Karlsruhe Institute of Technology
Department of Computer Science
Institute for Anthropomatics
Prof. Dr. Rainer Stiefelhagen

Diploma Thesis

Adaptive Color Transformation for Person Re-identification in Camera Networks

Submitted by

Clemens Siebler

April 2010

Advisors

Prof. Dr.-Ing. Rainer Stiefelhagen
Dipl. Inf. Keni Bernardin

Computer Vision for Human-Computer Interaction Research Group
Institute for Anthropomatics
Karlsruhe Institute of Technology
Title: Adaptive Color Transformation for Person Re-identification in Camera Networks
Author: Clemens Siebler

Clemens Siebler
Reinhold-Frank-Strasse 21
76133 Karlsruhe
Email: clemens.siebler@googlemail.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, April 29, 2010

.....
(Clemens Siebler)

Abstract

The task of observing a person in one camera and finding that person again in another camera is often referred to as the *Person Re-Identification problem*. The problem has a significant importance in large security scenarios as for example airports or train stations. Re-identifying a suspicious person in another camera is often done manually by an operator, which is a tedious and error-prone task. Therefore, it is desirable to have computer-aided assistance. Due to its complexity, the problem of matching one person with given multiple targets is still being studied in recent research.
A large variety of techniques exist to describe the appearance of a person. The face and the gait are popular descriptors in recognition tasks, but they tend to produce less accurate results when employed in combination with low resolution cameras. However, those cameras are widely used in real world scenarios, mostly because of economical reasons. Often, color features are used for matching, because of their ability to deal with deformation of pose and viewpoint changes. However, on the other hand, color features are very sensitive to different illumination conditions, which most times exist at different camera sites. As a consequence, automated systems tend to produce erroneous results. The problem has been addressed in research by calculating a transfer function between camera pairs in order to map colors between the different views. Most often, a fixed training stage with a constant illumination condition is used. As long as lighting stays constant, these functions allow for great improvements in accuracy. However, in real world scenarios, illumination changes over time. Therefore, the calculated transfer function does not provide a correct mapping any more. As a consequence, recognition performance decreases and wrong matching occurs more often. This challenging problem has only been investigated by few and due to its great impact on the matching process, it needs further study.
In this thesis a novel approach is presented which is able to deal with changing illumination over time. Exhaustive experiments were carried out on the PETS2007 data set. The footage of this multi-camera scenario was taken at an airport and therefore includes the common, real world problem of illumination changes over time in crowded scenes. Matching between cameras was done with between 81 to 101 persons. The proposed method shows an improvement of approx. 12% in recognition accuracy for Rank 1 over just using the Cumulative Brightness Transfer Function (36% vs. 48% recognition accuracy). For Rank 10, an improvement of approx. 20% is measured (64% vs. 84% recognition accuracy).

Kurzzusammenfassung

Das Wiederfinden einer Person in einem Kameranetzwerk ist in der englischsprachigen Literatur als das *"Person Re-Identification problem"* bekannt. Speziell in Überwachungsszenarien, wie z.B. Flughäfen oder Bahnhöfen, ist dieses Problem von großer Bedeutung. Die Wiedererkennung einer verdächtigen Person wird in solchen Szenarien oft von einem Menschen manuell durchgeführt. Ein solches Vorgehen ist aber äußerst ermüdend und deshalb auch sehr fehleranfällig, weshalb eine rechnerbasierte Unterstützung wünschenswert ist. Bedingt durch seine hohe Komplexität ist das Problem der Wiedererkennung von Personen in Kameranetzwerken immer noch ein aktuelles Thema der Forschung.
In der Praxis gibt es eine große Vielzahl an Möglichkeiten eine Person mathematisch zu beschreiben. So werden in Wiedererkennungsaufgaben oft das Gesicht oder der Gang als charakteristische Personenmerkmale gewählt. In Überwachungsszenarien werden jedoch, meist aus ökonomischen Gründen, Kameras mit niedriger Auflösung verwendet. Dies hat zur Folge, dass speziell das Merkmal Gesicht nur noch eingeschränkt als Wiedererkennungskriterium verwendet werden kann, da niedrige Auflösungen und die oft schlechte Bildqualität die Erkennungsrate sehr stark negativ beeinflussen. Deshalb werden in der Praxis oft Farben der Kleidung als Wiedererkennungsmerkmal verwendet. Diese bringen zusätzlich den Vorteil mit sich, dass sie robust gegenüber Veränderungen der Pose und des Ansichtswinkels der Person sind. Andererseits sind Farbmerkmale sehr anfällig für unterschiedliche Beleuchtungbedingungen. In Überwachungsszenarien weisen die durch die Kameras aufgenommenen Szenen oft sehr unterschiedliche Beleuchtungen auf, was primär dadurch bedingt ist, dass die Kameras an unterschiedlichen Orten angebracht sind. Die Wiedererkennung anhand von Farbmerkmalen wird dadurch negativ beeinflusst. Es wurden deshalb Ansätze vorgestellt, die eine Übertragungsfunktion (engl. Brightness Transfer Function) für Farben zwischen den Kamerapaaren berechnen. Dadurch erhalten die Farben ein konstanteres Erscheinungsbild, wodurch wiederum ein robusteres Wiedererkennen möglich ist. Jedoch werden diese Funktionen meist in einer zeitlich fixen Trainingsphase berechnet, in der die Beleuchtungsbedingungen konstant sind. Die so erhaltene Funktion erzielt einen guten Farbtransfer unter diesen Bedingungen, jedoch verändert sich die Beleuchtung in der Praxis meistens über die Zeit. Die zuvor berechnete Funktion ist folglich nicht mehr gültig und erzeugt deshalb fehlerhafte Farbanpassungen. Die zeitliche Veränderung der Beleuchtungssituation wurde bisher nur sehr spärlich untersucht. Da es sich aber um einen sehr einflussreichen Faktor in der Wiedererkennung handelt, ist eine Kompensierung der Beleuchtungsveränderung weiter zu untersuchen.
In dieser Arbeit wird ein neuer Ansatz zur Kompensierung der zeitlichen Veränderung

der Beleuchtung vorgestellt und auf dem PETS2007 Datensatz evaluiert. Dieser praxisorientierte Datensatz wurde an einem Flughafen aufgenommen und zeichnet sich durch die Beleuchtung aus, welche sich über die Zeit hinweg stark verändert. Das Wiedererkennen wurde mit 81 bis 101 Personen durchgeführt. Die vorgestellte Methode verbessert den Ansatz der Cumulative Brightness Transfer Function im Mittel um ca. 12%, wenn eine perfekte Zuweisung stattfinden soll (von 36% auf 48% Erkennungsrate). Soll die gesuchte Person unter den ersten zehn Treffern liegen, dann erzielt das Verfahren im Durchschnitt eine Verbesserung von ca. 20% (von 64% auf 84% Erkennungsrate).

Contents

1 Introduction

In literature, the problem of observing and finding a specific person again in a camera network is referred to as *person re-identification* or *person recognition*. It is an important topic in computer vision, because there are many applications in real world scenarios that can profit from such an automated re-identification. Especially in large security scenarios as for example airports or train stations multi-camera systems are employed. There, re-identifying a suspicious person in another camera is often done manually by an operator, which is a tedious, error-prone and expensive task. Therefore, it is desirable to have computer-aided assistance. Matching one person with given multiple targets is still being studied in recent research.
Color features are often used to describe the appearance of a person, because they have the ability to deal with deformation of pose and viewpoint changes fairly well. On the other hand, using color features to re-identify persons can be difficult, because they are sensitive to the given lighting condition. It is very common that those conditions differ from camera to camera. Therefore, real world applications require that the different illumination conditions at different camera sites be compensated by automated algorithms.

1.1 Motivation

Person re-identification can be done in several ways. Using the face to recognize persons has been widely studied and can provide good matching performance. Another discriminative characteristic of human beings is for example their gait, which can also be used to identify a specific person. A big advantage of these features is that they are very likely to stay constant over a longer period of time. While the gait of a person may change slowly within the lifespan of a person, the face tends to change more quickly with aging. Furthermore, there are additional natural or artificial factors that change the appearance of the face such as beards or makeup. Other challenges might arise due to occlusion (e.g. glasses, hair or clothing) or varying facial expressions.
While these difficulties can decrease recognition performance, facial recognition can still produce good results when using data that satisfies a certain standard of quality, e.g. in terms of image resolution, image compression and illumination of the face. In surveillance scenarios, usually low-resolution cameras with a large field-of-view are used. For the operator of a system, it is of great interest to cover large areas while minimizing the cost for hardware and system operation. In addition to that, those scenarios are often characterized by having different viewpoints and different and

Figure 1.1: In the first row, which pair of faces show the same child? The task becomes much simpler when clothing information is included, as seen in the second row. (Source: [1])

constantly changing illumination conditions at each camera sites. Face detection approaches usually fail to detect faces in low resolution images, which is an essential step in automated face recognition. Even if detection works, results suffer due to the little amount of information that is contained in a small face.

In surveillance scenarios a person usually needs to be re-identified within a small time window, varying between a few minutes and a few hours. The clothing of a person is usually fixed within that time window. A simple example shows the complementary information of clothing in contrast to faces. It is shown in Figure 1.1 and was proposed by Gallagher et al. in [1]. The first row shows six faces that belong to three different children. It is quite difficult to tell which pairs shows the same child. The second row shows the same faces, but in a larger crop that includes parts of their clothing. The additional information provided by the clothing makes the task much simpler.

Since clothing is mostly characterized by its color and its texture, it is important to have a constant appearance of color among all cameras. Real world scenarios usually do not fulfill that requirement. Reasons are especially the different illumination conditions that exist at different camera sites. One site may be indoor and another one outdoor. In addition to that, artificial or ambient light might change or get mixed, resulting in different appearance of colors. Furthermore, cameras tend to produce different colors, even when using the same type of camera [2]. Since color feature are very sensitive to the mentioned problems, it is desirable to have an adjustment of colors for each frame the cameras provide. Such an adaption would make color features more robust and therefore more applicable for person re-identification tasks in real world scenarios.

1.1.1 Applications

Visual surveillance. Visual surveillance is a very important topic, especially in scenarios where a large amount of people occupies a rather small amount of space. Nowadays, airports, public places and train stations employ surveillance systems. Furthermore, they are used in shopping malls or grocery stores. The motivation for using such systems is very widespread: especially at airports and public places, it is of great interest to detect suspicious events and automatically track the involved person through the camera network. The main motivation may be minimizing harm for others. In shopping malls for example, the tracking can be used to keep track of shoplifters. Here the motivation may be minimizing financial loss. All these scenarios tend to exhibit very different lighting conditions at the different camera sites, mainly because of ambient light and a mixture of in- and outdoor settings. Here, a color adaption between cameras is necessary in order to provide robust matching performance.

Service-Oriented Robots. Research in service-oriented robots has the goal to build robots that can assist humans in everyday life. Such robots are usually equipped with stereo cameras, which give them the ability to see. But, it is still very difficult to give the robot the ability to keep track of the happenings in its environment. Further sensors such as distributed cameras can provide additional information to the robot, e.g. in a household scenario, where the robot is supposed to help the human. By using external cameras, the robot could keep track of a specific person more easily. Again, the different camera sites are highly influenced by the illumination and therefore produce a mismatch in color.

Camera Calibration. Camera calibration is often done by using a checkerboard pattern to compute the intrinsic parameters of a camera. On the other hand, persons can be used as calibration objects in order to compute both the intrinsic and extrinsic parameters. In order to do so, a matching of persons between the overlapping cameras has to be established. Because cameras usually look at the scenery from different angles, colors tend to look different. This is especially the case when ambient lighting illuminates parts of the scene. A color adaption would help to make the colors appear more constant in each camera, therefore making the matching more robust.

1.2 Contributions

Brightness Transfer Functions have already been used to improve recognition accuracy in multi-camera matching problems [3, 4, 5, 6]. Most of these methods employ a fixed training stage and assume that illumination stays constant over time at the camera sites. Therefore, accuracy decreases when illumination conditions change. Only a few authors have dealt with the problem of changing illumination over time [7, 8].

In this thesis a novel method for improving the accuracy of Brightness Transfer Functions is proposed. The effects of changing illumination over time are compensated by an iterative approach. Therefore, a fixed training stage for the Brightness Transfer Function can be used, because the intra-camera appearance of colors is rendered more constant. The proposed method does not require background segmentation or entry/exit regions such as [7]. While [8] can produce error-prone results when large illumination changes occur over time with no new observations made, the presented approach always gives constant results. Exhaustive experiments are carried out on the PETS2007 data set [9]. The footage of this multi-camera scenario was taken at an airport and therefore includes the common, real world problem of illumination changes over time and crowded scenes.

1.3 Thesis Outline

The remainder of this thesis is organized as follows:

In Chapter 2, related work on the person re-identification task is presented. The basic principles of automated multi-camera matching are described in Chapter 3 on an exemplary system. It makes use of a Brightness Transfer Function, whose objective is to map brightness values between different cameras in order to compensate for the illumination differences. Then, the here proposed extensions to the approach, which allow to handle changing illumination over time, are presented. This makes it possible to maintain the accuracy of the BTF, which usually decreases when the lighting condition under which it was trained changes. Extensive experimentation is carried out in Chapter 4. Many different situations are evaluated in order to gain new insights into the addressed problem of changing illumination. The obtained results are summarized in Chapter 5 where also further work is pointed out.

2 Related Work

In this chapter, related work to the multi-camera matching task is presented. Most focus is put on the multi-camera matching problem, person re-identification and color constancy. A real world application of a person re-identification system can also make use of person detection and segmentation algorithms. Therefore, some algorithms that fall into that category are briefly mentioned.

2.1 Multi-camera Matching

Javed et al. proposed the use of a Brightness Transfer Function (BTF) in [3]. Under the assumption that the given illumination at each camera site is constant and that objects appear planar, the BTF maps brightness values for each color channel between the cameras, similar to the approach in [10]. The BTF makes use of hand labeled observations of persons that appeared in a pair of cameras. Probabilistic PCA [11] was used to learn a low dimensional subspace, where the training Brightness Transfer Functions lie in. In [4] Javed et al. extended their approach to incorporate inter-camera space–time and appearance relationships. In order to learn entry and exit regions, transition times and velocities, they used kernel density estimation, which also results in an automatically learned camera network topology.
Prosser et al. extended the BTF in [5] to a Cumulative Brightness Transfer Function (CBTF). Given only a sparse training set, they claim that their transfer function makes better use of the color information. A bi-directional matching was used in order to avoid false positives. Furthermore, they compared it to a Mean Brightness Transfer Function which showed slightly worse results than their proposed CBTF. Additionally, the CBTF clearly outperformed the subspace method used in [3].
While these approaches did not take the temporal illumination change into account, Prosser et al. proposed an adaptive CBTF [7] that is able to compensate the changes that occur due to change in illumination over time. Their idea was to map the current color situation back to the situation where the CBTF was trained. The approach requires fixed entry/exit regions, a fixed background and a well working background segmentation.
Chen et al. also used a BTF to solve the problem of having changing illumination conditions at each camera site [8]. Instead of having a fixed training phase for the BTF, they updated their BTF over time by using the most recent observations of matched persons. Only matches that showed a small distance during matching were used for updates. While the adaptive learning process adapts to graduate changes in illumination, it can produce error-prone results when large illumination changes occur

over time with no new observations made at the same time.
D'Orazio et al. further investigated the differences between the MBTF and CBTF in [6]. They receive contrary results to Prosser et al. in [5] and state that both functions, MBTF and CBTF, provide comparable results.
Madden et al. proposed an online k-means clustering for fast calculation of the main colors of an object [12]. In order to compensate for the varying illumination conditions at each camera site, a controlled equalization of the cumulative histograms is used. The controlled equalization is computed locally for each object. Their approach, in a modified version, was outperformed by Prosser et al. in [5] by using the CBTF.
Columbo et al. investigated color constancy in [13] by normalizing the first- and second-order statistics of observed foreground data in each camera. Normalizing mean and covariance is similar to the gray world assumption [14] which assumes that the average scene chromaticity is a neutral gray. On the one hand, their results showed that the application of color constancy does not necessarily improve the performance of multi-camera matching. On the other hand, normalizing mean and covariance of observations may not be a sufficiently sophisticated technique for compensating the different illumination conditions at camera sites.
Gilbert et al. presented an unsupervised approach in [15] that tracks objects across spatially separated cameras. Their method uses an incremental learning method that models color variations and probability distributions of spatio-temporal links between cameras. As color transfer method, transformation matrices were used, similar to rotation matrices in RGB color space [2].

2.2 Person Re-identification

The multi-camera matching of persons focuses mainly on finding the same person again in a camera network. Given two observations of persons, the goal is to decide whether the two images show the same person or not. Many techniques of varying complexity have been proposed over the last years and only a few relevant publications should be mentioned, since the main objective of this work is the color adaption over time.
Hamdoun et al. showed in [16] that local feature based approaches using SIFT provide good results under relatively fixed viewpoints. Hence, generally speaking, the performance of SIFT usually drops when viewpoint changes occur. Wang et al. proposed a matching using line signatures [17] which can deal with viewpoint changes much better than SIFT, but their approach is only evaluated on rigid objects, e.g. buildings, etc. Due to the deformable properties of clothing and human pose, it is not clear how their approach would perform in the person re-identification task.
Gray et al. evaluated color and texture features in combination with boosting in [18]. Their approach was tested on a large database, including changing viewpoints and poses and produced good accuracies. In [19], Gray et al. evaluated different appearance models, especially different types of histograms for re-identifying persons. They

used the same database with changing viewpoints and received relatively robust results when using histograms. Different color spaces (e.g. RGB, HSV, YCbCr, etc.) performed very similarly. On the other hand, these techniques were not evaluated in combination with a color adaption approach.
In [20], Hu et al. noticed that the principal axis among different persons is often different. Therefore, it is important to model the spatial color distribution of a person in order to improve accuracy. Park et al. conclude similar results in [21].

2.3 Color Constancy

Color constancy aims to achieve an illumination invariant description of a scene taken under an illumination whose spectral characteristics are unknown. Without any given reference object in the image, it is an ill-posed problem. Nevertheless, color constancy is an important research topic. Especially in computer vision many applications rely on color features which are very sensitive to illumination variation.
In [22], Agarwal et al. give a good overview over established color constancy algorithms that have been proposed by researchers during the last decades. Many of these techniques were the initial idea for several of the cited approaches in Section 2.1.
Ilie et al. made use of a color calibration chart in order to ensure color constancy among multiple cameras [2]. Even given the same scene illumination for a pair of cameras, they will still produce different colors in the image data. Reasons for such a behavior are for example aperture variations, fabrication variations, electrical noise and interpolation artifacts.
Given the same scene or object observation under two different illuminations, Porikli made use of a correlation model function [23] in order to map brightness values for each color channel between both conditions. In [10], Grossberg et al. established a mapping between two images of the same scenery, taken with different exposure times. While this may be not directly related to color constancy or the multi-camera matching problem, their proposed transfer function is the basis for [3, 4, 5, 7, 8, 6], which try to compensate different illumination conditions at different camera sites.

2.4 Person Detection

The task of detecting persons in a still image or video has been studied for several years and is an important topic for human-computer interaction (HCI) and surveillance scenarios. It is a challenging problem due to variations in people's poses, lighting conditions and inter- and intra-person occlusions. Most approaches can be categorized by either trying to detect a human person by using one single detection window or trying to detect the parts of a human body separately, e.g. head/face, arms, legs, torso, etc. For the second category, the detection results are then fused to form a single hypothesis in a further step. While Gavrila et al. gave a good overview over the approaches up to the year 2000 [24], a large amount of new techniques has been

proposed during the last decade. By now, human detection is a very large field and is not the topic of this thesis. Therefore, only some remarkable publications are pointed out in this section.
Dalal et al. proposed using Histograms of Oriented Gradients (HOG) for human detection [25]. While obtaining a classifier with a simple architecture, their technique outperformed comparable approaches and became a baseline for many other approaches.
Given a typical surveillance scenario with a static camera, Grabner et al. simplified the complexity of detecting a person in an arbitrary environment by detecting persons only in a fixed scenery [26]. They employed a highly overlapping grid of classifiers in combination with Haar-like features. By using fixed update rules they tried to avoid drifting of the adaptive classifiers over time.
Schwartz et al. extracted HOG, texture and color features from pedestrians to form a high-dimensional feature vector [27]. The Partial Least Squares Analysis [28] was used to reduce dimensionality, resulting in a good separation between human and non-human training examples, even when reducing the dimensionality to as low as 20 latent variables. They also propose a similar, part-based approach that is able to deal with occlusions by detecting head, torso and legs separately [29].

2.5 Person Segmentation

A very important point is the segmentation of a person. Usually, person detectors return bounding boxes which contain the person and parts of the background. In most cases the background information is not needed and should therefore be discarded. Segmentation is usually done within the first steps of processing the given data and therefore often impacts the following steps in the processing chain. A good segmentation will allow the following steps to perform better, while a bad segmentation will distort results in the later steps.
A popular technique to segment persons from the background is using a background modeling algorithm. Stauffer et al. proposed modeling the background with a mixture of Gaussians [30]. While the approach may be outdated, it is still used by many researchers, due to its simplicity and freely available implementation in OpenCV [31]. An improved version was presented by Kaewtrakulpong et al. in [32]. A further speedup and a slightly better segmentation was achieved by Zivkovic in [33]. Instead of using a fixed number of Gaussians, the needed number of components per pixel was automatically selected.
Kim et al. proposed a background segmentation approach utilizing code books in [34]. The algorithm showed improvements especially in low contrast scenes. Their approach was further improved by Ilyas et al. in [35].
Rother et al. used a variation of graph-cuts called 'Grabcuts' to segment an object in its bounding box in [36]. Shi et al. proposed normalized cuts for image segmentation in [37]. Many similar methods that can produce accurate segmentation exist, but due

to their computational effort they may not be suitable for real-time application, which is required in real world scenarios.

3 Multi-camera Matching using the Iterative Brightness Transfer Function

An automated or computer-aided multi-camera system often employs several components in order to solve the re-identification problem. In this chapter an exemplary system is described. Without loss of generality, it is assumed that the system only consists of two distributed cameras. This case can be easily extended to multiple cameras.
In a first step, incoming frames from the cameras are usually processed with a background segmentation algorithm in order to separate the foreground (i.e. the persons) from the background. Afterwards, blob detection is used for obtaining bounding boxes. Within the last years person detection became more robust and fast. Therefore, blob detection can be combined with a person detector in order to minimize false positives and enhance overall detection robustness. While many different tracking techniques exist, only some are applicable due to real-time requirements. A commonly used technique is histogram back-projection [38] or a simple blob tracking. The person region obtained by the background segmentation may be very inaccurate, so it may be the object of a further post-processing step. When using a color transfer technique, such as the Brightness Transfer Function, the observations are now transformed in order to compensate for different illuminations at the camera sites. Person descriptors are built and compared with the already observed persons that were stored in a database. If a match is found, the observation can either be used to update the entry in the database or simply be discarded. If no match was made, a new entry is added to the database. The comparison between two descriptors can be additionally combined with spatio-temporal information (not discussed in this thesis). This makes sense in many scenarios, because it is often more likely to observe a person within a pre-learned time window than at completely random points in time.

3.1 Background Modeling and Person Segmentation

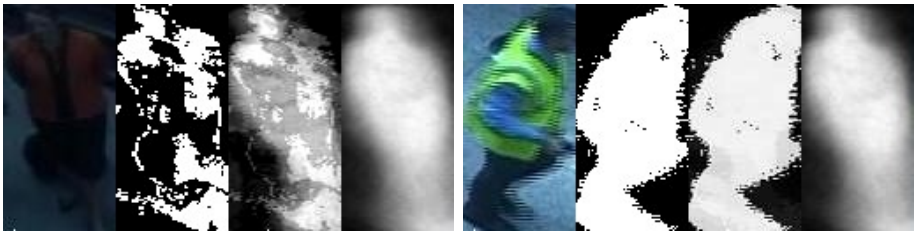
A system for person re-identification can make use of background modeling because of two reasons. Firstly, it is used for person detection by doing blob analysis. Secondly, it is employed to obtain masks of the observed persons. Background information often affects the person descriptors negatively and should therefore be discarded.
A very popular background segmentation approach was presented in [30], where each

background pixel is modeled by a mixture of Gaussians. The basic idea behind the approach is that a background pixel is most likely characterized by a small variance and a large weight of its corresponding Gaussian. Each new incoming pixel is matched against all other Gaussians and if it does not lie within a specific amount of the standard deviation of the distribution, the distribution with the smallest weight is replaced. Otherwise, the matching distribution is updated. Since the background can have a multi-modal distribution (due to e.g. flickering lights or moving leaves) a threshold is used to determine which of the Gaussians account for the foreground and which model the background.
Since many scenes tend to have a large amount of pixels that always show background, the total number of Gaussians can be dynamically adjusted [33] in order to save computational effort. For further details, the reader is referred to the references.

After obtaining a mask of the overall foreground, blobs can be detected [39] in order to obtain bounding boxes for the persons. Furthermore, a pre-trained person detector such as [25] can be used to get more robust results, especially when persons are appearing close to each other. In this case, blob detection tends to merge the observations and produce one big bounding box. Person detectors achieve a decent precision by using e.g. histograms of oriented gradients. When using such a detector, a detection window is moved over the whole image at different scales. In each window, a grid of histograms is calculated, where each cell is filled with the strongest gradient direction obtained by calculating the first derivative of the image. Those histograms usually capture the overall silhouette of humans fairly well, but can also be combined with texture and color information [27] for further detection improvement. It should be noted that in this thesis, person detection was not applied. Rather, hand-labeled data was used, but person detectors can be used to further enhance the techniques presented.

Now, in order to accumulate further information from the detected person, a tracking approach can be used. Person and object tracking is a very large topic itself and is not part of this thesis, therefore it is not discussed in this section.

The masks obtained by the background segmentation often provide a very inaccurate segmentation of the person, with the major reason being low contrast regions where background segmentation produces very scattered results. Therefore, further processing may be desired. Since real-time performance is an important aspect of a person re-identification system, a trade-off between segmentation quality and speed has to be made. A method that satisfies both needs does not exist. On the other hand, an accurate segmentation is required for improving results in further steps.
The background segmentation method described earlier [33] provides reasonable results in many cases, but suffers from poor segmentation when a person with a similar appearance as the background is segmented. On the other hand, one may notice that the masks of the observed persons in a surveillance scenario will look very similar,



(a) (b)

Figure 3.1: Background segmentation examples (person, background segmentation, weighted mask, mean mask)

even if persons can take arbitrary poses. For example, the center region of the mask is most likely occupied by the upper body. Therefore, this information can be used to enhance segmentation quality. A poor background segmentation mask $\mathbf{M}$ at a certain position $\mathbf{x}$ in the image $\mathbf{I}$ can be improved by weighting it with the masks that have been observed around that position in the past. Therefore, given n masks $\mathbf{M}_i$ (resized to match the size of $\mathbf{M}$), obtained from the background segmentation algorithm, a weighted mask $\tilde{\mathbf{M}}$ can be calculated according to Equation (3.1).

$$\tilde{\mathbf{M}} = \frac{1}{\sum_{i=0}^{n} w_i} \sum_{i=0}^{n} w_i \cdot \mathbf{M}_i \text{ with } w_i \propto dist(\mathbf{x}, \mathbf{x}_i) \tag{3.1}$$

While the mask $\tilde{\mathbf{M}}$ will still include background pixels, it is very likely that most of the probability will be centered on the person itself. The weights w_i can be chosen according to the distance between the center points of the masks $\mathbf{M}$ and $\mathbf{M}_i$. This weighting penalizes masks that are spatially further away. The use of cameras with a very short focal length, i.e. a wide field-of-view, is very common in surveillance settings. Therefore, the observations of persons are distorted especially when they are made near the border of the camera image. The first principle component of the mask is most likely to point in a very different direction on the left side of the field-of-view than when observing the same person on the right side. Consequently, a very small weight should be assigned to the masks spatially further away in the image plane in order to obtain acceptable results.

Examples of the different masks are shown in Figure 3.1. Both examples (a) and (b) were taken from the same camera. The first image shows the person, the second the result of the background segmentation, the third the weighted mask and the fourth image shows the mean mask obtained from all samples in this camera. The mean mask shows a good estimate for example (a), but is obviously not suitable for (b). The background segmentation is insufficient in (a), because of the low contrast in the image. The weighted masks show a good compromise in both cases.

3.2 Brightness Transfer Function

So far, techniques for obtaining the bounding boxes and segmentation masks of persons in each camera have been described. In real world scenarios, it is very unlikely to have the same illumination conditions at two different camera sites, especially when camera sites are in- and outdoor. To overcome this mismatch, a commonly used technique is the application of a Brightness Transfer Function (BTF). The objective of the BTF is to map brightness values between the different camera sites. It is assumed that the illumination conditions at each camera site are fixed. Obviously, this assumption is seldom true for real world scenarios and the case of changing illumination over time will be discussed later in this chapter. While the BTF is used to match persons in non-overlapping camera views, its underlying idea comes from [10], where the main goal was recovering images that captured a static scene with different exposure times.

In order to calculate a Brightness Transfer Function, a pair of observations O_i and O_j of the same person, obtained from cameras C_a and C_b, respectively, are required. The Brightness Transfer Function between O_i and O_j is denoted as f_{ij}. This function f_{ij} maps an arbitrary brightness value B_u in O_i to the corresponding brightness value B_v in O_j.
Given pixel to pixel correspondences between the observations O_i and O_j, the Brightness Transfer Function f_{ij} could be established according to Equation (3.2). In order to obtain a bijective function f_{ij}, a least squares approach could be used to eliminate one-to-many mappings.

$$B_v = f_{ij}(B_u) \tag{3.2}$$

However, due to the deformable properties of cloth, self occlusion, change of scale and geometry and the arbitrary pose a person can take, it is not possible to find pixel to pixel correspondences between O_i and O_j. Thus, a different approach has to be chosen.

If it is assumed that the percentage of image points in O_i with a brightness less or equal to B_u is equal to the percentage of image points in the observation O_j with a brightness less or equal to B_v, it is possible to calculate such a mapping. Even if this assumption simplifies reality, a simple example can justify why it will still produce reasonable results in real world scenarios. If the observation O_i is made under a certain illumination, e.g. a simple light bulb in an empty room with no windows and the observation O_j under the same illumination, but with the bulb being dimmed, it can obviously be observed that O_j will look much darker than compared to O_i. A look at the histograms of O_i and O_j will reveal that the distribution will look slightly similar, but the one calculated from O_j will be scaled. Especially in real world scenarios, this scaling will be in a highly non-linear fashion, but it would still be a monotone function. Bright values in O_i will never correspond with even brighter values in O_j.

Now, in order to calculate the BTF f_{ij} between O_i and O_j, let H_i and H_j be the normalized cumulative histograms for O_i and O_j, respectively. Recall the assumption that the percentage of image points in O_i with a brightness less or equal to B_u is equal to the percentage of image points in the observation O_j with a brightness less or equal to B_v. Equation (3.3) follows from the brightness value B_u in O_i and the corresponding value B_v in O_j that satisfy the assumption. Since a BTF f_{ij} will still follow Equation (3.2), B_v can be substituted with $f_{ij}(B_u)$.

$$H_i(B_u) = H_j(B_v) = H_j(f_{ij}(B_u)) \tag{3.3}$$

By calculating the inverted cumulative histogram H_j^{-1}, the Equation (3.3) can be used to obtain the Brightness Transfer Function f_{ij}.

$$f_{ij}(B_u) = H_j^{-1}(H_i(B_u)) \tag{3.4}$$

Thus, in order to calculate the Brightness Transfer Function between observations O_i and O_j, only the cumulative histograms H_i and H_j, respectively, are required. By inverting the cumulative histograms H_j, the BTF can be obtained by simply following Equation (3.4). By using a nearest-neighbor interpolation, a mapping for all possible brightness values is established.
So far, the BTF only maps brightness values in gray-scale images. In order to map color images, each channel (red, green and blue) is processed separately. Obviously, a Brightness Transfer Function is required between each pair of cameras, i.e. given n cameras, a total of $\frac{n(n-1)}{2}$ functions is required to map between each possible camera link.
Given the observations of several different persons or multiple observations of one specific person in a camera site, there exist several ways to fuse the information to form one single BTF. This BTF can then be used to transform the colors of unseen observations between cameras.

3.2.1 Mean Brightness Transfer Function

Given the corresponding observations O_i and O_j for a total of n persons obtained from camera C_a and C_b, respectively, a single Brightness Transfer Function f_{ij} can be calculated for each person correspondence. Let $M = \{f_1, f_2, \ldots, f_n\}$ be the Brightness Transfer Functions obtained from the n persons with the observations O_i and O_j. A mean Brightness Transfer Function (MBTF) can be calculated according to Equation (3.5) by taking the mean of all given functions.

$$MBTF = \frac{1}{n}\sum_{i=1}^{n} f_i \tag{3.5}$$

If p observations of a person at the camera site C_a and q observations of the same person in camera C_b exist, a total amount of $p \cdot q$ Brightness Transfer Functions can

be calculated. As mentioned in Section 3.1, the foreground masks of the observations may contain errors. Therefore, many error-prone BTFs are going to be calculated. In this case, a better performing transfer function may be obtained by accumulating the multiple observations of the person in one histogram per view. Afterwards, the BTF can be calculated.

3.2.2 Cumulative Brightness Transfer Function

If n persons with the corresponding observations O_i and O_j are given for training a BTF, the Cumulative Brightness Transfer Function (CBTF) is another way to merge the given information in order to form one transfer function. The idea of the CBTF is to accumulate all observations of all persons view-vise in one histogram in a first step. In a second step, the regular BTF calculation procedure is applied. The BTF for a single person, which does not cover all brightness values in at least one view, will have a flat response either at beginning or the end of the brightness value range. This range is still mapped by the BTF, but in a many-to-one fashion, because there was no information present to learn the mapping of this range. When calculating the MBTF, such functions contribute incorrect information to the final function and therefore produce a less accurate result. The CBTF accumulates all information and will most likely cover the whole brightness range. This is due to the fact that accumulation of many different examples, which do not cover the complete range independently, will most likely cover the whole range when merged (assuming that they are drawn from a representative set). As a consequence, a less error-prone mapping will be established. In Section 4.4.5, Chapter 4, the differences between the MBTF and the CBTF are further studied and compared.

3.3 Color Histograms

Now, the characteristic illumination setting at one camera site can be adjusted to match the condition at the other site by using the BTF (and vice versa). Since the observations then share a common appearance of colors, it is possible to calculate a person descriptor that will produce much better matching results than without a color adaption.
Histograms are a widely used technique and have shown, apart from person recognition, a successful application in tracking, object recognition, texture classification and many other areas. A quick review of histograms and how to compare them is shown in the next section.

3.3.1 Definition

Given a gray-scale input image $I(x)$, a histogram H representing the brightness statistics can be calculated according to Equation (3.6), with i indexing the histogram bins.

$$H(i) = \sum_{x=0}^{N} \delta_i(I(x)) \text{ with } \delta_i(x) = \begin{cases} 1, & \text{if } x \text{ falls into bin } i \\ 0, & \text{else} \end{cases} \tag{3.6}$$

If $p(x)$ models the probability of each pixel being foreground information in the input image $I(x)$, a probabilistic histogram can be calculated according to Equation (3.7). In the non-probabilistic case each pixel accounts for one increment in the corresponding bin. In the probabilistic case, each pixel only accounts as much as the value of its assigned probability.

$$H(i) = \sum_{x=0}^{N} p(x)\delta_i(I(x)) \text{ with } \delta_i(x) = \begin{cases} 1, & \text{if } x \text{ falls into bin } i \\ 0, & \text{else} \end{cases} \tag{3.7}$$

For calculating a Brightness Transfer Function as shown in Section 3.2, the cumulative histogram H_c is required. H_c can be obtained from a histogram H according to Equation (3.8).

$$H_c(i) = \sum_{j=0}^{i} H(j) \tag{3.8}$$

The gray-scale case can easily be extended to be used on multi-channel images, e.g. color images. By splitting the color input image to its separate channels, e.g. the red, green and blue channel, a histogram can be calculated for each channel separately. On the downside, the independent processing of the channels removes the brightness relations, i.e. the color information. In many cases it is desired to keep this information, therefore often histograms of a higher dimension are used.

Given a three-channel input image $\mathbf{I}(x)$, a three dimensional histogram $\mathbf{H}$ with the color histogram bins $r \times g \times b$ can be calculated according to Equation (3.9) in order to represent the color statistics. The probabilistic formulation is straight-forward.

$$\mathbf{H}(r,g,b) = \sum_{x=0}^{N} \delta_{r\times g\times b}(\mathbf{I}(x)) \text{ with } \delta_{r\times g\times b}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ falls into bin } r \times g \times b \\ 0, & \text{else} \end{cases} \tag{3.9}$$

When using color images with three channels and a bit depth of 8, the 3d histogram could have $2^8 \cdot 2^8 \cdot 2^8 = 16777216$ bins. While such a large number occupies a large amount of memory, especially when using multiple histograms, it is not feasible to work with such fine quantized objects. Therefore, a quantization is applied to reduce dimensionality, especially when using multi-dimensional histograms. For $R \times G \times B$ histograms, a quantization of for example $16 \times 16 \times 16$ bins could be chosen.

3.3.2 Comparison

A commonly used distance measure to compare two histograms H_1 and H_2 is the Bhattacharyya distance as shown in Equation (3.10). N denotes the total number of bins and $\bar{H}$ denotes the average of all histogram bins $H(i)$.

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\bar{H}_1 \bar{H}_2 N^2} \sum_{i=0}^{N} H_1(i) \cdot H_2(i)} \tag{3.10}$$

When calculating a single histogram for each color channel (in case of an RGB image), a distance comparing all three histograms can be realized for example by taking the mean of all distances (Equation (3.11)).

$$d(H_1, H_2) = \frac{1}{3} \cdot (d(H_1^R, H_2^R) + d(H_1^G, H_2^G) + d(H_1^B, H_2^B)) \tag{3.11}$$

Without normalization, histograms are not scale-invariant and therefore are not useful for person re-identification. Hence, in order to compare histograms, it is desired to normalize them first. A normalization according to the ℓ_1 norm (see Equation (3.12)) compensates for the bounding box size and enables a comparison between histograms that is not influenced by it. Note that an equivalent normalization as in Equation (3.12) is already modelled in Equation (3.10).

$$H_{norm}(i) = \frac{H(i)}{\sum_{i=0}^{N} |H(i)|} \tag{3.12}$$

With this information given it is possible to build a person re-identification system. A background modeling approach and a person detector can be used to detect and track persons within each camera. The background segmentation can further be used to extract masks for the observed person. An unsupervised method was proposed here to improve the segmentation of erroneous masks by weighting them with the surrounding masks. Given hand labeled correspondences between the cameras, a Brightness Transfer Function can be calculated to compensate the different, fixed illumination conditions at the camera sites. Afterwards, histograms and the Bhattacharyya distance can be used to compare the different observations and obtain matching scores between observations.

3.4 Iterative Brightness Transfer Function

In the first half of this chapter, it was assumed that the illumination at each camera site stays constant over time. This is a very restrictive assumption, since it rarely

applies to real world scenarios. The learned function may produce good color adjustments when the illumination stays fixed at both sites, but will produce erroneous results when lighting conditions change. Therefore, a major drawback of the Brightness Transfer Function approach is the fixed training phase. An adaptive, unsupervised BTF can drift and may have problems with a correct initialization, especially when illumination settings largely differ.

In [7], an adaptive BTF is proposed that tries to map the current observation (in an entry/exit region) back to training conditions by utilizing a BTF obtained between the background seen while training and the currently observed background. While the results improve when transferring colors under new illumination conditions, it requires a well-working background segmentation and perfectly overlapping camera images. The entry and exit regions are required to have a static background, which is rarely given in crowded scenes. In [8], the BTF is constantly updated by using new observations. Only observations that gave a small matching distance are chosen to update the BTF. Sudden lighting changes are compensated by giving the spatial-temporal descriptors more weight than the appearance model. Still, there are situation where the system can drift, as for example when observing several persons at a time after a large change in illumination occurred with no new observations made in the meantime.

In the following section a novel approach is presented that makes use of the Brightness Transfer Function for an iterative adjustment of the intra-camera illumination in a scenery. As mentioned before, the fixed training phase is a very critical point in real world scenarios and will generate error-prone results when illumination conditions change. The idea is to map the circumstances of a new frame back to pre-defined conditions. Having such a pre-defined fixed condition would justify the use of a fixed training stage for the inter-camera BTF, because illumination in the incoming frames could be mapped back to this condition and a more constant appearance of colors within a camera would be given.

Observing a single, static scene at two different points in time can obviously result in two very different observations. Several reasons can be pointed out for this behavior such as the change of the position of the light, e.g. the sun, which results in different casts of shadows. Furthermore the color temperature may have changed, especially when ambient and artificial light are combined or changed. In addition to that, the amount of light that reaches the scenery may vary, caused by day and night or by switching off or dimming artificial light. All these reasons will induce a decrease in performance of the pre-trained inter-camera Brightness Transfer Function.

Let us assume that the effects by shadow casts are negligible. In a static scene the Brightness Transfer Function can be used to map the new illumination condition back to the former condition. In this case the observations O_i and O_j would be the whole scenery images, with O_i being an image from the training phase (where constant con-

ditions are assumed) and O_j being an image of the scenery under a new condition. In theory, mapping O_j back to the conditions of O_i would produce an identical observation O'_j, but since the BTF cannot handle one-to-many and many-to-one mappings, which may occur, the images would look similar, but not perfectly identical. Those one-to-many and many-to-one mappings are likely to occur due to over- or underexposure.

However, in real world surveillance scenarios, scenes are usually populated by humans and other non-static objects, e.g. cars. Given two observations, similar to the last case, with non-static objects present, a background segmentation algorithm could identify foreground regions. The Brightness Transfer Function can then be calculated between the images by discarding the areas marked by the union of the foreground masks. An assumption that justifies this approach is that the foreground objects undergo the same changes due to illumination as the background does. On the one hand, this assumption is not correct, since the reflectance properties of especially clothing differ from the reflectance properties of rigid materials as for example concrete or wall paint. One the other hand, this is a similar approach to [7], where it is shown that this technique produces a reasonable mapping for transferring a new lighting condition back to the training condition. A drawback is the background modeling algorithm. An object positioned somewhere in the scene will be adapted by the background model after a while. As a consequence, it would introduce errors to the BTF, because it would be used to calculate the BTF even without appearing in the other image. Furthermore, background segmentation rarely mirrors the ground truth, resulting in an incorrectly estimated BTF.

The iterative approach proposed here, in the following refereed to as *iterative BTF*, can be used to adjust a new incoming frame to a pre-selected frame in order to compensate for different illumination conditions. A simple, unsupervised algorithm for selecting the reference frame is also proposed.
Given two images I_i and I_j of the same scene, taken at different times with different illumination conditions, the iterative BTF will map the conditions of I_j back to I_i. The assumption, that the amount of foreground observed is smaller than the amount of background in both images, is valid for most real world surveillance scenarios. The algorithm works in a iterative fashion:
In a first step, the Brightness Transfer Function between I_i and I_j is calculated. If both images do not include any foreground areas, i.e. the scene is empty, it is the same case as the first one discussed in this section and the calculated BTF will be a good estimate for the mapping between the different conditions. When foreground areas are present, which is very likely, the obtained BTF will be error-prone, because it is not very likely that the foreground has the same color distribution and occupies the same amount of pixels in both images. Still, since the foreground did only occupy a small part of the whole image, the BTF can be applied to the image I_j to obtain I'_j. The image I'_j will be a rough estimate of the conditions in I_i. As a consequence, the

image I'_j will have a more similar illumination condition to I_i as I_j did have.
In a second step, the colors that only exist in one of the images I_i and I'_j are discarded. Let K_i and K'_j be the colors in I_i and I'_j, respectively. The colors $\tilde{K}_i = K_i \setminus K'_j$ do only exist in image I_i and $\tilde{K}'_j = K'_j \setminus K_i$ do only exist in image I'_j. These colors, since they only exist in one image, produce errors in the Brightness Transfer Function and should therefore be removed from the images I_i and I_j. This can be done easily by using a binary mask over the whole image. It should be noted that for masking out the pixels in I_j, the transformation of $\tilde{K}'_j$ back to conditions of I_j needs to be applied by using the corresponding inverted Brightness Transfer Function. Obviously, at this point in time, $\tilde{K}_i$ and $\tilde{K}_j$ will include nearly all colors that appear in the images, because the mapping for obtaining I'_j was not very accurate. Consequently, only pixels in I_i that show a large distance to the pixels in I'_j should be masked out and vice versa. It can happen, that one image shows many colors that do not exist in the other image. Therefore, the proportion of pixels masked out would be very different, resulting also an inaccurate BTF. Since the images do not necessarily overlap, no pixel correspondences are known. The union of both masks would therefore not be usable. One way to overcome this problem is to randomly mask out pixels in the image where less pixels were removed.
This procedure can be repeated until the change in difference between the current I'_j and I_i drops below an error bound ϵ. In the first step, the Brightness Transfer Function will be inaccurate, but it improves in the second step, because colors that did not exist in both images were removed from calculation when obtaining the new BTF. The whole procedure is summarized in Algorithm 1.

Data: I_{ref}, I_{new}
Result: BTF for color adaption of I_{new} back to I_{ref}
repeat
  Calculate BTF f from I_{new} to I_{ref};
  $I'_{new} = f(I_{new})$;
  K_{ref} = Histogram(I_{ref});
  K'_{new} = Histogram(I'_{new});
  $\tilde{K}_{ref} = K_{ref} \setminus K'_{new}$;
  $\tilde{K}_{new} = f^{-1}(K'_{new} \setminus K_{ref})$;
  Mask colors $\in \tilde{K}_{ref}$ in I_{ref};
  Mask colors $\in \tilde{K}_{new}$ in I_{new};
  Randomly mask pixels in I_{ref} or I_{new} in order to match the difference in masked pixels between I_{ref} and I_{new};
until $\Delta dist(I_{ref}, I'_{new}) < \epsilon$;

Algorithm 1: Iterative BTF

Deciding which colors are present in both images and which are not is a critical step. Comparing every pixel in I_i to every other pixel in I'_j would require too much computational effort. A simple solution is employing 3d histograms in $R \times G \times B$

space. During the first iteration steps, the quantization for the histogram bins should be very coarse in both images, because the mapping function is not very accurate. In further steps the quantization can be changed to a more fine scale, because the Brightness Transfer Function becomes more accurate. Deciding whether the colors in a bin exist in the other image can be done by comparing the same bins in the $R \times G \times B$ histograms for both images against each other. If the difference between bins is significantly large, it is likely that those colors do not exist in both images. Especially when one bin is empty and the corresponding bin for the other image contains pixels, it is very likely that those pixel colors do not exist in the other image. Consequently, those pixels can be masked out in the next iteration step.

3.4.1 Selection of a reference frame

The iterative BTF can be used to adjust a new incoming frame I_j to a reference frame I_i. Selecting a reference frame can be done by hand in order to meet subjective criteria such as overall brightness, appearance of colors, etc. One the other hand, an unsupervised algorithm can be used to obtain such a reference frame. The idea is to capture frames over a longer period of time in order to cover the most common illumination conditions. Each frame I is then scored according to Equation (3.13) with $\bar{I}$ being the mean color of the image. *Underexposed* and *Overexposed* refer to the percentage of pixels in I that were underexposed or overexposed, respectively. A pixel can be considered being underexposed if its brightness values for the red, green and blue channels are all below a small value γ (e.g. 3). Are the values of a pixel greater than a maximum value δ (e.g. 253) then it can be considered to be overexposed. Those pixels are likely to produce one-to-many or many-to-one mappings in the BTF, because they do not contain any useful information. Therefore, they should penalize the score of the image. The constant ϵ is a small number in order to avoid assigning a too great score to images that perfectly satisfy the gray world assumption but having many over- or underexposures at the same time.

$$\text{Score}(I) = (\|\bar{I} - \begin{pmatrix} 127 \\ 127 \\ 127 \end{pmatrix} \|_2 + \epsilon) \cdot (\text{Underexposed}_I + \text{Overexposed}_I) \tag{3.13}$$

All in all, each frame is scored according to how well it satisfies the gray world assumption and what percentage of the image pixels were over- or underexposed. After scoring each frame, the frame with the lowest score is picked as a reference frame. It should be noted that this process is not time critical, because it only needs to be performed once before training the BTF between cameras.

In order to improve the system described in the first part of the chapter, the iterative BTF and the selection of the reference frame are employed as follows:
First, a large amount of video data is collected. Then, each frame is scored and a

reference frame is selected. This is a completely unsupervised process and therefore does not require any human interaction. This is done for every camera separately. From then on, each incoming camera frame is adjusted to its reference frame by using the iterative BTF in order to compensate for changing illumination conditions. Now, the process described earlier can take place: background segmentation, collecting training examples for the inter-camera BTF, training the transfer function and then taking the system into use.

3.5 Summary

In this chapter, an exemplary system for re-identifying persons in multiple, non-overlapping cameras was described. A big problem in such scenarios are the different illumination conditions that exist at the different camera sites. Under the assumption of a constant illumination at each site, a Brightness Transfer Function can be used to map brightness values from one camera to another, where different illumination condition exist. Since the assumption of constant illumination does seldom apply to real world scenarios, the accuracy of the BTF decreases when conditions at one site change. In order to overcome these problems, the BTF was extended to an iterative version which tries to render illumination more constant by adjusting each new incoming frame to a pre-selected reference frame in each camera. An unsupervised algorithm for automatically selecting such a reference frame was also proposed.

4 Experiments

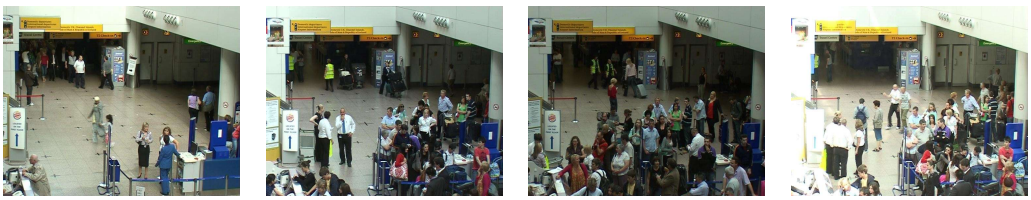
In this chapter the experimental results are presented: Firstly, the inter-camera Brightness Transfer Function is evaluated and analyzed. In the second part of this chapter, the iterative BTF is used to compensate changing illumination over time intra-camera-wise and the gain in accuracy is measured.

4.1 Data Set

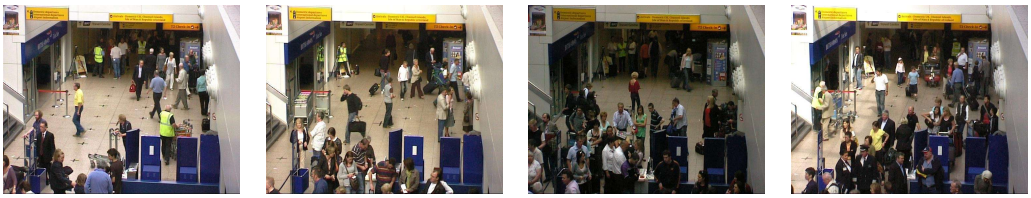
The following experiments were carried out on the PETS 2007 benchmark data [9]. The data set was captured at an airport in order to evaluate performance of algorithms detecting loitering, attended luggage removal and unattended luggage in general. It is a multi-camera data set and includes four different views. Each camera captures roughly the same scenery, but each from a very different perspective. The data is split into ten sequences of varying length, which add up to approx. 21 minutes of video data (at 25 fps). The data was not captured in one session, therefore time gaps exist between the sequences. Hence, the data shows only segments of a time window longer than 21 minutes including many different illumination settings. Some randomly chosen frames from each camera are shown in Figure 4.1. The different lighting conditions inter- and intra-camera-wise raise a difficult problem for re-identifying persons among the views. Furthermore, the frames of the different sequences do not overlap, presumably because of moving the cameras during and between the sequences.
A total of 118 persons were hand labeled using rectangular bounding boxes (b.b.) in order to simulate the output of a tracker/detection algorithm. In order to keep the labels as consistent as possible, the boxes were aligned to the head, shoes and the shoulders. Labeling was performed when most parts of the body were visible. A few examples are shown in Figure 4.2. In order to save time while labeling, only a few labels were created for each person in each camera. The observations of one exemplary person are shown in Figure 4.3. The intra-camera statistics for all labels are printed in Table 4.1. Since the matching was done between cameras, the number of persons observed in a pair of cameras are listed in Table 4.2.

4.2 Experimental Setup

Before conducting any experiments, an overview over the used descriptors, comparison method and recognition accuracy measurement is given.

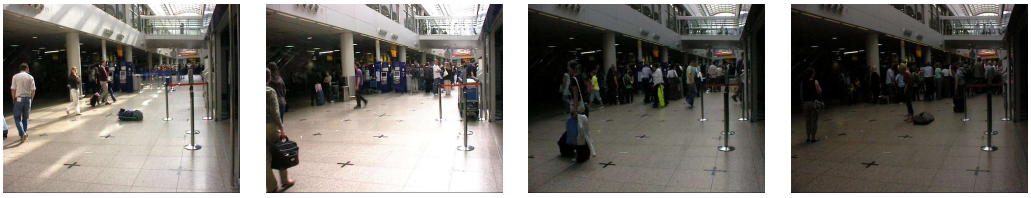


(a) Examples camera 1



(b) Examples camera 2

(c) Examples camera 3



(d) Examples camera 4

Figure 4.1: Examples from the four camera views

Table 4.1: Data set: Intra-camera stats

Camera	Persons	Average label count per person	Average b.b. size
1	118	2.5	37×141
2	111	2.5	38×152
3	97	3.1	42×129
4	91	2.9	41×166

Table 4.2: Data set: Inter-camera stats

Camera pair	1-2	1-3	1-4	2-3	2-4	3-4
Number of persons	111	97	91	90	90	84

(a)

(b)

(c)

(d)

Figure 4.2: Person examples for camera (a) 1 (b) 2 (a) 3 (b) 4 (resized for viewing purposes)

Figure 4.3: Observations of one exemplary person in all four cameras

4.2.1 Descriptors and comparison

For describing the appearance of a person, two different kinds of histograms are employed:

- **Histogram 1d**: A single histogram is built for each color channel (red, green, blue) separately.

- **Histogram 3d**: A histogram with three dimensions is built over the $R \times G \times B$ space with a quantization of $16 \times 16 \times 16$ bins.

A drawback of histograms is that they do not capture any spatial information. While this may not be important in some other areas, it is very desirable to retain this property in person recognition. In many cases, the upper body and the lower body part of a person are dressed in different colors [20, 21]. Obviously, this clue can be used to improve recognition rates. A large variety of techniques exist for incorporating the spatial information e.g. spatiograms [40]. In this approach, a very simple and intuitive technique is employed. Instead of computing a single histogram for the whole bounding box, the box is divided into smaller slices along the vertical axis. For all experiments, the box is sliced into $n = 10$ non-overlapping slices of equal height. Two sliced histograms $H_i = \{H_i^1, H_i^2, \ldots, H_i^n\}$ and $H_j = \{H_j^1, H_j^2, \ldots, H_j^n\}$ can be compared according to Equation (4.1) with $Bhatt(\cdot, \cdot)$ referring to the Bhattacharyya distance. The weight w_i^s accounts for histogram slice H_i^s and is proportional to the amount of foreground in slice s divided by the total amount of foreground in the bounding box. Therefore, slices with less foreground information receive a smaller weight. A small amount of foreground is likely to occur due to an insufficient foreground segmentation, therefore the slice may not represent the color distribution well within the slice.

$$d(H_i, H_j) = \sum_{s=1}^{n} w_i^s \cdot Bhatt(H_i^s, H_j^s) \text{ with } \sum_{s=1}^{n} w_i^s = 1 \tag{4.1}$$

Probabilistic histograms are employed for all experiments in this chapter in order to model the uncertainty of an imperfect segmentation mask. Unless otherwise stated, a weighted foreground mask was used for all experiments (see also Chapter 3, Equation (3.1)). The overall mask $\tilde{\mathbf{M}}$ for each observation at the pixel position $\mathbf{x}$ was calculated as shown in Equation (4.2) with $\mathbf{x}_i$ being the spatial position of the mask $\mathbf{M}_i$ in the image. The variance σ^2 was determined empirically and set to 0.1.

$$\tilde{\mathbf{M}} = \frac{1}{\sum_{i=0}^{n} w_i} \sum_{i=0}^{n} w_i \cdot \mathbf{M}_i \text{ with } w_i = e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|_2}{\sigma^2}} \tag{4.2}$$

4.2.2 Rank and recognition accuracy measurement

For measuring the overall recognition performance of the system, the *Rank and Recognition accuracy* measurement is used. Without a doubt, it is very desirable to be able to match persons perfectly among views, but as the number of targets grows this task becomes more difficult. In surveillance scenarios it is more important to know, how long it would take an operator to find the target in a sorted list (see [18]). This list can be sorted according to the distance of the targets to the previously picked person, which leads to the *Rank and Recognition accuracy* measurement. Therefore, the matching problem is regarded as a ranking problem. This form of measurement relates the rank n to the recognition accuracy under which the correct match can be found within the n first results of the sorted list.

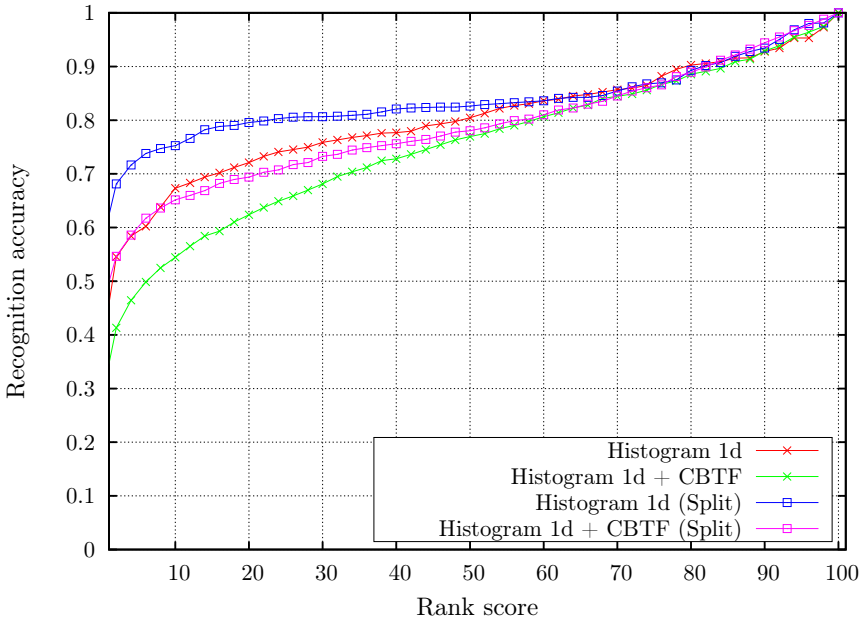
In order to measure the recognition accuracy in combination with a Brightness Transfer Function, i.e. obtaining a rank and recognition accuracy curve, cross validation is employed. Given two cameras C_a and C_b with n observations of the same persons in both cameras, the n correspondences are split into a training and a testing set. The training set is used compute the Brightness Transfer Function and the testing set is used to estimate the rank and accuracy curve. Unless otherwise stated, a fixed number of 10 examples is used to train a BTF and the remaining $n - 10$ examples are used for evaluation. Therefore, the number of folds required is adjusted to n. Instead of picking the training examples at random, the whole observation set is sorted according to the time of appearance in camera C_a. Then, the set is split into $\lfloor \frac{n}{10} \rfloor$ training sets with consecutive examples in terms of temporal appearance. It should be noted that the last $n - 10 \cdot \lfloor \frac{n}{10} \rfloor$ observations at the end of the sorted list are never used for training (i.e. just for testing). Otherwise, a fixed sized training and testing set would not be possible.

4.3 BTF Evaluation

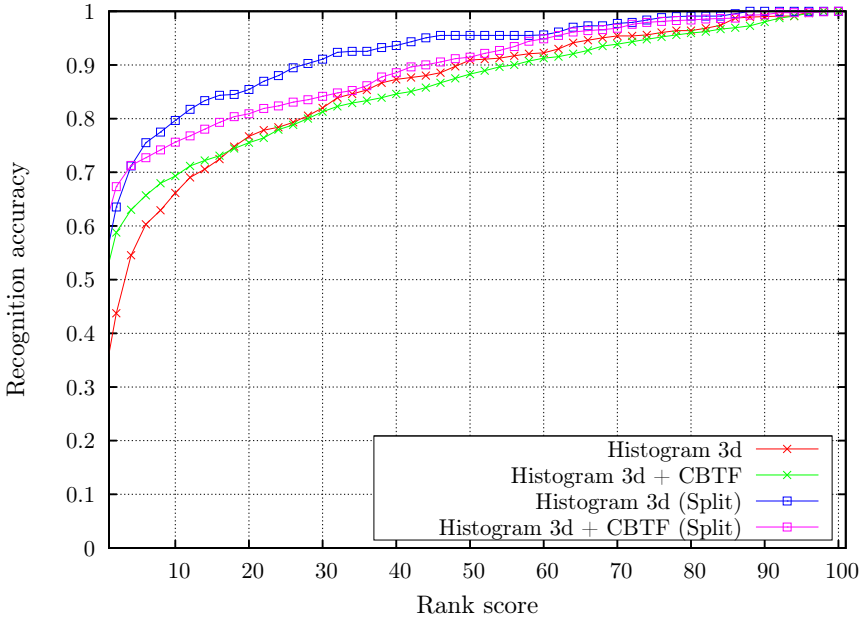
In this section the benefits of the BTF and some of their drawbacks and differences are discussed. The illumination conditions over time are assumed to be constant. Although this does not reflect reality, this assumption is essential to make use of the BTF. Without it, the idea of a fixed training phase is pointless. The temporal aspect of changing lighting will be discussed in the further sections.
In the following sections, camera 1 is chosen as the site where the initial observation of a person is made. Consequently, the task is to find this person again in cameras $2, 3$ and 4. When the BTF is used, the colors in camera $2, 3$ or 4 are mapped back to the colors in camera 1.

4.3.1 Improvement due to the CBTF

In a first experiment, the recognition accuracy between cameras 1 and 2 is measured. As mentioned earlier, these two cameras share a fairly similar view and do not show a large difference in color appearance.
At first, Histograms in 1d space are evaluated and the results can be seen in Figure 4.4(a). Two observations can be made: a gain in accuracy when employing histogram splits and a drop in precision when applying the CBTF. The first observation is not surprising, since the spatial color distribution is an important clue that is captured by the histogram slices. It is more surprising that the performance decreases when employing the CBTF. Generally speaking, an explanation may be that the BTF introduces small errors, which are most likely due to three reasons: First, the CBTF is trained on a rather small training set with only 10 examples. Second, the training examples, i.e. the persons, do not appear under exactly the same pose, viewpoint and may contain under- or overexposed pixels (i.e. exceeding the dynamic range of the camera). Those pixels are likely to lead to one-to-many and many-to-one mappings. Third, illumination changes over time, but is regarded as fixed in this section. Therefore the BTF is not perfectly accurate. It will be shown later that the second and third points have a much bigger influence than the number of training examples (still the third point will be the most important one). Since the colors are already fairly similar, the BTF worsens the color appearance and consequently the results.
The matching results when using histograms in 3d $R \times G \times B$ space are shown in Figure 4.4(b). As far as the sliced histograms are concerned, the same observations as before can be made. A very interesting observation can be seen when looking at the lower ranks, where the CBTF improves matching performance. It may be possible that the coarse quantization of the histograms compensates the errors introduced by the BTF. Still, due to the similar colors in both views, matching results are fairly equal.
In a next experiment, the matching performance between cameras 1 and 3 is tested. This case is more interesting due to the strong bluish appearance in camera 3 and the different camera angle. Matching results for the 1d histograms are shown in Figure 4.5(a) and results for the 3d histograms in Figure 4.5(b). Both curves show that a Brightness Transfer Function is essential in this scenario. Without it, the accuracy is similar to chance, i.e. just guessing performs equally well. When employing the CBTF, matching accuracy increases drastically. This example also shows that 3d histograms are better suited for person recognition than 1d histograms. The gain obtained when using 3d histograms in combination with a brightness transfer is not trivial, since the BTF maps each brightness channel separately. Hence, it is not necessarily given that a brightness triplet (r, g, b) is mapped correctly. On the one hand, a coarser quantization of the histogram bins may compensate for the small inaccuracies that are introduced by the BTF. On the other hand, a coarse histogram quantization may also worsen the matching performance, since a very coarse categorization of colors does not represent the observation of a person accurately. A trade-off between color transfer error, his-



(a)



(b)

Figure 4.4: Comparison between camera pair 1-2 (a) 1d histograms (b) 3d histograms

togram bin quantization and matching performance has to be made. In this example, the given parameters show a good compromise, since the overall accuracy is improved.

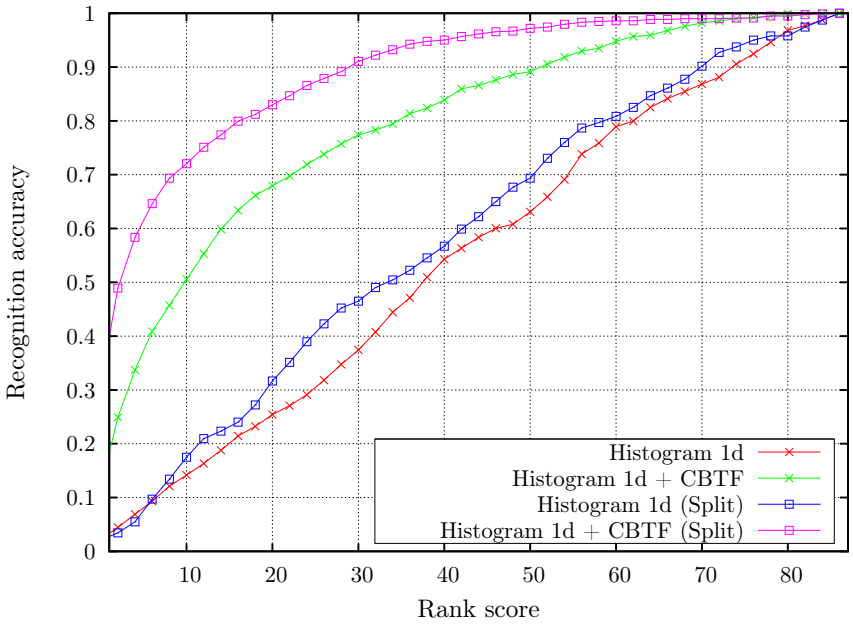
Even with changing viewpoints, the histogram splits still capture the spatial color distribution and therefore improve results. This is due to the fact that a different viewpoint does still preserve the order of the splits, at least as long as the observation angle between camera and floor is much smaller than 90 degrees.

Results of the matching between camera 1 and camera 4 are shown in Figure 4.6(a) and Figure 4.6(b). The viewpoint in camera 4 is on eye level, therefore also very different from camera 1, which is above head level. On the one hand, the brightness transfer improves results, even if the gain is not very big. On the other hand, the gain is always dependent on how different the colors appear in the views. In this example in particular, only a small subjective mismatch exists, which verifies the rather small gain. It can be concluded that in this case, the findings are similar to the previous scenario between cameras 1 and 3 and that the same argumentation holds.

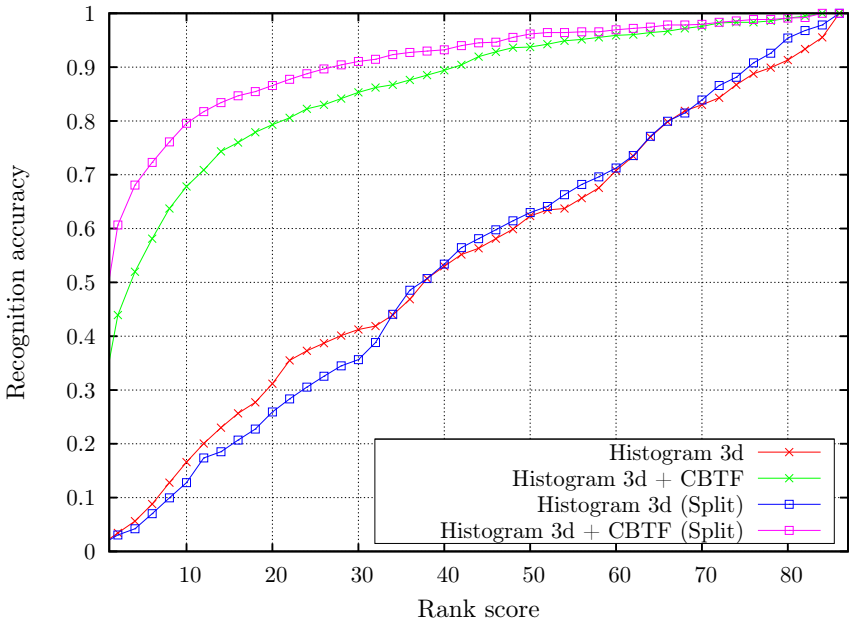
A few corresponding observations between the cameras are shown in Figure 4.7(a)-(c). Examples from camera 1 (Figure 4.7(a)) and the corresponding examples in cameras 2, 3 and 4 (Figure 4.7(b)) show a large color mismatch, especially between cameras 1-3 and 1-4. After applying the CBTF (Figure 4.7(c)) the colors look much more alike, however the colors in the second and third row look very faded. On the other hand, a direct comparison with the image before applying the CBTF reveals that those faded areas used to be close to black. In this range different hues lie close to each other and therefore it is very difficult to recover the correct color information.

4.3.2 Evaluation of the foreground masks

In the previous experiments the foreground mask for each person was obtained by a weighted combination of the surrounding foreground masks from other observations. It is not self-explanatory that this mask performs better than the foreground mask obtained by using the background segmentation. Furthermore, it is not clear whether a mask is required at all.
Therefore, recognition accuracy is evaluated on the previous camera pairs by using the split 3d histograms in combination with the different types of foreground masks. Figures 4.9(a)-(c) show the recognition performance between cameras 1 and 2, 1 and 3, 1 and 4, respectively. Surprisingly, the accuracy does not decrease drastically when using no mask at all. An explanation may be that each observation roughly occupies the same space within the bounding box and that in most cases the background color distribution does not match well with the distributions of the persons. When looking at the background segmentation results on the data, it can be seen that it did not perform well in low contrast regions. In cameras 3 and 4, this problem could be seen most often. In contrast to the masks obtained by the background

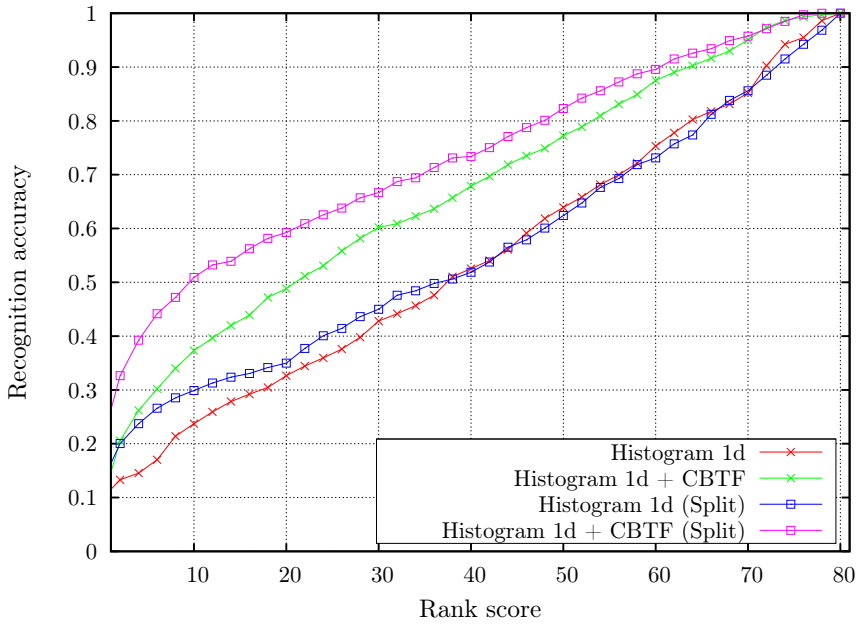


(a)

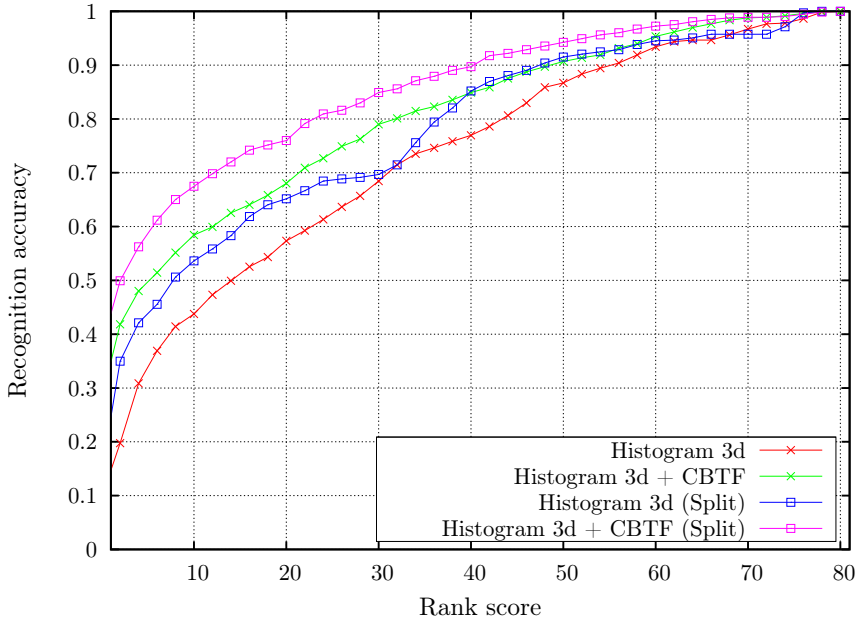


(b)

Figure 4.5: Comparison between camera pair 1-3 (a) 1d histograms (b) 3d histograms

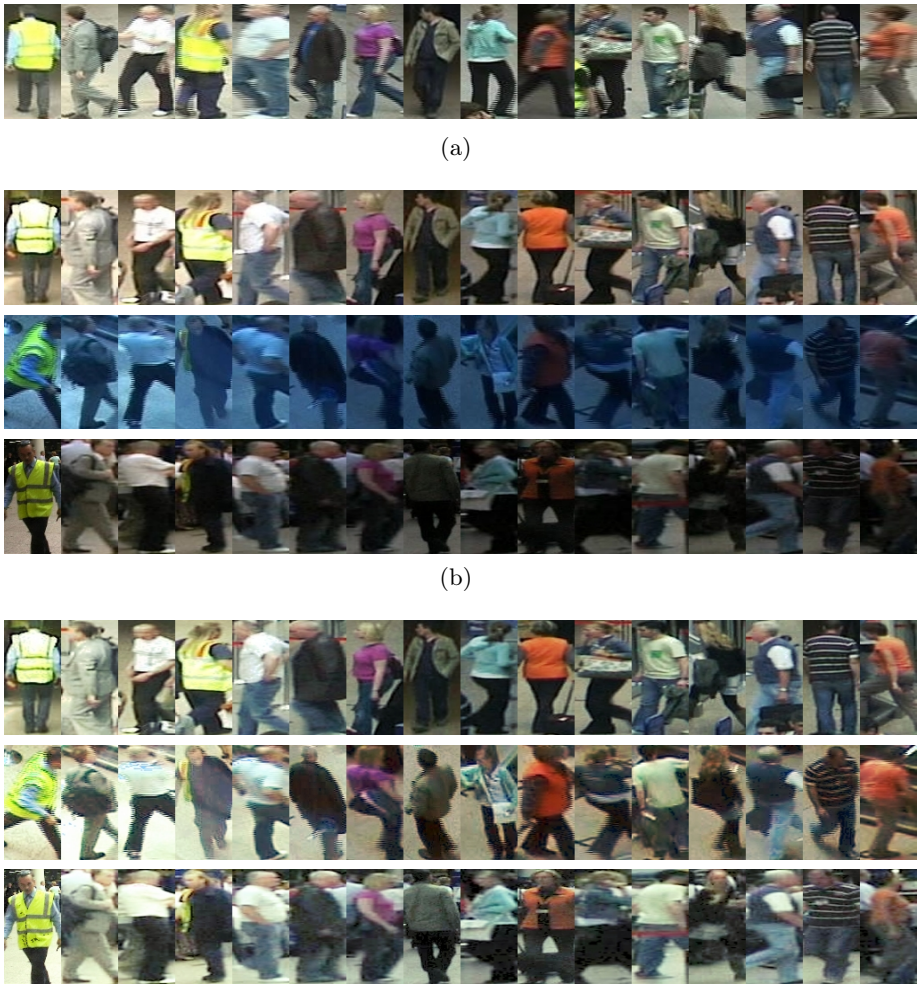


(a)



(b)

Figure 4.6: Comparison between camera pair 1-4 (a) 1d histograms (b) 3d histograms



(a)

(b)

(c)

Figure 4.7: A few corresponding observations between camera (a) 1 and (b) 2, 3, 4 (without CBTF applied) and (c) 2, 3, 4 (with CBTF applied)

(a) (b)

Figure 4.8: Some erroneous examples of the (a) foreground segmentation masks and the resulting (b) weighted masks

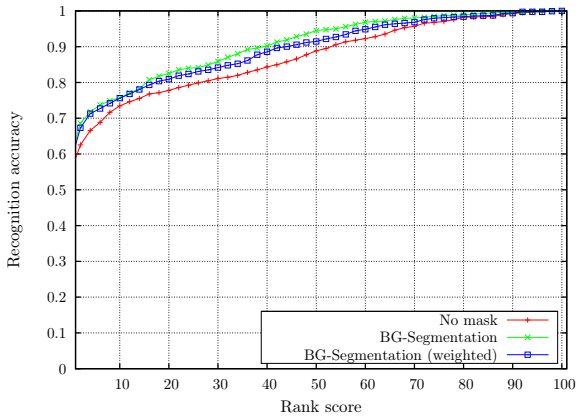
modeling algorithm, the weighted masks can overcome this problem by fusing the very inaccurate foreground mask with the weighted mask of the surrounding masks, which leads to a better performance in camera 3 and 4. On the contrary, these problems did occur very seldom in cameras 1 and 2. Consequently, the foreground and the weighted mask perform nearly identical. The improvement due to the weighted masks can be seen in Figure 4.8, where some examples with a very inaccurate foreground estimation are shown in (a). The corresponding weighted masks in (b) show a much better segmentation.
(Note: The weighted masks contain pixels within the range of $[0, 1]$. For better visualization only pixels greater than 0.4 are shown and the weighting information for each pixel is discarded, i.e. set to 1.)

4.3.3 Summary

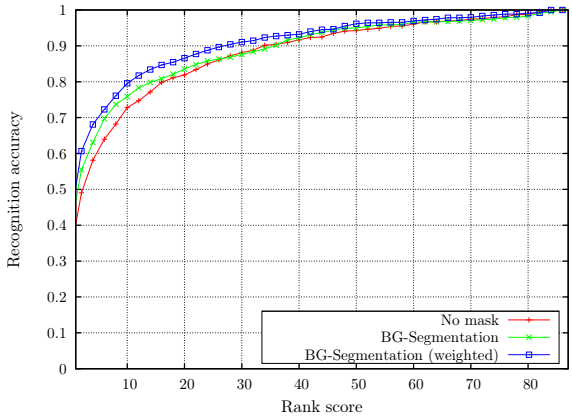
In this section the CBTF and the different foreground segmentation masks were evaluated. In order to apply a BTF, the assumption of a fixed illumination condition was made for all experiments. The segmentation masks perform equally well, but the weighted masks show improvements when segmenting persons in low contrast regions. As far as the CBTF is concerned, the insight is twofold. When transferring brightness values between cameras with fairly similar colors, the BTF introduces small errors due to its inaccurate estimation. On the other hand, the Brightness Transfer Function can make up for very different illumination conditions at the camera sites and therefore improve results drastically.

4.4 Iterative BTF Evaluation

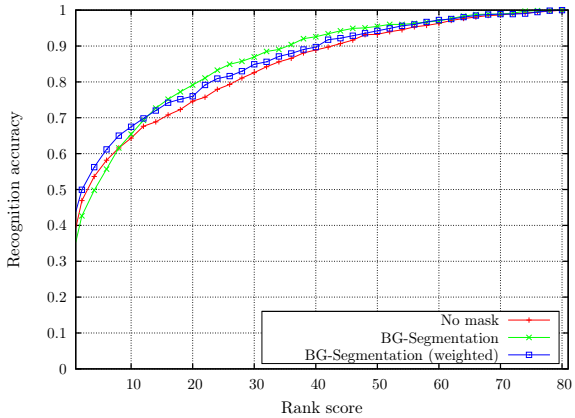
In the last section, the change of illumination over time was assumed to be constant. For most real world scenarios, this assumption is not valid. Therefore it is necessary to reduce the influence of the temporal change in lighting on the matching process. Given



(a)



(b)



(c)

Figure 4.9: Comparison of the foreground masks for camera pair (a) 1-2 (b) 1-3 (d) 1-4

such a compensation, the Brightness Transfer Function only needs to be trained once and would not rely on any fixed illumination condition. Therefore, the intra-camera iterative BTF is evaluated in this section.

4.4.1 Examples of the iterative BTF

The iterative BTF was carried out as mentioned in the last chapter:
Given are two images of the same scene, but taken at different points in time, therefore, most likely, showing different illumination characteristics. Until the change in the Bhattacharyya distance between these two images is smaller than a pre-defined threshold ϵ, the iteration steps are repeated. For the experiments, a threshold $\epsilon = 0.05$ was chosen. The Bhattacharyya distance was calculated on the corresponding $16 \times 16 \times 16$ histograms, which were also used to identify non matching colors. At each iteration, a BTF between the two images was calculated with respect to the pixels that were not discarded in each image at previous iteration steps. This BTF was used to map the second image back to the conditions of the first one. Then the isolated colors are identified using the $16 \times 16 \times 16$ histograms: A color (due to the histogram quantization, a whole color range) was defined isolated by comparing the histogram bins. A bin with a five times higher count as in the corresponding bin was classified as isolated and therefore the corresponding colors were masked in the images. In most cases, the algorithm stopped after 2-4 iterations.
When a different amount of pixels got masked, randomly chosen pixels were masked in the image with less removals. Otherwise a less accurate BTF would be estimated. Since roughly the same scene was observed (the cameras moved slightly between and during the sequences), the same amount of pixels needed to be masked in each image. Since no pixel correspondences were known, the only possibility was to mask out pixels randomly.

In Figure 4.10 an iteration process is shown. The iteration of the automatically chosen reference frame is shown in (a)-(c). The iteration of the new frame, obtained at another point in time, is presented in (d)-(f). It can be seen that the algorithm quickly discards colors that are not present in both frames (Note: the images show the unprocessed frame, i.e. no BTF is applied). In the lower row, it can be seen that the red shirts (bottom right area of the image) get masked out quickly, since no red correspondence can be found in the reference image. Also, the yellow vests in the upper rows tend to disappear very fast. Since the masking process removed more pixels in the new frame, the pixels that were randomly masked out can be well seen in 4.10(c).

A comparison between a new frame, the adjusted new frame and the reference frame can be seen in Figure 4.11. The examples were taken from camera 1. Subjectively judged, the colors in (b) look much more similar to (c) than those in (a). This impression may come from the the global adjustment and the better matching of the colors of the floor and the walls. It is hard to tell whether the foreground regions also

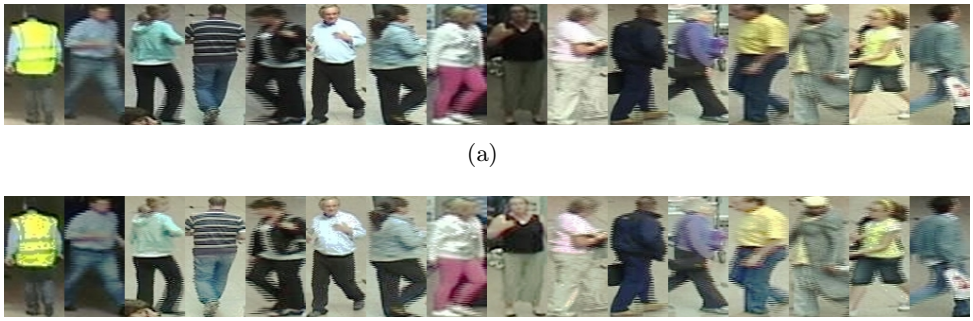
(a) Iteration 0 (b) Iteration 1 (c) Iteration 2

(d) Iteration 0 (e) Iteration 1 (f) Iteration 2

Figure 4.10: Masking process during the iterative BTF. (a)-(c) Reference frame iterations, (d)-(f) New frame iterations

(a) New frame (b) Adjusted (c) Reference

Figure 4.11: Results of the iterative BTF pre-processing. (a) The new frame under a changed global illumination (b) the adjusted frame using the iterative BTF (c) the reference frame (automatically chosen)



(a)

(b)

Figure 4.12: A few examples taken from camera 1 (a) before applying the iBTF (b) after applying the iBTF

got adapted well, because different persons were observed in the images. Figure 4.12 shows a few observation examples taken from camera 1. The first row shows the initial examples, the second row the examples after applying the iterative BTF. Again, it is hard to tell how well the foreground regions got adapted. On the other hand, the 5^{th} and 9^{th} example show well how the algorithm recovers an overall fairly dark illumination.

4.4.2 Improvement due to the iterative BTF

In this subsection, recognition accuracy is evaluated when employing the iterative BTF intra-camera-wise in the system. The evaluation setting is the same as in Section 4.3, i.e. the matching is done for camera pairs 1-2, 1-3 and 1-4. Only the sliced histograms in 3d RGB space are used, because they showed the best overall matching performance in the last section. For each camera separately, the iterative BTF is used to adjust all frames to an automatically selected reference frame. The advantages

of this method are evaluated in this section. The reference frame was automatically selected, according to the process described in Section 3.4.1, Chapter 3.

A total of four different cases are investigated (note that 1. and 2. were already evaluated in the previous section):

1. **No color processing**: Within the whole matching process, no color adaptation is used.
2. **CBTF**: The CBTF is used to establish a color mapping between the cameras.
3. **iterative BTF (pre)**: The iterative BTF is used to adjust all images in a camera to one reference frame. No mapping between cameras is established.
4. **iterative BTF (pre) + CBTF**: Firstly, the iterative BTF is used to adjust all images in a camera to one reference frame. Secondly, a CBTF is calculated between the camera views (on the adjusted images).

Matching results for camera pair 1-2 are shown in Figure 4.13. Recall that the cameras observe the same scene from a very similar view. Nevertheless, results improve when using the intra-camera iterative BTF without any inter-camera transfer function between the views. The reason for that is because the effects of changing illumination over time are decreased. Since no spatio-temporal information is used, it can happen that a person observed at a certain point in time matches well with another person that was observed much later in time, due to the change of overall illumination. This happens because the appearances of the persons are not adapted to the new illumination condition. When using the iterative BTF, the appearances are mapped to a common color appearance (defined by the reference image) and therefore do not match any more.
It can be seen, that no color processing at all performs better than when using the CBTF. On the other hand, the CBTF improves results drastically, when the effects of the temporal illumination are decreased. The most obvious explanation is the following: The CBTF uses a fixed training phase, therefore it produces erroneous mappings when the overall illumination conditions change in one view. As a consequence, it may be better not to adapt colors at all. Nevertheless, the CBTF is able to produce fairly correct color mappings, when the illumination conditions stay fixed. As a result, also smaller changes in color are adapted well. A few of the matching results are shown in Figure 4.14 (note that only the iterative BTF has been applied to the images). The first column shows the person in camera 1. The following columns are the matching results, sorted by their distance. Notice that only matches from rank 1 to 22 are shown.

Recognition performance for camera pair 1-3 is shown in Figure 4.15. It is not surprising that there is only a small gain in accuracy when just using the intra-camera iterative BTF to adjust all frames to a reference frame and not using any inter-camera CBTF. The color mismatch due to the bluish image is still present when all frames are

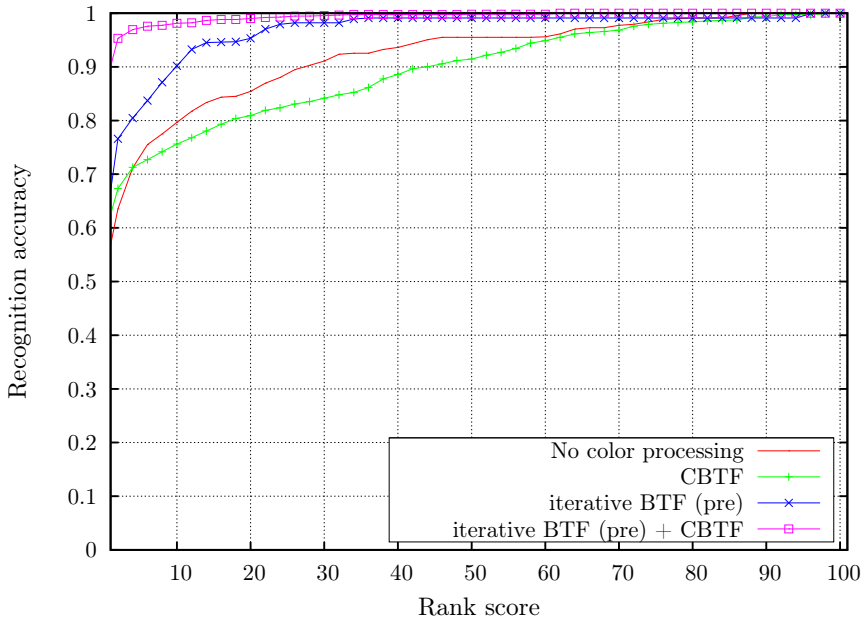


Figure 4.13: Improvements due to iterative BTF for camera pair 1-2

adjusted to one, also bluish, frame in camera 3. Therefore, the improvements are not as significant as in the other views. When additionally using the inter-camera CBTF, there is an obvious gain in accuracy. However, the gain is smaller than in the other tested camera pairs. One may think of the following reasons: The dynamic range of the color channels is not used very well (since the camera shot through a blue window). Hence, only a small portion (about a third) of the possible red and green brightness values were used. Even the blue channel was not fully used (only about 70%). Results are a dark image. Now, when mapping the color between the cameras, especially the red and the green channel need to be amplified between 100-200%, in order to match the brightness values in camera 1, which makes good use of the possible color range. In this case, a problem may be the boost of image noise in the dark regions of the image.

In Figure 4.16, the re-identification performance for camera pair 1-4 is printed. Similar findings as in the first case between cameras 1 and 2 can be made. Just pre-processing the image already improves results, because the person descriptors are more robust due to the compensation of illumination changes that happen over time. The additional usage of a CBTF between cameras improves the results further.

In order to make use of the iterative BTF in real world scenarios, no supervised training phase is needed. A representative amount of unlabeled camera frames can be easily obtained by just capturing data for a day or two. The algorithm will auto-

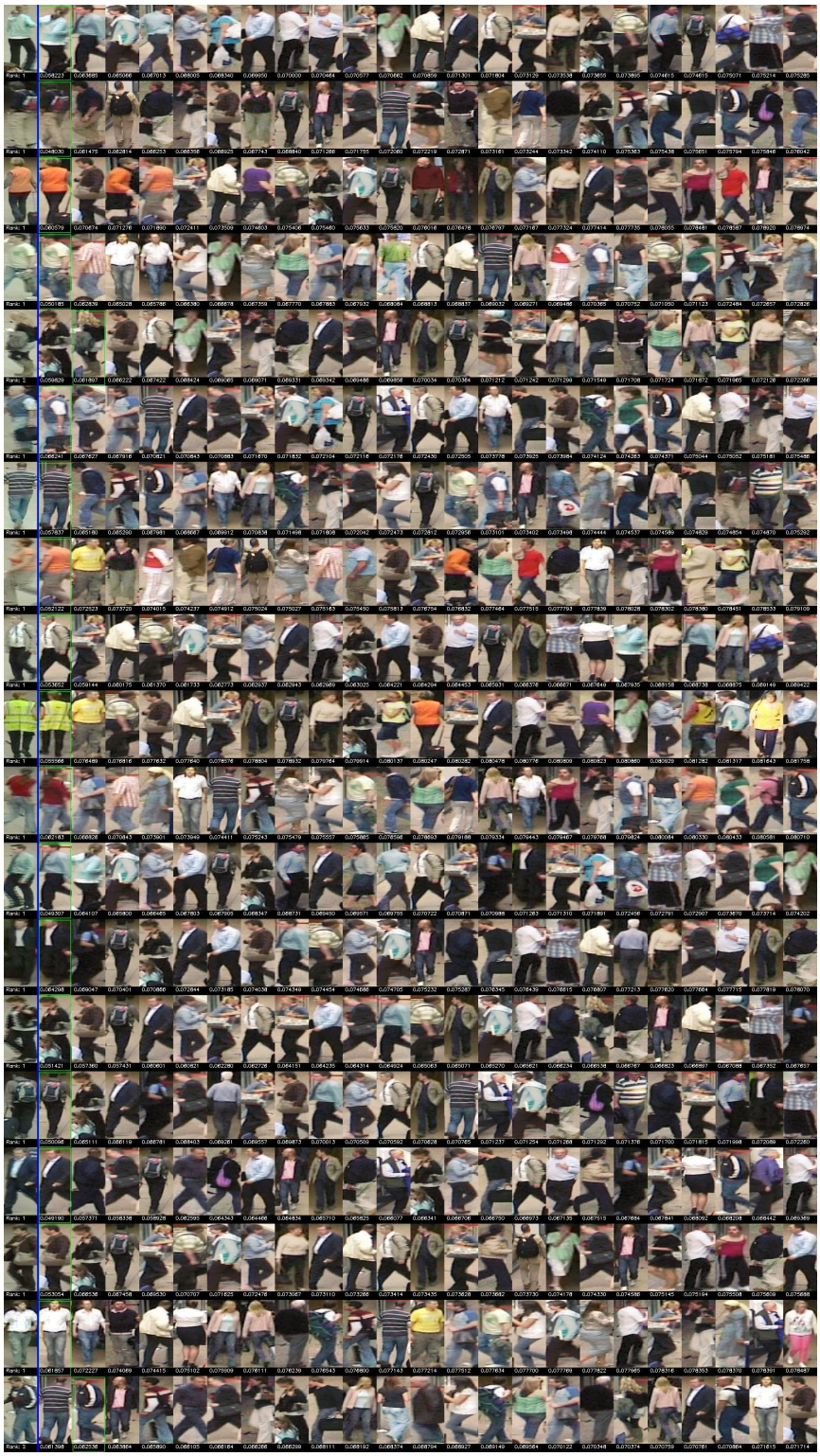


Figure 4.14: Matching examples for camera pair 1-2

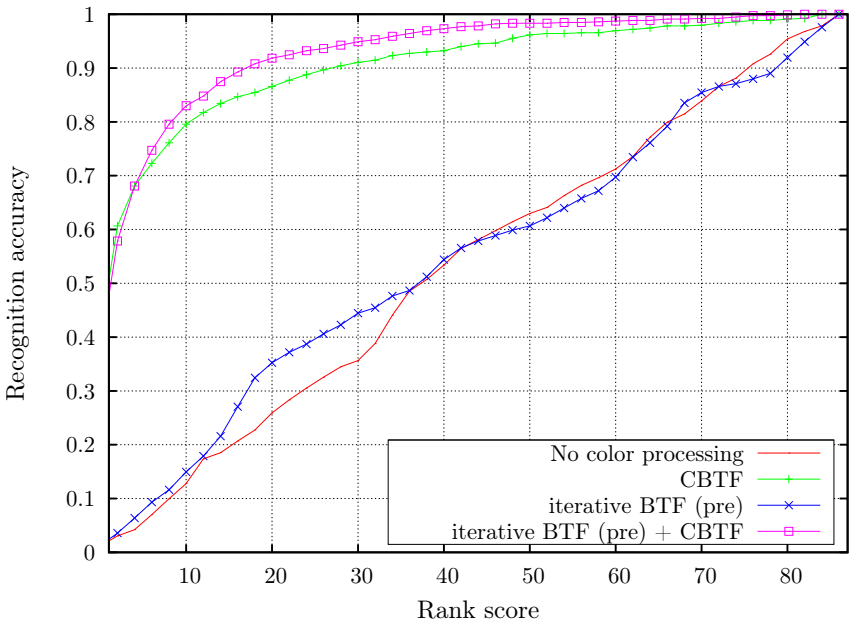


Figure 4.15: Improvements due to iterative BTF for camera pair 1-3

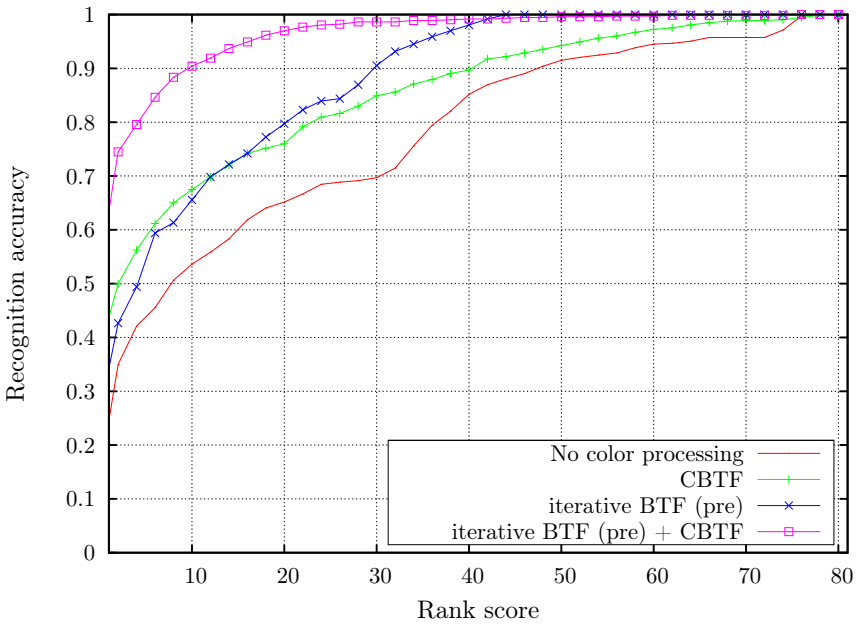


Figure 4.16: Improvements due to iterative BTF for camera pair 1-4

matically select a reference frame and then adjust each new incoming frame to that reference. Therefore a fixed training phase can be employed. The color transfer between cameras is neither depended on re-training nor on adaption over time, which could cause the BTF to drift.

4.4.3 Mean and standard deviation analysis in crossvalidation

An important topic that needs to be addressed is the change of the BTF due to the pre-processing using the iterative BTF. Javed et al. state in [3] that all BTF lie in a low dimensional subspace - under the assumption of constant illumination. In the previous experiments a cross validation approach was used, hence at each fold one inter-camera CBTF was calculated. It is interesting to investigate how the mean and standard deviation of these CBTFs change when pre-processing is employed.

At first, the mean and standard deviation for camera pair 1-2 and 1-4 are investigated (the results are fairly similar). In these cases, the cross validation had 10 and 8 folds, respectively. In Figure 4.17 and Figure 4.19, the results for each brightness channel are printed. Two obvious remarks can be made: The mean of all CBTFs is very different when using pre-processing and the standard deviation gets much smaller. According to [3], the BTF all lie in a low dimensional subspace if illumination is constant. Therefore, the standard deviation is large when not using any pre-processing, since the training examples in each fold come from different illumination conditions. On the other hand, this is another proof that the iterative BTF pre-processing compensates for illumination changes, since the standard deviation gets much smaller. It should be noted that even with a perfect illumination compensation, the standard deviation will never be zero. Reasons are for example the small errors introduced by the BTF and the appearance of persons in different poses in the cameras. The means of the CBTFs are likely to follow different curves, since the pre-processing tries to adjust the overall image to a reference frame. Therefore it maps all images to a common color appearance (defined by the reference frame) and from there on the inter-camera transfer is calculated.

At second, the mean and standard deviation for camera pair 1-3 are investigated (Figure 4.18). This setting should be considered separately, because the iterative BTF did only show a small improvement in comparison to camera pairs 1&2 and 1&4. It can be seen that while the means drastically differ, the standard deviations are much more similar than in the previous examples. Regarding the average standard deviation over all brightness values with and without pre-processing, they are close to identical. This may be an evidence why the results did not improve as drastically. An explanation for the observed behavior may be most likely the bad utilization of the dynamic range. There is only little variation in the images over time due to illumination (all images are bluish and fairly dark). Only a few frames exist where the situation is a little bit better. The unsupervised pre-processing step automatically

chooses one of these frames and then adjusts all the other frames. But still, because of the bad usage of the dynamic range in the red and green channel, there is only little information contained. The result of this problem can be seen in Figure 4.18(a) and (b). The upper brightness range shows a large variance, because of the lack of brightness values that actually can be mapped. Therefore each time a CBTF is calculated, the small errors within the CBTF results in an overall big variance in the regions, where only little information is present.

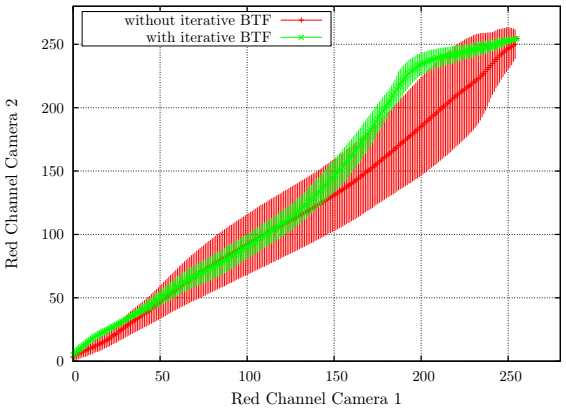
4.4.4 Number of examples

When wanting to obtain a BTF between cameras, it is not clear how many training examples should be used. According to Javed et al. in [3], the BTF can be described using a low dimensional subspace with few parameters, because under constant illumination, all BTFs from single persons will look similar. On the other hand, this would mean that when given a constant illumination, only a few training examples would be enough to train a representative BTF.

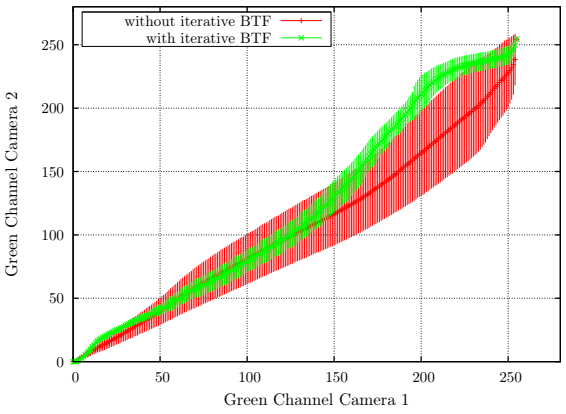
In previous examples, 10 training observations were used in combination with cross validation. Now, within these 10 examples, the results are investigated when discarding the last 2, 4, 6 or 8 of them. Again, the following plots are gained using the 3d histograms with 10 splits in vertical direction. In Figure 4.20(a)-(c), the findings for camera pairs 1-2, 1-3 and 1-4 are printed. It can be seen that accuracy quickly reaches its maximum when using more than 4 examples. On the other hand, it does not improve significantly when using more than about 6 examples. This is not surprising, when comparing these results with Javed et al. in [3]. In their work they state that between 5 to 7 principle components account for 99% of the variance when using probabilistic PCA. Under the assumption that the picked training examples in the experiments already have a large variance in appearance, they already define the CBTF fairly well. Consequently, there is neither a need for more training examples, nor would it improve precision. Additionally, it can be seen that in camera pair 1-2 only few examples are necessary to train the CBTF, since colors are already fairly similar. Camera pair 1-3 shows a very significant mismatch and it seems that more examples are required.

4.4.5 Comparison between MBTF and CBTF

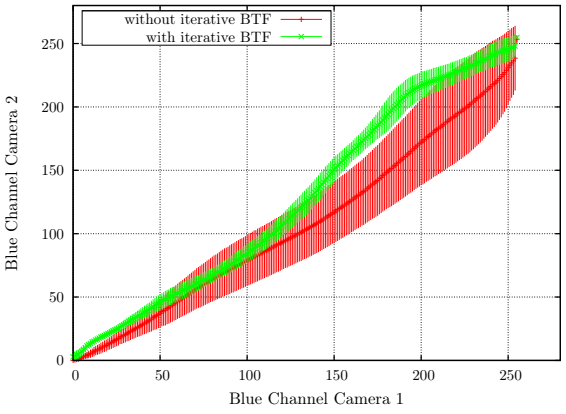
Until now, the differences between the CBTF and the MBTF have not been investigated. Prosser et al. did a direct comparison of both functions in [5]. They state that the CBTF makes better use of the color information that is provided in the training set. A simple example can point out the differences between the MBTF and the CBTF. Given the observations of three different persons at each camera site, three BTFs can be calculated. For the sake of simplicity, only the red channel is analyzed. In Figure 4.21 the Brightness Transfer Function (red channel) for each person and



(a)

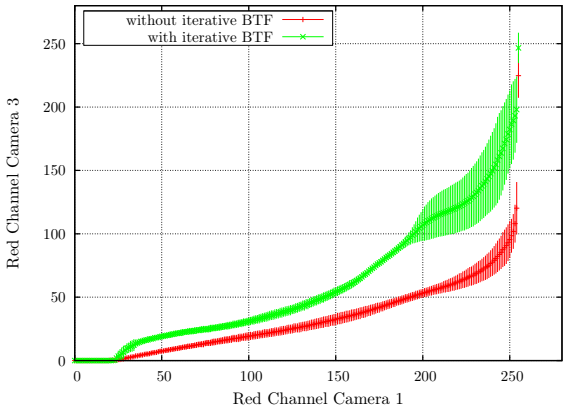


(b)

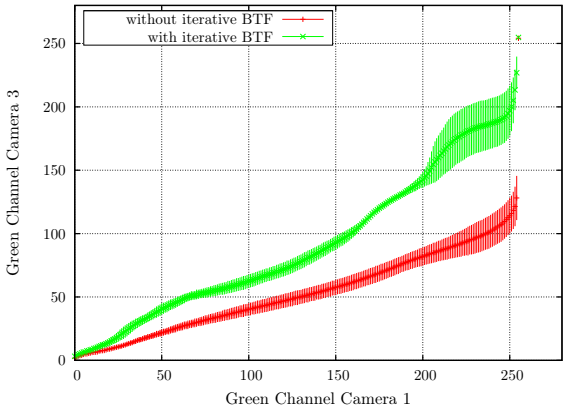


(c)

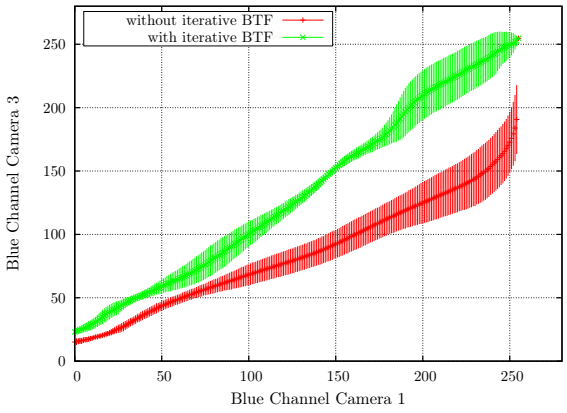
Figure 4.17: Mean and standard deviation of CBTFs for camera pair 1-2 (a) red (b) green (c) blue channel



(a)

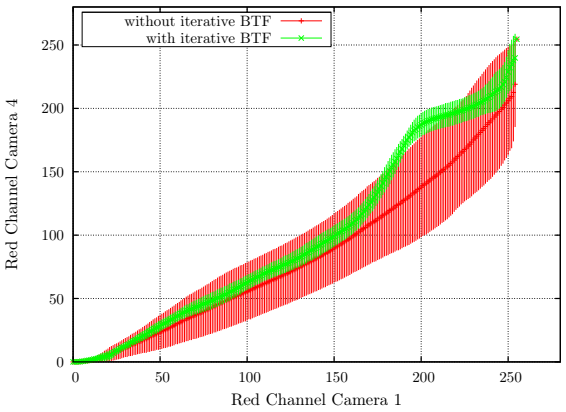


(b)

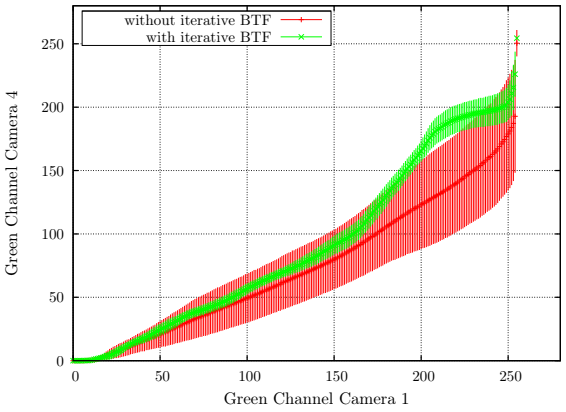


(c)

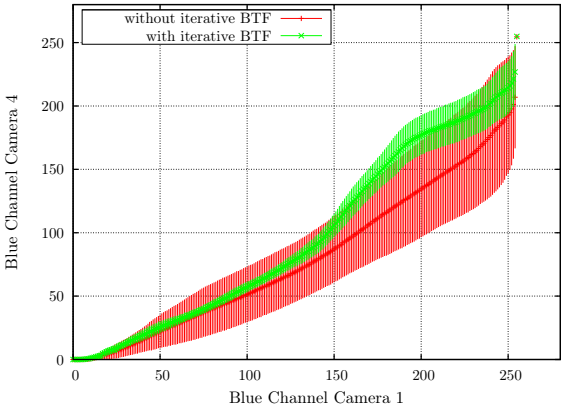
Figure 4.18: Mean and standard deviation of CBTFs for camera pair 1-3 (a) red (b) green (c) blue channel



(a)

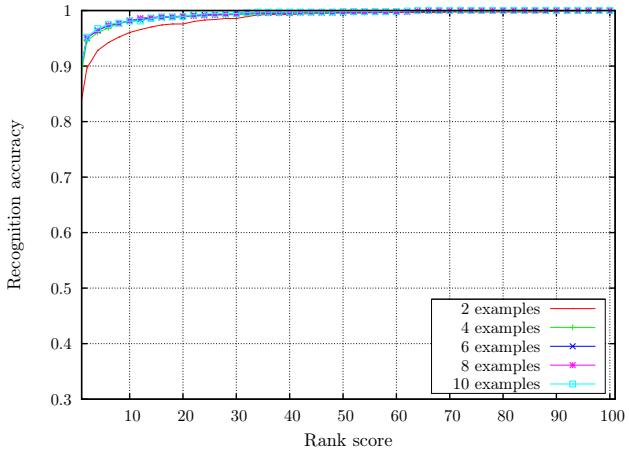


(b)

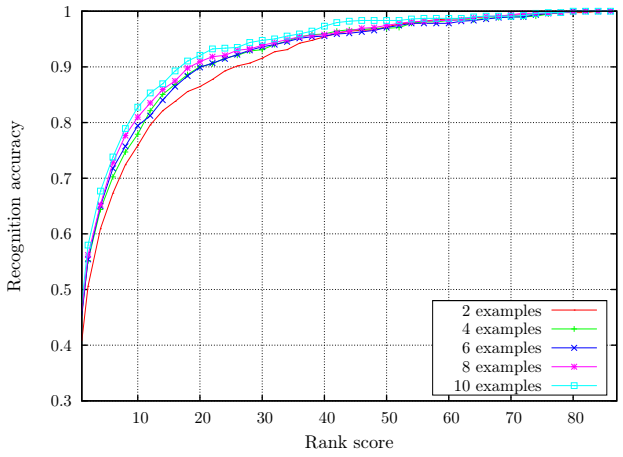


(c)

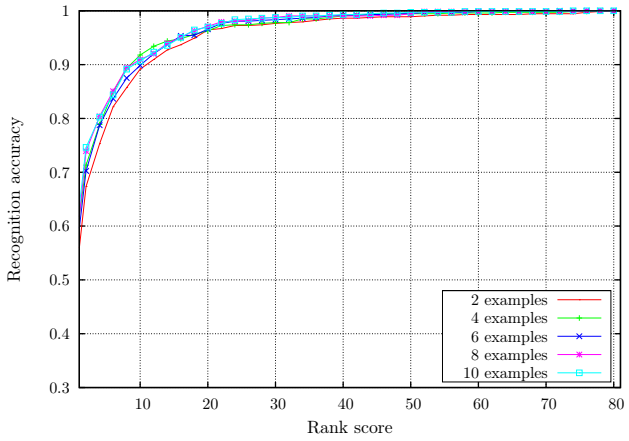
Figure 4.19: Mean and standard deviation of CBTFs for camera pair 1-4 (a) red (b) green (c) blue channel



(a)



(b)



(c)

Figure 4.20: How the number of examples affects recognition in camera pair (a) 1-2 (b) 1-3 (c) 1-4

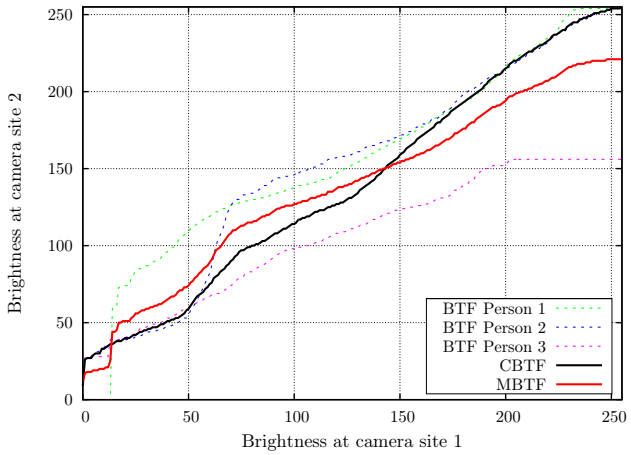


Figure 4.21: Brightness transfer functions for several persons and the resulting MBTF and CBTF

the resulting MBTF and CBTF are displayed. The corresponding examples that were used can be seen in figure 4.22(a) and 4.22(b).
Figure 4.21 reveals that person 1 and person 2 occupy the upper brightness ranges between 200 and 255 in both views, while person 3 does not provide any information in this range at all. The response for person 3 in the upper range (> 200) is therefore flat (indicating that no bright pixels were present in view 2). The responses for person 1 and 2 follow a rather linear response in the upper range. Here, the differences between the MBTF and the CBTF can be seen fairly well: The CBTF only uses examples 1 and 2 for calculation in the upper range while the MBTF takes all 3 examples, even if the third example does not provide any useful information. As a consequence, the CBTF maps the bright values of person 1, 2 and 3 correctly, while the mean mapping of the MBTF does not reflect the reality for any of the three examples. A similar finding is made in [5]. On the other hand, it must be said that it is not clear how much this affects the color transfer if more training examples are used. The color transformed observations of the examples from camera site 2 in Figure 4.22(b) are displayed for the CBTF in Figure 4.22(c) and for the MBTF in Figure 4.22(d).

The influences of this behavior are not clear for a bigger scenario. Therefore, a comparison between the MBTF and the CBTF is carried out and presented in Figure 4.23. Surprisingly, the MBTF performs only slightly worse than the CBTF and not as drastically as one may think due to the theoretical analysis. The findings therefore do not match those of by Prosser et al. in [5]. Reasons may be the small and maybe unrepresentative data set that they used in their experiments.

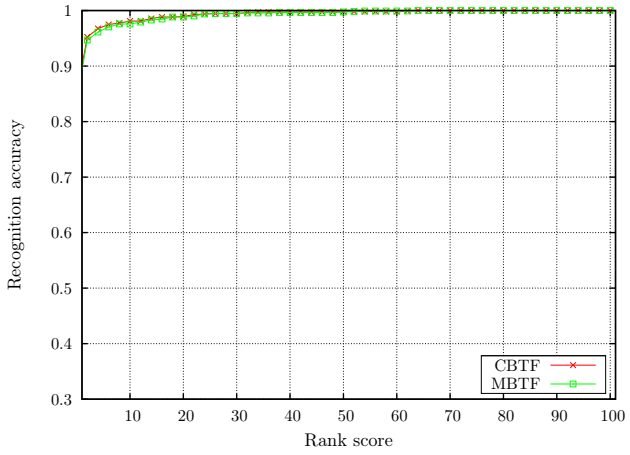
(a) Persons at site 1 (b) Persons at site 2

(c) CBTF applied (d) MBTF applied

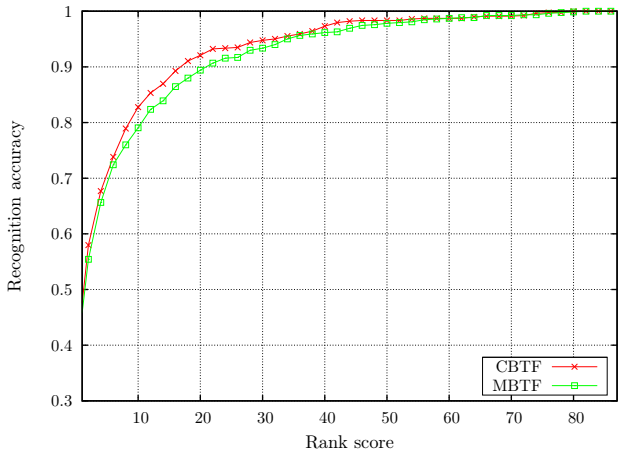
Figure 4.22: Comparison examples between CBTF and MBTF

4.4.6 Summary

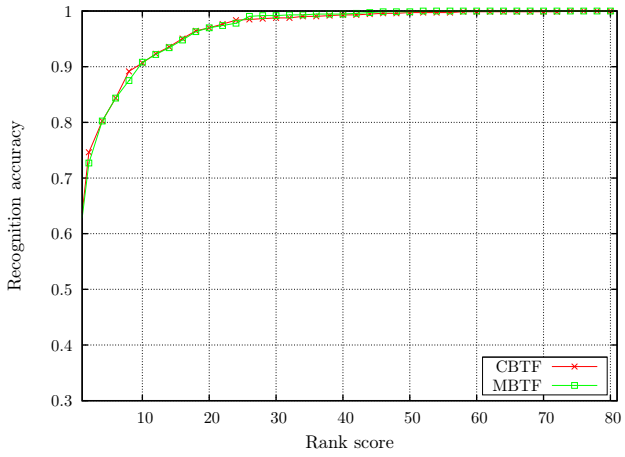
Changing illumination over time is a big problem in surveillance scenarios. Color descriptors, such as histograms, and Brightness Transfer Functions are very sensitive to these changes and tend to produce erroneous results when conditions change. An iterative BTF was proposed to compensate the lighting change over time. An automatically selected frame from unlabeled video data was used as a reference frame and new incoming frames were adjusted to the illumination condition given in that frame. Therefore, a fixed training stage for the BTF could be used. A further analysis showed a smaller standard deviation of the obtained CBTF in the folds of the cross validation, when using the proposed algorithm. This could be taken as further evidence that the effects of illumination over time were decreased. Further experiments showed that the inter-camera CBTF can be trained with a only 6 to 10 examples of labeled correspondences. Due to the pre-processing using the iterative BTF, this is a very reasonable amount, because no re-training is required when illumination conditions change.



(a)



(b)



(c)

Figure 4.23: Comparison between MBTF and CBTF in camera pair (a) 1-2 (b) 1-3 (c) 1-4

4.5 Evaluation of the Split Histograms

4.5.1 Recognition accuracy per histogram slice

When dividing the bounding box into several vertical slices, it is interesting to gain information about the discriminative power each single slice has. In order to get a rating of each slice, the whole recognition accuracy measurement is done with only one single slice. The factor that is used in all other experiments to weight the separate slices is not used in this experiment.

When analyzing the human body, a separation between head, upper body and legs can be made. A common assumption is that the ratio between those body parts is a 2:4:4 or a 2:3:5 ratio. In this work, it is assumed that the second ratio reflects reality. Therefore, the body parts relate to the slices as follows:

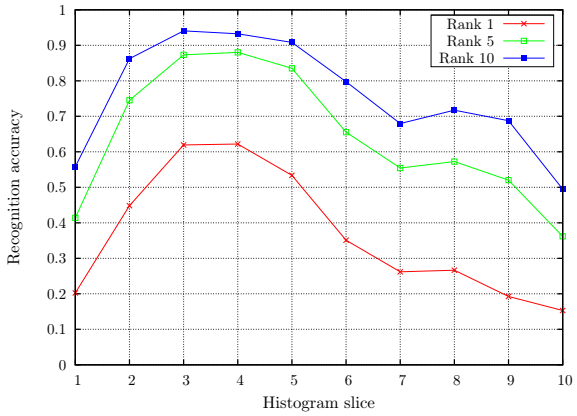
- Head: Slices 1, 2
- Upper body: Slices 3, 4, 5
- Legs: Slices 6, 7, 8, 9, 10

Results for recognition accuracy per slice can be seen in Figure 4.24. The accuracy on the y-axis is printed over the separate histogram splits on the x-axis. All three plots reveal the same insight: The upper body has most discriminative power for the person matching problem. The upper part of the head has nearly no relevance. It could happen that some parts of the lower head region also showed some parts of the upper body, therefore providing more information than its top counterpart. This may explain the better precision of the second slice. The legs (slice 6-10) also provide little information. The reasons may be twofold: Subjectively judging, the legs showed more segmentation errors, especially due to shadows. On the other hand, it could be possible that the legs show a smaller variance in colors than the upper body part. Probably, people tend to wear less colorful pants. Common colors for pants are blue, black, white and gray and it is not likely to observe red or green pants. In contrast, the variety of colors of upper body clothing is much bigger: In the used data set a broad spectrum of colors with white, black, yellow, blue, green, red, purple, pink, etc. exists.

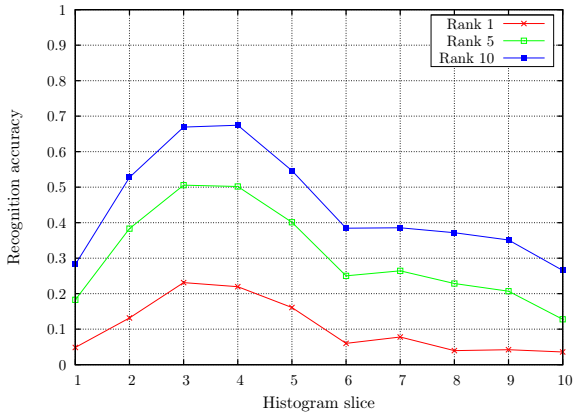
4.5.2 Recognition accuracy under occlusion

Now, since it is known how much each slice provides in terms of discriminative power, it can be investigated how accuracy changes when using only a subset of the slices. Therefore three cases are considered:

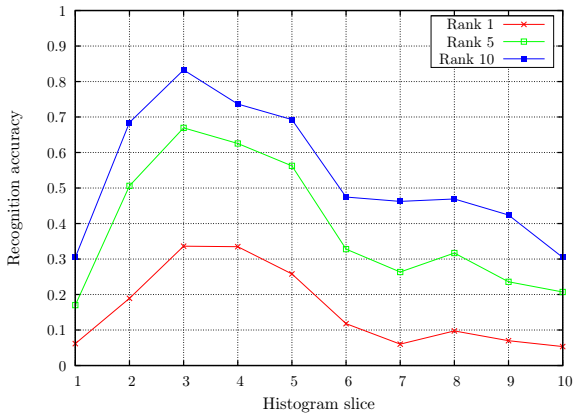
- Full body: Slices 1-10 (for comparison)
- Upper body: Slices 3-5



(a)



(b)



(c)

Figure 4.24: Recognition accuracy for every separate histogram slice for camera pair (a) 1-2 (b) 1-3 (c) 1-4

- Full body without head and lower legs: Slices 3-8

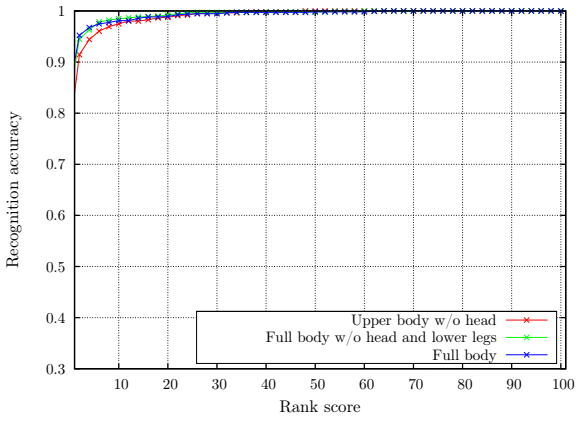
The results for the cases in the given multi-camera scenario are presented in Figure 4.25. The same observations can be made for all three camera pairs. Even if the head and the lower leg region provide only little information, they still generate a small gain in accuracy. It is not surprising that recognition performance decreases when only using the upper body. On the other hand, performance does not decrease significantly. This was not unsuspected since Figure 4.24 revealed that most information lies in the upper body region. A common problem in real world surveillance scenarios is the occlusion of the legs. Under the assumption that occlusion can be detected, it is interesting to know that matching of occluded persons is still possible without a too large loss in accuracy.

4.5.3 Summary

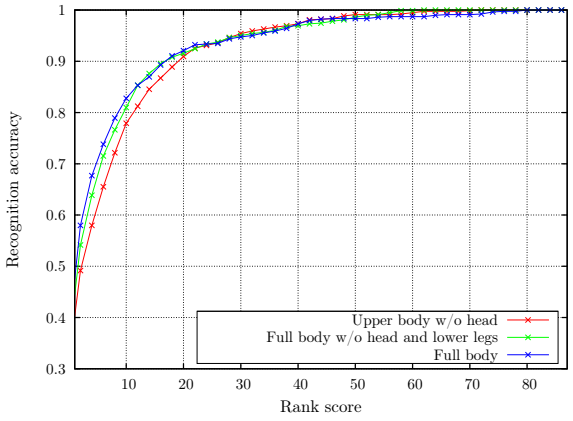
Experiments showed that the upper body region provides the most discriminative power, while the overall accuracy is still best when employing all the information given. The legs showed a low discriminative power. This may be due to two reasons: Error prone segmentation because of shadows and a low variance of colors for pants. In real world scenarios, occlusion often occurs and therefore it is essential to know that persons can still be identified fairly well without having the color information that is provided by the legs.

4.6 Bi-directional Matching

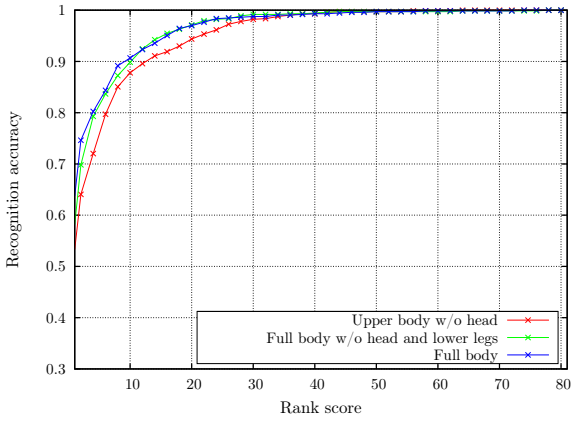
Until now, only the matching from camera 1 to 2, 3, 4 has been investigated. In this final experiment, the matching performance for all possible camera combinations is tested. It is not clear if a good matching in one direction also works well in the other direction. Recall that until now, color was always adapted to camera 1.
Therefore, the accuracies for all possible camera pairs are shown in Figure 4.26 and 4.27. Inverse matching directions are plotted in the same graph. The observations are similar to those made by Prosser et al. in [5] where a bi-directional matching was investigated. The swap of the matching direction, i.e. instead of adapting the color in camera 2 to the color in camera 1, the color is adapted from 1 to 2, does provide very similar results. However, it may not produce the same ranking of targets. Prosser et al. evaluated how to fuse the information of both directions in order to boost performance, but the proposed schemes only showed very small differences. The proposed techniques were also tested in this work, but did only show very minor changes in accuracy over just choosing one arbitrary direction in order to perform the matching. Still, there may exist other possibilities for fusing the results of both directions and by doing so, further improvements in the results may be achieved.



(a)

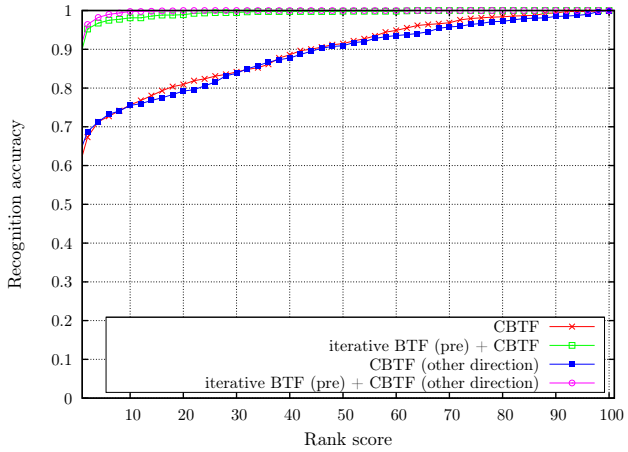


(b)

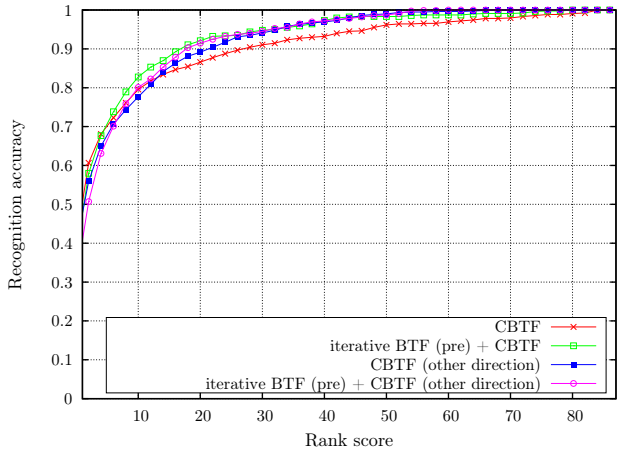


(c)

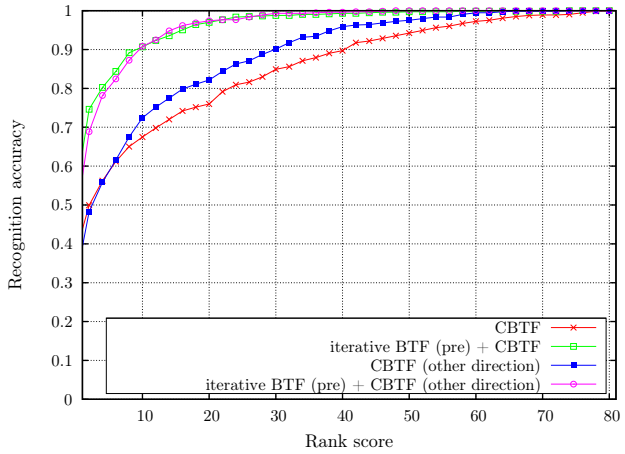
Figure 4.25: Recognition accuracy when using only a certain body part in camera pair (a) 1-2 (b) 1-3 (c) 1-4



(a)

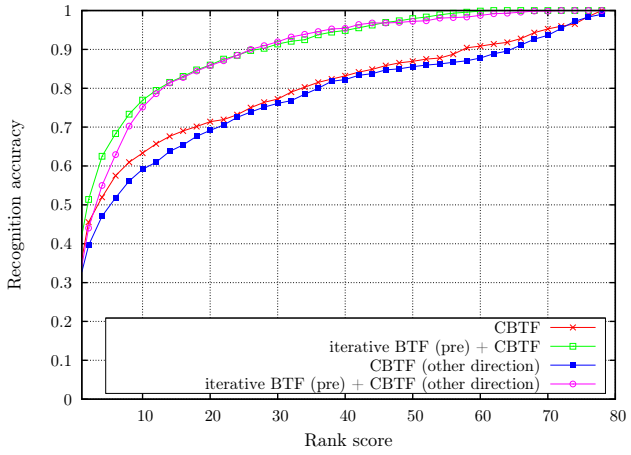


(b)

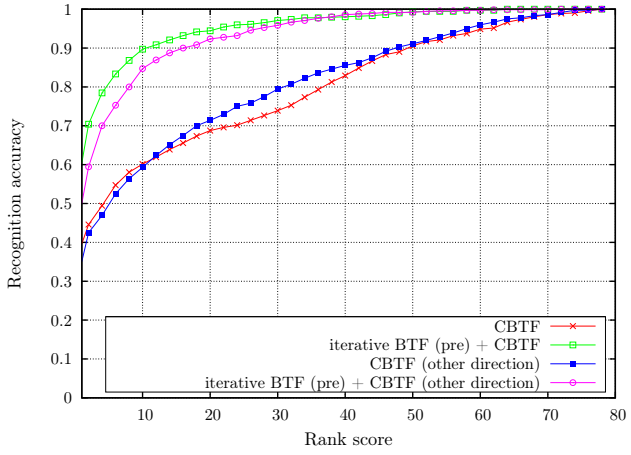


(c)

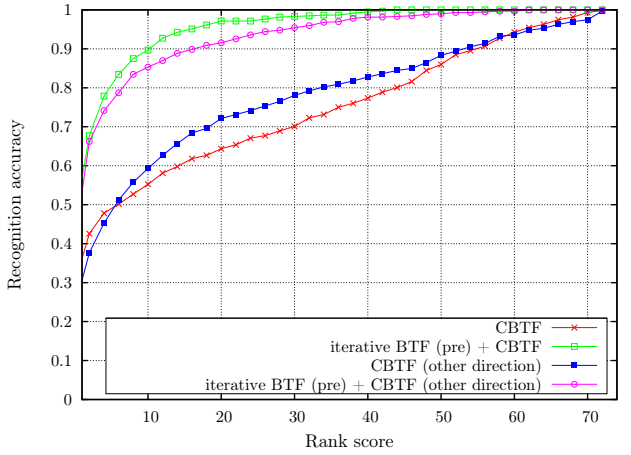
Figure 4.26: Recognition accuracies among camera pairs (a) 1-2 (b) 1-3 (c) 1-4



(a)



(b)



(c)

Figure 4.27: Recognition accuracies among camera pairs (a) 2-3 (b) 2-4 (c) 3-4

4.6.1 Summary

Using a bi-directional matching between cameras may be able to improve results even further. However, it did only show very minor changes on the given data. Therefore, randomly picking one matching direction did not bring any noticeable disadvantages in most cases.

5 Conclusion

In this thesis the person re-identification problem in a multi-camera network has been addressed. The main focus was put on the adaption of colors between cameras. In surveillance scenarios, the different cameras are usually distributed across a wide area and therefore do not share common illumination conditions. As a consequence, color based person recognition suffers from the different appearances of colors.

The use of Brightness Transfer Functions can compensate the color mismatch between camera sites and therefore improve results. Without any reference objects, it is difficult to measure the mapping error other than just in terms of recognition accuracy. Several experiments have been conducted in this work that show that the BTF will most likely produce a less accurate mapping when colors are lying close to black or white. The BTF maps each color channel separately, which is the reason why it does not necessarily map colors well between (r, g ,b) color pairs. It has been investigated and revealed that in combination with histograms in $R \times G \times B$ space, the BTF clearly outperforms 1d histograms. On the one hand, a coarser quantization of the histogram bins can compensate a less accurate mapping between (r, g ,b) color pairs. On the other hand, a very coarse quantization will represent the appearance of a person less accurately. Therefore a trade-off between the quantization and the mapping error has to be considered. Dividing the histograms into vertical slices improved results further as it captures the spatial distribution of colors.
One drawback of the BTF is the employment of a fixed training phase. While the mapping performs well under the training conditions, it decreases as soon as illumination conditions at one camera site change. This change is very likely to happen in real world scenarios, especially when the scene is illuminated by ambient light. Manual re-training is impractical and adaptive functions can drift. The fixed training stage of the BTF and the changing illumination over time were addressed here. An approach for compensating the temporal change of lighting has been proposed. A reference frame is chosen automatically and colors of each new incoming frame are adjusted to this pre-selected frame. This step takes place in a completely unsupervised fashion. Since illumination is rendered more invariant, it is then possible to make use of a fixed training phase.
Cross validation was used to show how the standard deviation of the BTF between cameras decreases when employing the proposed method. This can be seen as a proof that illumination was rendered more constant.

All the experiments were carried out on a real world data set, collected at an airport and the matching between cameras was done with between 80 to 101 persons. The data showed significant changes in illumination over time and was therefore a well suited data set for evaluating the pre-processing step. A total of 12 different camera pairs were considered and the proposed algorithm showed an average improvement of approx. 12% in recognition accuracy for Rank 1 over just using the CBTF (36% vs. 48% recognition accuracy). For Rank 10, an average improvement of approx. 20% was measured (64% vs. 84% recognition accuracy).

5.1 Further Work

A variety of possible enhancements exists, which could further improve results. Two examples are the incorporation of spatio-temporal information [4] or the boosting of features [18]. Speaking more generally, each processing step in Chapter 3 could be improved and hopefully provide a more robust matching. But focusing more on the aspect of color adaption inter- and intra-camera-wise, a couple of points and ideas should be mentioned:

Evaluation of the BTF. So far, the performance of the BTF has only been measured via the matching problem. In order to reveal how well the BTF actually maps each brightness channel, a reference chart could be used. This chart could show red, green and blue rectangles with different, known intensities. After training a BTF between the cameras, it could be used to estimate the mapping error per channel by placing the chart in the field-of-view of each camera.
Another important point would be to measure the mapping performance between $R \times G \times B$ color pairs. Again, a color chart could be used to estimate the error. An example of such a color chart can be found in [2].

Iterative BTF. The iterative BTF introduced in this work showed an overall improvement in recognition performance. Still, there exist several possibilities that may improve its accuracy.
The detection of the same colors in each image was done using histograms. If images perfectly overlay, this step would be obsolete, but on the other hand it still could be used in order to avoid a background segmentation step. It would be possible to detect the same colors by applying an online approximation of the K-Means algorithm, such as in [12]. Then, clusters could be compared and removed if necessary. This would give a much finer granularity than the coarsely quantized histograms.
The selection of the reference frame could also be improved. Especially if images perfectly overlap, instead of choosing one single frame, a median frame could be calculated and chosen as reference. More likely, the mapping of a new frame back to the median frame would include less many-to-one or one-to-many mappings. This would be due to the reason that most of the colors in the reference

frame would be lying somewhere in the middle of the dynamic range of the camera and not at the outer ends.

Color mapping. One major drawback of the BTF is the separate mapping of each color channel, because it does not necessarily lead to a correct mapping between pairs in $R \times G \times B$ space. A BTF in a 3-dimensional space could improve results even further. On the other hand, it is a very difficult problem to establish such a mapping without any given pixel or region references. It is not clear whether it is solvable at all, because of its ill-posed nature.

Acknowledgments

This work would not have been possible without the help of the following people:
Rainer Stiefelhagen, thank you for giving me the possibility to work at your institute and doing interesting research. Every single person in the CVHCI group, thank you all for your patience, comments and inspiration you gave me within the last two years. Special thanks to Keni Bernardin for supervising all my work - you were a great mentor. Thank you. Sebastian Geiger and Simon Friedberger, thank you for proof-reading.

Prof. Dr. Alex Waibel and the whole interACT advisory board, thank you for giving me the possibility to study aboard at Carnegie Mellon in Pittsburgh, PA.
It was an awesome time in Pittsburgh and I am remembering every single day of this wonderful experience. Kelly, Kristie and Mark - thank you for the great time, without you I may not have felt home. Stefan, thanks for introducing me to Pittsburgh within the first weeks, the great time in Toronto and at GD. Simon, thanks for the great time in both of the labs and being a good listener whenever I was stuck and desperate! Thomas, it has been a great help to have you around - and I mean that in every possible way. Thank you!

This work would not have been possible without the support of my friends and my family. I would like to express my deepest gratitude to my mother, father and sister. Thank you for supporting me throughout my entire life, accepting the decisions I made and showing me how much you love me.
I love you.

Bibliography

[1] A. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[2] A. Ilie and G. Welch. Ensuring color consistency across multiple cameras. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2:1268–1275, 2005.

[3] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 26–33, 2005.

[4] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162, 2008.

[5] B. Prosser, S.G. Gong, and T. Xiang. Multi-camera matching using bi-directional cumulative brightness transfer functions. In *Proceedings of the British Machine Vision Conference*, 2008.

[6] T. D'Orazio, P.L. Mazzeo, and P. Spagnolo. Color brightness transfer function evaluation for non overlapping multi camera tracking. In *Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras*, 2009.

[7] B. Prosser, S.G. Gong, and T. Xiang. Multi-camera matching under illumination change over time. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[8] K. Chen, C. Lai, Y. Hung, and C. Chen. An adaptive learning method for target tracking across multiple cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[9] Workshop on performance evaluation of tracking and surveillance. `http://www.cvg.rdg.ac.uk/PETS2007/data.html`, 2007.

[10] M.D. Grossberg and S.K. Nayar. Determining the camera response from images: What is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, 2003.

[11] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[12] C. Madden, E.D. Cheng, and M. Piccardi. Tracking people across disjoint camera views by an illumination-tolerant appearance representation. *Machine Vision Applications*, 18(3):233–247, 2007.

[13] A. Colombo, J. Orwell, and S.A. Velastin. Colour constancy techniques for re-recognition of pedestrians from multiple surveillance cameras. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[14] B.V. Funt. Color constancy in digital imagery. In *Proceedings of the International Conference on Image Processing*, pages 55–59, 1999.

[15] A. Gilbert and R. Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proceedings of the European Conference on Computer Vision*, pages II: 125–136, 2006.

[16] O. Hamdoun, F. Moutarde, B. Stanciulescu, and B. Steux. Interest points harvesting in video sequences for efficient person identification. In *Proceedings of the IEEE International Workshop on Visual Surveillance*, 2008.

[17] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[18] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Proceedings of the 10th European Conference on Computer Vision*, pages 262–275, 2008.

[19] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proceedings of the 10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

[20] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:663–671, 2006.

[21] U. Park, A.K. Jain, I. Kitahara, K. Kogure, and N. Hagita. Vise: Visual search engine using multiple networked cameras. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 1204–1207, 2006.

[22] V. Agarwal, B.R. Abidi, A.F. Koschan, and M.A. Abidi. An overview of color constancy algorithms. *Journal of Pattern Recognition Research*, 1(1):42–54, 2006.

[23] F.M. Porikli. Inter-camera color calibration by correlation model function. In *Proceedings of the IEEE International Conference on Image Processing*, pages 133–136, 2003.

[24] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, pages 82–98, 1999.

[25] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[26] H. Grabner, P.M. Roth, and H. Bischof. Is pedestrian detection really a hard task? In *Proceedings of the IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

[27] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human detection using partial least squares analysis. In *Proceedings of the International Conference on Computer Vision*, 2009.

[28] H. Wold. *Encyclopedia of Statistical Sciences*, volume 6, chapter Partial least squares, pages 581–591. Wiley-Interscience, 1985.

[29] W.R. Schwartz, R. Gopalan, R. Chellappa, and L.S. Davis. Robust human detection under occlusion by integrating face and person detectors. In *Proceedings of the Third International Conference on Advances in Biometrics*, pages 970–979, 2009.

[30] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 1999.

[31] Opencv (open source computer vision) library 2.0. `http://opencv.willowgarage.com/wiki/`.

[32] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. In *Proceedings of the 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.

[33] Z. Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition*, 2004.

[34] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172 – 185, 2005.

[35] A. Ilyas, M. Scuturici, and S. Miguet. Real time foreground-background segmentation using a modified codebook model. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 454–459, 2009.

[36] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[37] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, page 731, 1997.

[38] K. Bernardin, T. Gehrig, and R. Stiefelhagen. Multi-level particle filter fusion of features and cues for audio-visual person tracking. In *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR and RT*, 2008.

[39] O. Masoud and N.P. Papanikolopoulos. Robust pedestrian tracking using a model-based approach. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 338–343, 1997.

[40] S.T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1158–1163, 2005.