] Gaël Jaffré and Philippe Joly. Improvement of a person labelling method using extracted knowledge on costume. *Computer Analysis of Images and Patterns*, pages 489–497, 2005.

[6] J. Sivic, C.L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *Proceedings of the British Machine Vision Conference*, 2006.

[7] L. Goldmann, M. Karaman, J.T.S. Minquez, and T. Sikora. Appearance-based person recognition for surveillance applications. Technical report, Technical University of Berlin, Communication Systems Group, 2006.

[8] J. Annesley, J. Orwell, and J. P. Renno. Evaluation of MPEG7 color descriptors for visual surveillance retrieval. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 105–112, 2005.

[9] Niloofar Gheissari, Thomas B. Sebastian, and Richard Hartley. Person reidentification using spatiotemporal appearance. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1528–1535, 2006.

[10] O. Hamdoun, F. Moutarde, B. Stanciulescu, and B. Steux. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–6, 2008.

[11] CR Wren, A. Azarbayejani, T. Darrell, and AP Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[12] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 3–10, 2000.

[13] Yang Song and Thomas Leung. Context-aided human recognition-clustering. *Computer Vision - ECCV 2006*, pages 382–395, 2006.

[14] Dragomir Anguelov, Kuang-Chih Lee, Salih B. Gokturk, and Baris Sumengen. Contextual identity recognition in personal photo albums. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[15] B. Suh and B.B. Bederson. Semi-automatic image annotation using event and torso identification. *Human Computer Interaction Laboratory, University of Maryland, College Park, Maryland, USA*, 2004.

[16] Gaël Jaffré and Philippe Joly. Costume: A new feature for automatic video content indexing. In *Proceedings Recherche d'Information Assiste par Ordinateur*, pages 314–325, 2004.

[17] Jie Yang, Xiaojin Zhu, Ralph Gross, John Kominek, Yue Pan, and Alex Waibel. Multimodal people id for a multimedia meeting browser. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 159–168, New York, NY, USA, 1999. ACM.

[18] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. Full-body person recognition system. *Pattern Recognition*, 36(9):1997–2006, 2003.

[19] K. Yoon, D. Harwood, and L. Davis. Appearance-based person recognition using color/path-length profile. *Journal of Visual Communication and Image Representation*, 17(3):605–622, 2006.

[20] W. Zhang, T. Matsumoto, J. Liu, and B. Begole. An intelligent fitting room using multi-camera perception. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pages 60–69. ACM New York, NY, USA, 2008.

[21] Thanarat Horprasert, David Harwood, and Larry S. Davis. A robust background subtraction and shadow detection. In *In Proceedings of the Asian Conference on Computer Vision*, 2000.

[22] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

[23] M. Haehnel, D. Klunder, and K.F. Kraiss. Color and texture features for person recognition. In *IEEE International Joint Conference on Neural Networks*, volume 1, 2004.

[24] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[25] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005.

[26] Xiaogang Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[27] Agnés Borràs, Francesc Tous, Josep Llados, and Maria Vanrell. High-level clothes description based on colour-texture and structural features. *Pattern Recognition and Image Analysis*, pages 108–116, 2003.

[28] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, September 2004.

[29] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 17–32, 2004.

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

[31] Florica Mindru, Tinne Tuytelaars, Luc Van Gool, and Theo Moons. Moment invariants for recognition under changing viewpoint and illumination. *Comput. Vis. Image Underst.*, 94(1-3):3–27, 2004.

[32] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[33] H.P. Moravec. Visual mapping by a robot rover. In *Proc. of the 6th International Joint Conference on Artificial Intelligence*, pages 598–600, 1979.

[34] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50, 1988.

[35] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.

[36] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, November 1998.

[37] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004.

[38] A. C. Berg and J. Malik. Geometric blur for template matching. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 1:I–607–I–614 vol.1, 2001.

[39] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[40] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[41] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[42] Matthew Brown and David Lowe. Invariant features from interest point groups. In *British Machine Vision Conference*, pages 656–665, 2002.

[43] Joost van de Weijer and Cordelia Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, pages 334–348, 2006.

[44] P.-E. Forssen. Maximally stable colour regions for recognition and matching. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[45] G. Mori, Xiaofeng Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–326–II–333 Vol.2, 2004.

[46] F. Meyer. Color image segmentation. In *International Conference on Image Processing and its Applications*, pages 303–306, 1992.

[47] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477 vol.2, 2003.

[48] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, 1966.

[49] I. Gronau and S. Moran. Optimal implementations of upgma and other common clustering algorithms. *Information Processing Letters*, 104(6):205–210, 2007.

[50] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *Proceedings of the British Machine Vision Conference*, 2006.

[51] R. B. Fisher. Pets04 surveillance ground truth data set. In *Proc. Sixth IEEE Int. Work. on Performance Evaluation of Tracking and Surveillance (PETS04)*, pages 1–5, May 2004.

[52] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proceedings of the British Machine Vision Conference*, 2005.

[53] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893, 2005.