Camera based Human Localization and Recognition in Smart Environments

by

Henrik Siebo Peter Brauer

Thesis submitted in partial fulfilment of the requirements
of the University of the West of Scotland
for the award of Doctor of Philosophy

13th October 2014

Declaration

The research presented in this thesis was carried out by the undersigned. No part of the research has been submitted in support of an application for another degree or qualification at this or another university.

Signed: ____

Date: ____

Abstract

In recent years, the demand for sophisticated surveillance systems has risen significantly. A special case is indoor surveillance in smart environments such as smart homes or smart rooms. While general surveillance systems attempt to control access to resources or to detect individuals on a watch list, the aim of surveillance in smart environments is to capture human behaviour details in order to provide automated services. For example, when a person is watching TV, the smart home could automatically provide a list of preferred channels depending on who is watching TV.

This thesis investigates novel tracking, detection and verification algorithms in order to identify a person and to estimate his/her position, based on the last known position and his/her speed with improved real-time performance.

In particular a novel head detector is proposed that combines a human head, an upper-body and a body detector with a sigmoid function in order to derive a consensus decision. Based on this head detector a real-time framework for multi-target tracking is designed. This framework is first developed for perspective cameras and then later extended to fisheye cameras. Finally, two complete frameworks for video face recognition are presented. The first framework is based on visual tracking and Local Binary Pattern face recognition whereas the second relies on multi-target tracking, Local Quantized Pattern feature vectors and set-to-set similarity.

In order to evaluate the performance of the proposed approach a number of experiments were made on multiple datasets. Face recognition and face tracking has been evaluated on Honda/UCSD Video Database (100% accuracy), Smart Home Dataset (83.44% accuracy), YouTube Face Database (83.44% accuracy), YouTube Celebrities Dataset (84.07% accuracy) and the ChokePoint Dataset (100% accuracy). Face detection has been evaluated on the FDDB dataset and the AFW dataset (AP 95.32%). Head detection has been evaluated on the Town Centre Benchmark where an MR of 58% was achieved. Finally, multi-target tracking has been evaluated on the Town Centre Benchmark (MOTA 81.55%), Parking Lot Benchmark (MOTA 79.71%), Bomni-DB (MOTA 78.55%) and LivingPlace Fisheye Benchmark (MOTA 69.53 %).

Acknowledgements

I would like to thank the following people for their support, and professional guidance over the course of my PhD degree. Without their help and support, it would not have been possible for me to complete the course and this thesis.

First of all, I would like to thank my supervisors, Christos Grecos and Kai von Luck, for their guidance, patience and advice through the whole course of my research. Without their inspiration, support, advises, patience and their belief in my abilities over the years this thesis would not have been possible.

I would like to thank Qi Wang for becoming my new director of studies at such short notice and for his support and detailed reviews, comments and corrections for this thesis.

I specially thank Zita Schillmöller for her support and the organisation of the PhD program.

I would like to thank all my friends and colleagues at the Living Place Hamburg and in the PhD program from Hamburg University of Applied Sciences.

I want to thank my family and friends for their support and encouragement, especially, my father, my mother for their help and for their financial support. Without their support I would not have been able to complete the course and this thesis.

Finally, I would thank my girlfriend Julia Oelschläger who was always there to support me unconditionally.

List of Publications

- Henrik Brauer, Christos Grecos, Kai von Luck, "‘Combining Detectors for Robust Head Detection"’, *Irish Machine Vision and Image Processing Conference (IMVIP 2014)*, August 2014, Derry-Londonderry, Northern Ireland. **(Best Student Paper)**

- Henrik Brauer, Christos Grecos, Kai von Luck, "‘Robust False Positive Detection for Real-Time Multi-Target Tracking"’, *Image and Signal Processing*, Volume 8509 of *Lecture Notes in Computer Science*, pages 450-459. Springer International Publishing.

- Henrik Brauer, Christos Grecos, Kai von Luck, "‘Video-based Face Recognition with Whitened-PCA Matched Background Similarity"’, *European Workshop on Visual Information Processing (EUVIP 2013)*, June 2013, Paris France.

Contents

List of Figures

List of Tables

Abbreviations

3DMM	3D Morphable Model
AAM	Active Appearance Models
ACF	Aggregated Channel Features
AP	Average Precision
CLM	Constrained Local Model
DCT	Discrete Cosine Transform
EM	Expectation-Maximization
EVLBP	Extended Volume Local Binary Pattern
FFT	Fast Fourier Transform
FN	False Negatives
FP	False Negatives
FPPI	False Positives Per Image
GMM	Gaussian Mixture Model
GS	Generalized Similarity
HOG	Histograms of Oriented Gradients
HUVD	Honda/UCSD Video Database
ICF	Integral Channel Features
IP	Integer Programming
JPDAF	Joint Probabilistic Data Association Filter
KLT	Kanade-Lucas-Tomas
KNN	K-Nearest Neighbour
KRR	Kernel Ridge Regression
LBP	Local Binary Pattern
LQP	Local Quantized Patterns

LPH	Living Place Hamburg
LTP	Local Ternary Patterns
LUT	Lookup Table
MAP	Maximum a Posteriori
MBGS	Matched Background Similarity
MCMCDA	Markov Chain Monte Carlo Data Association
MHT	Multi Hypotheses Tracking
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MR	log-average Miss Rate
MWIS	Maximum-Weight Independent Set
NCC	Normalized Cross-Correlation
PCA	Principal Component Analysis
RLS	Regularized Least Squares
ROC	Receiver Operating Characteristic
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
SVR	Support Vector Regressor
TN	False Positives
TP	True Positives
VLBP	Volume Local Binary Patterns

Chapter 1

Introduction

In recent years, the demand for vision based surveillance systems has increased significantly. A special case is indoor surveillance in smart environments such as smart homes or smart rooms. While general surveillance systems attempt to control access to resources or to detect individuals on a watch list, the aim of surveillance in smart environments is to capture human behaviour details in order to provide automated services. For example, when a person is watching TV, the smart home could automatically provide a list of preferred channels depending on who is watching TV.

The concept of smart environments evolves from the definition of Ubiquitous Computing that, according to Mark Weiser, refers to the idea of:

> *"‘a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network"’* [Weiser et al., 1999].

Smart homes further develop this idea with the aim to create an environment which behaves intelligently, recognizes the user, learns or knows her/his preferences, and has the capability to exhibit empathy with the user's mood and current overall situation [Augusto, 2009]. In smart homes, learning means that the environments

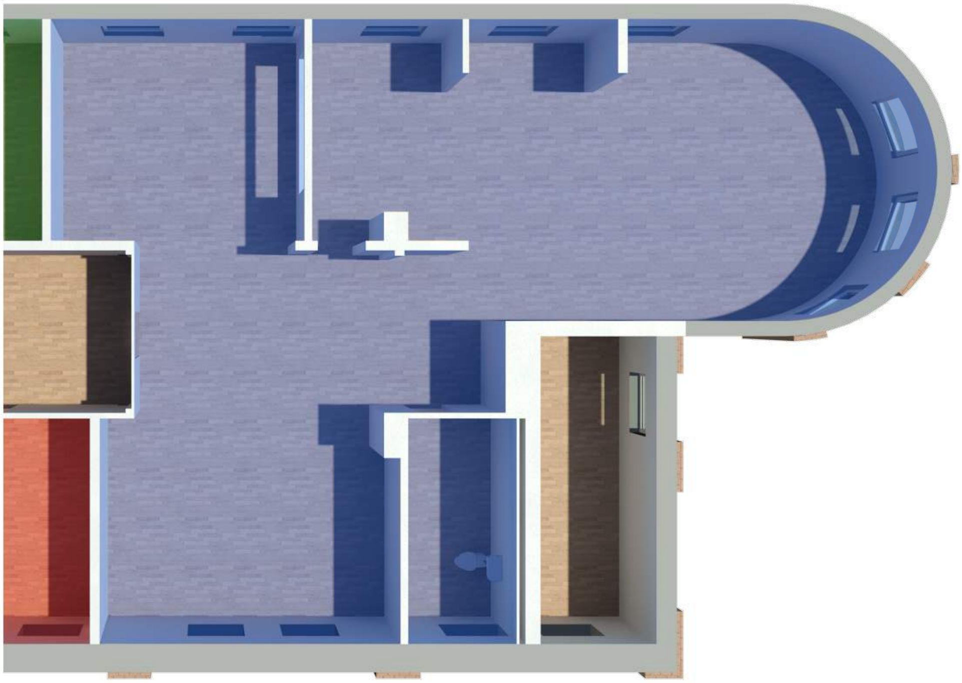


Figure 1.1: Living Place Hamburg top down view.

must gain knowledge about the preferences, needs, and habits of the user in order to be in better position to assist the user adequately [Juan Carlos Augusto, 2006, Aztiria et al., 2012]. Camera-based sensors in combination with computer vision technologies provide a setting that allows gaining this knowledge. It is physically intangible and depending on the number, the location and the type of cameras that are used. It is not necessary that the user is at a particular place in order to operate.

This thesis is part of the Living Place Hamburg (LPH) smart home project. It is mainly funded by the Hamburg Ministry of Commerce and the Ministry of Science and Research. The LPH covers different areas of IT-based urban living. In addition to typical questions from the smart home area, general questions of urban living are investigated. It is a $140m^2$ loft style apartment located at the campus of the University of Applied Sciences (HAW) in the centre of Hamburg (see Figure 1.1 and 1.2). The apartment consists of one large room with different sections for dining, living, cooking, sleeping and working as well as a separated bathroom. The LPH is a complete functional apartment and therefore suitable for making experiments under real-life conditions. Experimental living can range

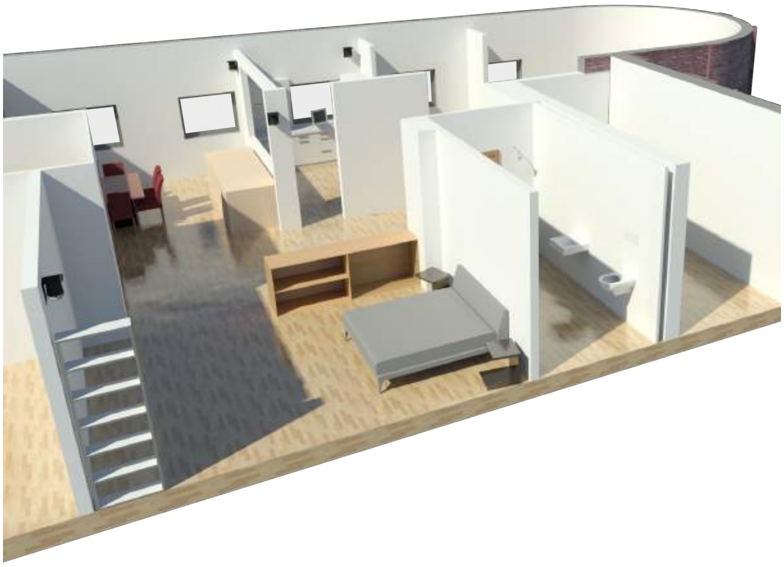


FIGURE 1.2: Living Place Hamburg side view.

from hours to several days. All these experiments can be controlled and supervised through a controller room [Place, 2012]. The camera installation consists of five pan-tilt-zoom cameras and three 360° cameras.

The research aim of the LPH is to analyse the relations between inhabitant and smart home and therefore to develop smart indoor environment that can identify and track their inhabitant as unobtrusively as possible and answer queries about their whereabouts and emotional state.

This thesis is part of multiple works that are dealing with these topics. The scope of this work is limited to developing and investigating tracking, detection and verification algorithms in order to identify a person and to estimate his/her position, based on the last known position and his/her speed. Results of this work are input to later work that focuses on facial expression recognition, emotion recognition and smart home interaction. Figure 1.3 shows an overview of the framework.

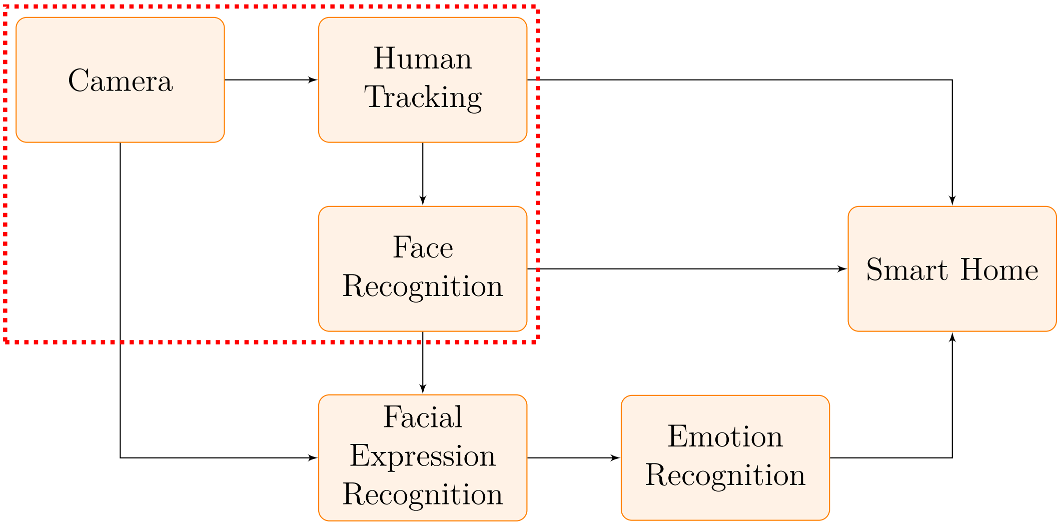


FIGURE 1.3: Overview of the LPH framework. The area marked with a red dotted line is the scope of this thesis.

1.1 Challenges

A significant volume of research has been done on human detection [Dollár et al., 2012], however most of this research focuses on full body detection of standing people. This is perhaps a reasonable assumption in an outdoor scenario, such as a shopping street, however, in an indoor scenario it is not. In indoor locations people are regularly sitting and they are often occluded by objects such as tables or chairs. In order to solve this problem new strategies are necessary that focus on regions of the body, that are visible if a person is sitting or occluded, such as upper body and head regions.

Furthermore, another challenge for detectors is the computational costs. Detectors recently evaluated on the Caltech Pedestrian dataset range in time from 1 to 30 seconds per frame on 640 × 480 resolution videos [Dollár et al., 2012]. In an application such as indoor surveillance, real-time is needed for High Definition (HD) resolution videos (1920 × 1080), and in order to achieve this, appropriate strategies are needed that minimise the computing cost. At least HD resolution is needed, because experiments have shown that videos with lower resolution are not sufficient to reliable detected upper body and head regions of humans that are further away from the camera.

However, human detection is only the first step for a reliable indoor tracking system. The next step is a robust tracking model which accurately represents the error characteristics of the detections. The main challenges for such a model are false detections, imperfect detection, low resolution, abrupt motion, and illumination and appearance changes. A good tracking model must be robust to such problems and simultaneously accurate enough to create robust location estimates.

Another difficulty of camera based human tracking systems, particularly in indoor environments like the LPH, is the limited field of view of a single camera. A solution for this problem is to use multiple cameras to monitor a wider field of view. However, using multiple cameras has its own challenges. For example, to compute the topology of camera network a precise calibration step is necessary or if a human is observed in different camera views an association step has to be applied [Wang, 2013]. Apart from these aspects, it is not always possible to install multiple cameras.

In addition to the indoor position system, an important challenge is the identification of a person through face recognition. Numerous approaches have been developed for face recognition, however the majority of these approaches focus on still image recognition [Zhang and Gao, 2009]. In this thesis the focus is on video-based face recognition. Compared to still image face recognition, video-based face recognition has great advantages because videos contain more abundant information than a single image. As a result, more accurate results can be achieved by fusing information of multiple frames. However, video-based face recognition also suffers from several problems such as low quality images, illumination changes, pose variations and occlusions. In order to perform robust video face recognition intelligent strategies are needed to utilise information from multiple frames while retaining robustness to pose, lighting conditions and other misleading cues.

1.2 Objectives of Research

The objectives of this research are derived from the shortcomings of current approaches and motivated by the challenges in the smart living environment. The overall objective is to build a camera based human localization and recognition system. In this thesis, the overall objective is decomposed into four specific objectives which are addressed separately:

1. To design new frameworks for more accurate identification of individuals in video face recognition.
2. To investigate more reliable detection of individual people from overhead surveillance cameras under different viewpoints.
3. To devise more robust real-time systems for tracking multiple individuals through camera views, where inter-person and object occlusion may be present.
4. To implement and empirically evaluate the performance of the proposed approaches using different datasets.

1.3 Limitations

Several problems such as illumination changes, pose variation, and occlusion are difficult to solve and not all of them can be solved in this work. That is why the scope of this thesis is reduced with the following limitations:

- Known area: The final experiments are conducted in the LPH.
- Good light conditions.
- Humans have to be in an upright sitting or standing position.
- Frontal face with good resolution for face recognition.
- Good training data for face recognition.

1.4 Contributions

The major contributions of this thesis can be summarised as follows:

- Two fully automatic frameworks for video face recognition are proposed which include face detection, face tracking, face alignment and video face recognition. For the first framework a novel algorithm for building face tracks in real-life scenarios is proposed, that combines face detection and optical flow tracking to a face tracking algorithm. The temporal information which is provided by the face tracks is used to significantly improve the recognition rate of Local Binary Pattern (LBP) face recognition in video scenes. The second framework is based on multi-target tracking, where a novel face detector is combined with a head detector in order to create detections that are then used to build tracks. Two novel set-to-set similarity measuring schemes are proposed that determine whether faces appearing in the two tracks are the same subject.

- A combined detector is proposed to solve the problem of robust head detection. The idea is not to rely on a single detector. Instead, a head, an upper-body and a body detector are used for decision making by combining their individual opinions to derive a consensus decision.

- A real-time multi-target tracking system that effectively deals with false positive detections is proposed. In order to achieve this, a novel motion model is proposed that treats false positives on background objects and false positives on foreground objects such as shoulders or bags separately. In addition, a schema is proposed that includes the identification of true positives with the data association instead of using the internal decision-making process of the detector. Experiment results show that the system is superior to previous works.

- Finally, a second approach is proposed for real-time multi-target tracking, which extends the first approach for fisheye cameras. A camera model is described that allows fast projection between a fisheye image and a corresponding set of perspective images. These perspective images allow the application of standard detection algorithms. Then an algorithm is proposed that automatically generates, from annotated heads in fisheye images, sets of aligned heads in perspective images that are then used to train a combined multi-view detector. Based on this head detector, the first real-time multi-target tracking algorithm is extended for tracking humans on calibrated fisheye cameras. In addition, a new dataset for tracking humans in fisheye videos is created on which evaluation is performed.

1.5 Thesis Organisation

The rest of this thesis is organized as follows: Chapter 2 discusses related work on tracking and face recognition. Chapter 3 describes an entire framework for video face recognition from tracking to recognition. Next, in Chapter 4, a detector is proposed for robust head detection. Chapter 5 describes a real-time multi-target tracking system. In Chapter 6 a face detector, a novel set-to-set similarity measure and a fully automatic framework for video face recognition are presented. Chapter 7 describes an extended framework for multi-target tracking in videos captured with fisheye cameras. Finally, this thesis ends with a summary of conclusions and future work in Chapter 8. The relation between the different chapters is shown in Figure 1.4.

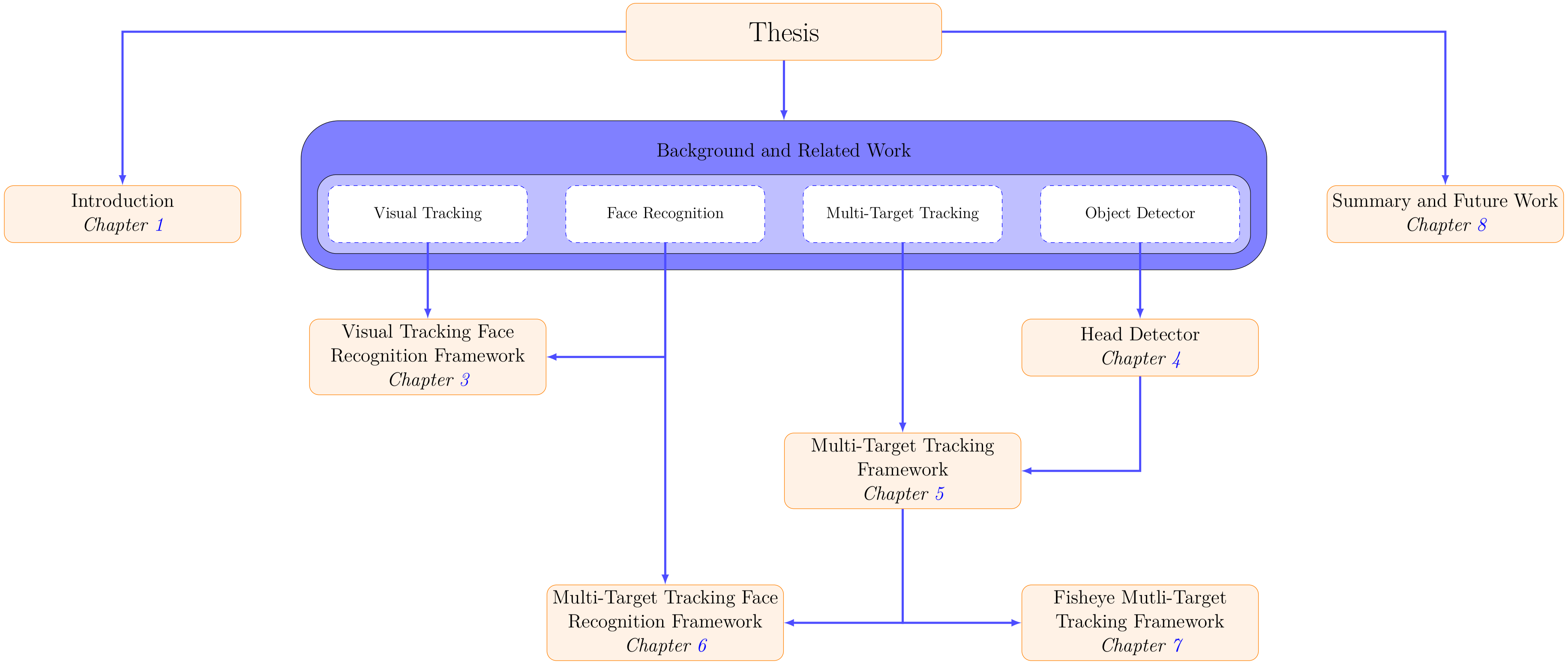


Figure 1.4: Overview of the thesis and the relation between different chapters.

Chapter 2

Review of Related Literature

The aim of this chapter is to provide a summary of recent methodologies employed for tracking and face recognition. In section 2.1, tracking approaches are described. These approaches have been divided into visual tracking and multi target tracking techniques. Section 2.2 provides information on face recognition. The focus of this review is on papers directly related to the methods proposed in this thesis, hence papers that use the same base algorithm. Furthermore, papers are reviewed that were evaluated on the same datasets that were used in this thesis for evaluation. Finally, pioneer work was include that was consistently mentioned in the related papers.

2.1 Tracking

Camera based tracking can be divided into two subcategories: visual tracking and multi-target tracking. Visual tracking focuses on tracking a single target or multiple targets separately whereas multi-target tracking focuses on tracking multiple targets jointly. In the following both approaches will be reviewed.

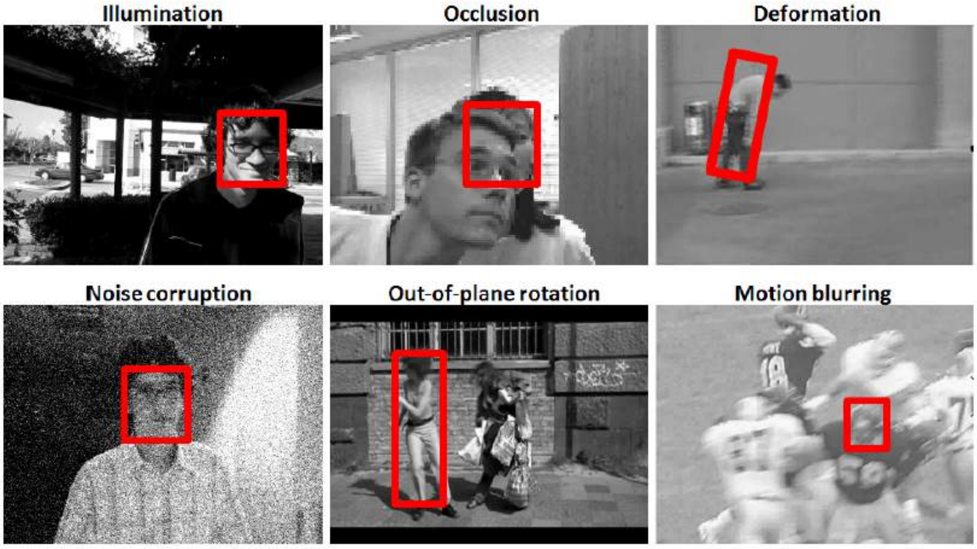


Figure 2.1: Illustration of complicated appearance changes in visual object tracking (taken from: [Li et al., 2013b]).

2.1.1 Visual Tracking

Visual tracking approaches focus on tracking a single target or multiple targets separately. These methods are able to track targets without regard to their category, from a manual initialization; no offline trained detector is needed, and tracking is usually based on appearance only to deal with abrupt motions. From the given label in the first frame, an appearance model is learned online to discriminate the target from all other regions; the appearance model is usually online updated to deal with illumination and view angle changes [Yang and Nevatia, 2014] (see Figure 2.1). In this section recent algorithms are reviewed in terms of the main modules that were defined in [Wu et al., 2013]: representation schemes, search mechanisms, and model update, context and fusion of trackers.

2.1.1.1 Representation Schemes

Object representation reflects the statistical characteristics of object appearance and is one of the main parts in every visual tracker. A wide range of schemes have been proposed [Li et al., 2013a, Wu et al., 2013]. One of the simplest and widely used representation for the object region are raw intensity or colour values. Such representations are usually constructed as either vector-based [Silveira and Malis, 2007, Ross et al., 2008] or matrix-based [Bolme et al., 2010, Henriques et al., 2012, Ben-Ari and Ben-Shahar, 2013]. In addition to raw pixel values, many other visual features have been used for visual tracking, such as colour histograms [Bradski, 1998, Comaniciu et al., 2003], Histograms of Oriented Gradients (HOG) [Tang et al., 2007], covariance region descriptor [Tuzel et al., 2006, Wu et al., 2012a], Haar-like features [Grabner et al., 2006], binary pattern [Kalal et al., 2010, Dinh et al., 2011, Kalal et al., 2011] and keypoint descriptors such as SIFT [Zhou et al., 2009] or SURF [He et al., 2009].

Since the work of Lucas and Kanade [Lucas and Kanade, 1981], optical flow has been consistently used to capture the spatio-temporal motion information of a target [Avidan, 2004, Kalal et al., 2010, Kalal et al., 2011]. Optical flow represents a dense field of displacement vectors of each pixel inside an image region.

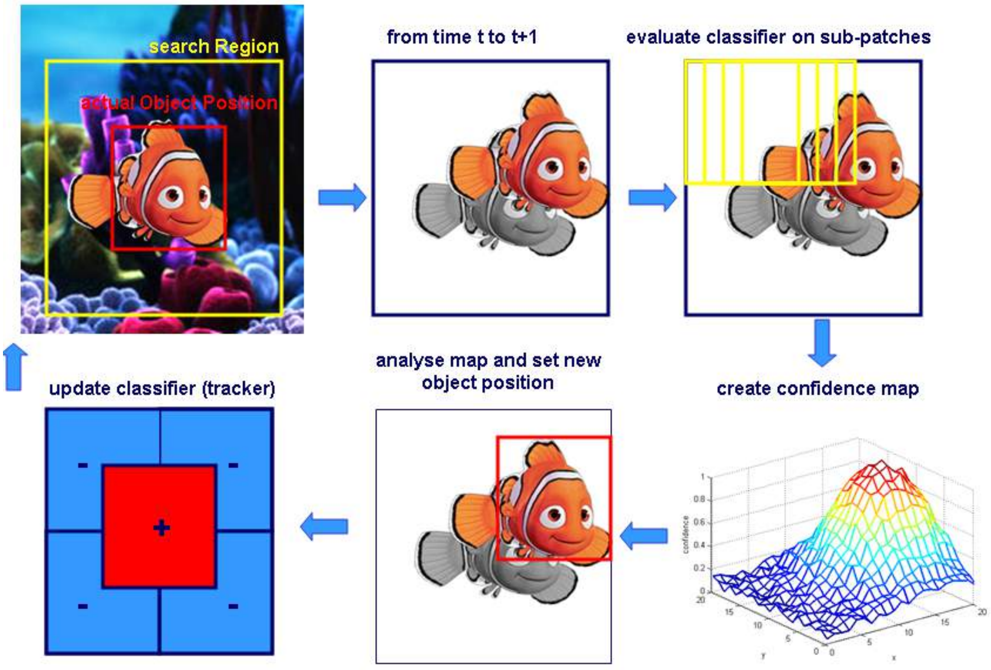

search Region
actual Object Position
from time t to t+1
evaluate classifier on sub-patches
create confidence map
analyse map and set new object position
update classifier (tracker)
-
-
+
-
-
confidence

Figure 2.2: Illustration of tracking-by-detection based on online boosting (taken from [Grabner et al., 2006]).

Recently, tracking-by-detection for visual tracking has become popular. In such methods tracking is viewed as a binary classification issue where an online learned classifier is used to discriminate the target from the background. Several learning algorithms have been proposed, such as SVM [Avidan, 2004, Hare et al., 2011, Bai and Tang, 2012], random forests [Dinh et al., 2011, Kalal et al., 2011] and boosting [Grabner et al., 2006, Avidan, 2007]. For example Figure 2.2 illustrates a tracking-by-detection based on online boosting which was proposed by [Grabner et al., 2006]. Given an initial position of the target in time t, their algorithm first evaluates a classifier at all possible positions in a surrounding search region in frame $t+1$. The achieved confidence map is analysed in order to estimate the most probable position and finally classifier is updated.

2.1.1.2 Search Mechanism

In order to search the target location, several methods have been proposed. If the objective function is differentiable with respect to the motion parameters the estimation process can be in the form of gradient ascent (descent)-based maximization (minimization) [Lucas and Kanade, 1981, Comaniciu et al., 2003]. However, these objective functions are usually non linear and contain many local minima [Wu et al., 2013]. Other methods adopt sliding window to avoid this problem [Avidan, 2004, Grabner et al., 2006, Avidan, 2007, Dinh et al., 2011, Hare et al., 2011, Kalal et al., 2011, Bai and Tang, 2012]. In order to reduce the computational load most algorithms are only applied in a small search window around the estimated target location (Figure 2.2 shows a sample). In [Henriques et al., 2012] a Fourier analysis based approach is proposed that allows the use of the Fast Fourier Transform (FFT) to quickly learn and detect from all sub-windows, without iterating over them. Other widely used algorithms are stochastic search algorithms such as particle filters [Perez et al., 2002, Ross et al., 2008, Zhong et al., 2012] which are a set of online posterior density estimation algorithms that estimate the posterior density of the state-space by directly implementing the Bayesian recursion equations.

2.1.1.3 Model Update

While early visual tracking approaches used models that do not change [Bradski, 1998, Perez et al., 2002, Collins, 2003, Comaniciu et al., 2003, Adam et al., 2006, Ben-Ari and Ben-Shahar, 2013], most of the recent approaches use adaptive models to account for appearance variations during tracking. For templates that problem has been addressed by a combining static and adaptive template updating [Matthews et al., 2004, Dowson and Bowden, 2005, Rahimi et al., 2008], as well as by updating reliable parts of the template [Jepson et al., 2003, Adam et al., 2006]. Templates have limited modelling capabilities, since they learn visual representations for the foreground object region information while ignoring the influence of the background. As a result, they often suffer from inaccuracies caused by the background regions with similar appearance to the object class [Li et al., 2013a]. In comparison, discriminative approaches update models with positive and negative (background) samples, with the aim to maximize the separability between the target and non-target regions. Several update algorithms have been proposed via online mixture models [Jepson et al., 2003], online boosting [Grabner et al., 2006], incremental subspace update [Ross et al., 2008], P-N learning [Kalal et al., 2011] and Regularized Least Squares (RLS) [Henriques et al., 2012].

2.1.1.4 Context and Fusion of Trackers

In order to improve tracking, recently some authors proposed approaches that exploit context information by mining auxiliary objects or local visual information surrounding the target to assist tracking [Yang et al., 2009, Grabner et al., 2010, Dinh et al., 2011]. Therefore, the region around the target is searched for supporting objects that have a similar motion model. These supporting objects are especially helpful to determining the position of the target when it disappears from the camera view or undergoes a difficult transformation.

Besides context information, authors proposed fusion methods to improve tracking. Several authors combines static and adaptive models in order to reduce the drifting problem that appears by self updates of online learning methods [Santner et al., 2010, Kalal et al., 2011, Dinh et al., 2011]. Multiple parallel running trackers [Kwon and Lee, 2011] and multiple feature sets [Yoon et al., 2012] are combined in a Bayesian framework to better account for appearance changes. Chan et al.

[Chan et al., 2013] proposed a collaborative tracking algorithm that uses a Bayesian framework to combine self-information and information from other objects based on a motion similarity measure.

More detailed reviews can be found in [Yilmaz et al., 2006, Yang et al., 2011, Li et al., 2013b] and a large benchmark of visual tracking algorithms can be found in [Wu et al., 2013].

2.1.2 Multi-Target Tracking

Multi-target tracking methods focus on tracking multiple targets of a pre-known class simultaneously. They usually first apply a pre-trained detector and then try to find a global optimal solution for all targets that associates all detections into tracks (data association).

Multi-target video trackers can be divided into causal and non-causal methods. Causal methods use only current and past observations to estimate the current state. Non-causal methods use also future information to estimate the current state. Although non-causal approaches are not suitable for time-critical applications, they can resolve ambiguities more easily. Since this work focuses on real-time tracking, the literature review focuses on causal methods.

Multi-target tracking can be divided in the detection and the data association steps. In the detection step a detector that was trained for a pre-known class is applied to detected all objects of this class. Data association describes the process of associating detections over multiple frames into tracks in such a way that detections of the same person belong to one track. Figures 2.3 illustrates this, the detections are represented by rectangles which were associated into tracks. Detections of the same track have the same colour. In the following both steps are reviewed separately.

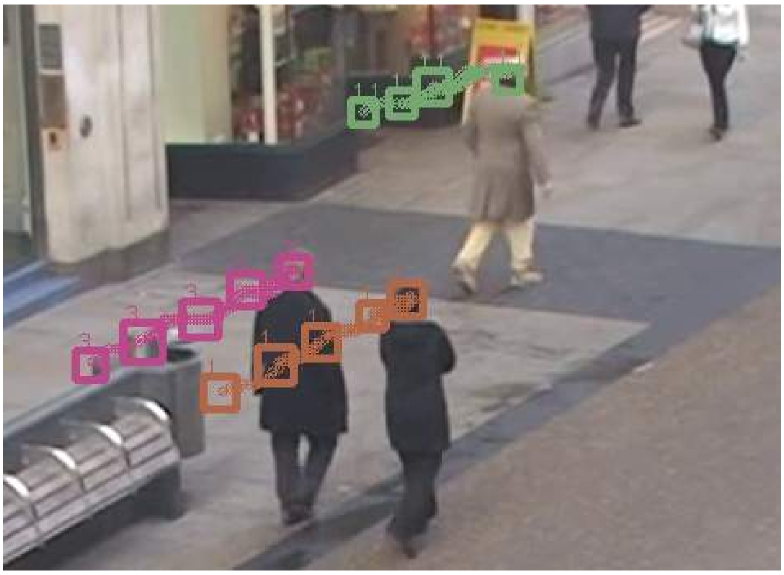


Figure 2.3: Detections that were associated into tracks.

2.1.2.1 Human Detection

One of the first sliding window detectors was proposed by Papageorgiou et al. [Papageorgiou and Poggio, 2000] which used Haar wavelet features in combination with a Support Vector Machine (SVM). Viola and Jones [Viola and Jones, 2002] built upon these ideas, by using AdaBoost to train a cascade structure classifier for efficient detection, and introducing integral images for fast Haar-like feature computation. Inspired by SIFT [Lowe, 2003], Dalal and Triggs [Dalal and Triggs, 2005] popularized HOG features for detection. The essential thought is that appearance and shape of an object can be described by the distribution of local intensity gradients or edge directions. In practice this is implemented by dividing the image window into small spatial regions (cells) and then accumulate for each region a local 2D histogram of gradient direction or edge orientations (see Figure 2.4). The combined histogram entries form the representation [Dalal and Triggs, 2005]. Since their introduction, several variants of HOG features have been proposed and nearly all modern detectors utilise them in some form [Dollár et al., 2012].

Several other features have been used for detection, such as shapelets [Sabzmeydani and Mori, 2007] which are a set of automatically learned gradient-based

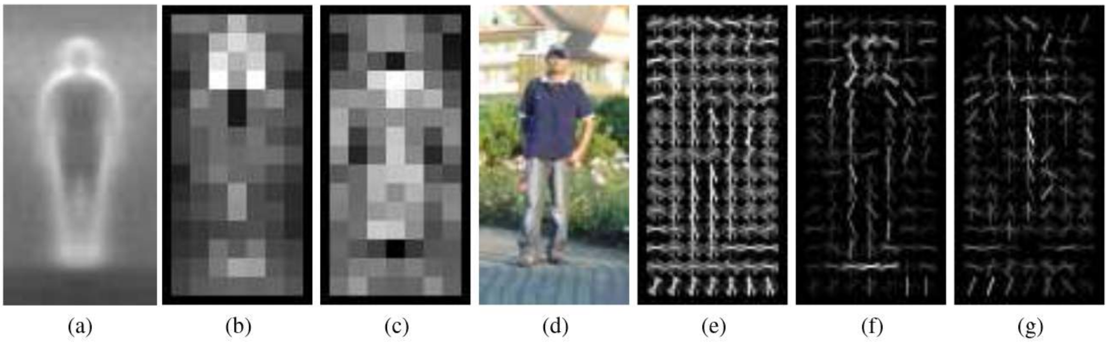


Figure 2.4: Illustration of the HOG feature (taken from: [Dalal and Triggs, 2005]) (a) The average gradient image over the training examples. (b) Each pixel shows the maximum positive SVM weight in the block centred on this pixel. (c) Likewise for negative SVM weights. (d) A test image. (e) Its computed HOG descriptor. (f) The HOG descriptor weighted by the positive SVM weights. (g) Likewise for negative SVM weights.

features. Boosting was used in order to combine multiple shapelets into an overall detector. Shape Context [Belongie et al., 2002] has originally been proposed as a feature point descriptor and was later exploited for people detection [Leibe et al., 2005]. The descriptor is based on edges which are then stored in a log-polar histogram. Granularity-tunable features were proposed in [Liu et al., 2009] that use granularity to define the spatial and angular uncertainty of line segments in Hough space. An extension to the spatial-temporal domain was proposed in [Liu et al., 2010]. In films and videos motion features can be used to improve detecting. Motion features were successfully incorporated into detectors in [Viola et al., 2005] by computing Haar-like features on difference images. Dalal et al. [Dalal et al., 2006] used motion descriptors based on optical flow.

While no single feature has been shown to outperform HOG, additional features can provide complementary information [Dollár et al., 2012]. Wojek and Schiele [Wojek and Schiele, 2008] combined several features such as Haar-like features, shapelets, shape context, and HOG features and showed that the combined feature outperforms any individual feature. This approach was extended by Walk et al. [Walk et al., 2010] by introducing self-similarity on colour channels and motion features. Dollar et al. [Dollár et al., 2009] proposed a framework for integrating LUV colour channels, grayscale, and gradient magnitude quantized by orientation features.

Several method based on part-based models have been proposed, Mikolajczyk et al. [Mikolajczyk et al., 2004] modelled humans as flexible assemblies of parts, which are represented by SIFT-like local features which captures the spatial layout of the part appearance. In [Felzenszwalb and Huttenlocher, 2005] pictorial structures for object recognition were proposed that represent objects by a collection of parts arranged in a deformable configuration. Each part captures local appearance properties of an object while the deformable configuration is characterized by spring-like connections between certain pairs of parts. Felzenszwalb et al. [Felzenszwalb et al., 2010b] described a latent SVM formulation that represents highly variable objects using mixtures of multi scale deformable part models. These part-based models are trained from overall bounding boxes without part location labels. Shu et al. [Shu et al., 2012] extended the approach with partial occlusion handling for multiple person tracking by examining the contribution of each individual part through a linear SVM.

Considerable effort has also been devoted to achieve real-time performance. Zhu et al. [Zhu et al., 2006], proposed fast computation of gradient histograms using integral histograms. However, the proposed system was real time for single scale detection only. Zhang et al. [Zhang et al., 2007] proposed a coarse-to-fine detection scheme that rejects the majority of negative windows at lower resolution, leaving a relatively small number of windows to be processed in higher resolutions. Prior knowledge is often used [Sudowe and Leibe, 2011, Benenson et al., 2011, Benenson et al., 2012] to reduce the search space thereby improving both speed and quality. Several authors ported their algorithm to GPUs [Wojek et al., 2008, Prisacariu and Reid, 2009, Benenson et al., 2012] or used parallel implementation on multiple CPUs [Benenson et al., 2012].

A frequently used method for speeding up classifiers is to split them up into a sequence of simpler classifiers [Viola and Jones, 2002, Felzenszwalb et al., 2010a, Zhu et al., 2006]. By having the first stages prune most of the false positives, the average computation time is significantly reduced. In order to do fast multi-scale detection, Dollár et al. [Dollár et al., 2010] demonstrated how features computed at a single scale can be used to approximate features at nearby scales. Figure 2.5

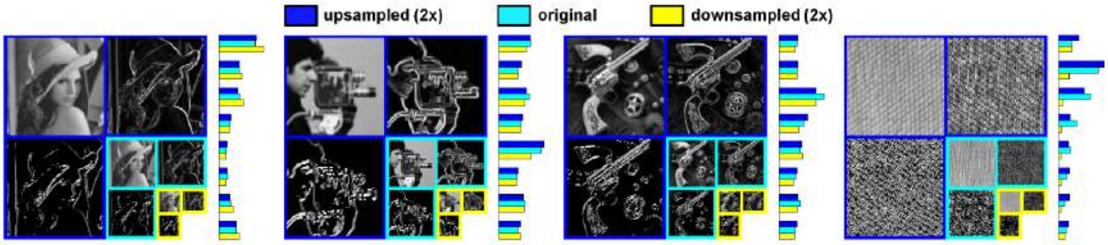


Figure 2.5: Example of approximated features (taken form [Dollár et al., 2010]).

shows some sample. For each image set, the original image (cyan border) is taken and an upsampled (blue) and downsampled (yellow) version are generated. Shown at each scale are the image (centre), gradient magnitude (right) and gradient orientation (bottom). At each scale a gradient histogram with 8 bins is computed and each bin is normalized by .5 and $.32^{-1}$ in the upsampled and downsampled histograms respectively [Dollár et al., 2010]). This approach was further improved in [Dollár et al., 2014].

2.1.2.2 Head Detection

Person detection is a well-studied problem in computer vision with many methods and evaluation benchmarks available. However, most of the methods consider full-body (pedestrians) or upper-body detection. In theory, the same algorithms can be used for head detection, but in practice, these algorithms do not achieve a satisfactory result. In order to overcome this problem, several authors have proposed strategies to exploit additional features.

Zhang and Gao [Zhang and Gao, 2009] constructed a categorical model for hair and skin, and trained the model in four categories of skin representing the different illumination conditions (bright, standard and dark) to increase pedestrian detection rates during an occlusion event. Head detection using a skeleton graph is proposed in [Merad et al., 2010]. The skeleton graph is extracted from the foreground mask which is obtained with background subtraction.

In [Venkatesh et al., 2012], interest points are detected using gradient information in order to approximately locate top of the head regions to reduce the search space. The interest points are then masked using a foreground region obtained using background subtraction techniques. A sub-window is then placed around the interest points, and it is classified as a head or non-head region using an AdaBoost classifier. Xie et al. [Xie et al., 2012] detected heads using the HOG feature. To improve the detection result, motion and appearance features are extracted and then the Bayesian posterior is used to represent the probability of the detected region belonging to actual human head region.

Marin-Jimenez et al. [Marin-Jimenez et al., 2014] proposed a two-level pipeline in what an upper-body detector is applied and then heads are detected within the upper-body detection areas. For both the upper-body and the head detector, they trained a part-based model.

2.1.2.3 Data Association

Early work mostly focused on recursive methods, where the current state is estimated only using information from previous frames. The Kalman filter approaches [Black et al., 2002] are a prominent example. The Kalman filter consists of a prediction and an update step. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. In the update step, the estimate is updated using a state transition model and measurements. Another popular approach is particle filtering [Isard and Blake, 1998] (also known as Sequential Monte Carlo), where a set of weighted particles-sampled from a proposal distribution is maintained to represent the current, hidden state [Giebel et al., 2004, Okuma et al., 2004, Breitenstein et al., 2009]. This allows handling non-linear models and multi-modal posterior distributions.

More recently, non-recursive tracking methods have grown more and more popular. The commonality of these methods is that all trajectories are estimated jointly within a given time window. A number of methods have recently been proposed. Classical approaches are Joint Probabilistic Data Association Filter (JPDAF) [Fortmann et al., 1983] and Multi Hypotheses Tracking (MHT) [Reid, 1979]. MHT considers multiple possible associations over several time steps. JPDAFs instead tries to make the best possible assignment in each time step by jointly considering all possible associations between targets and detections [Breitenstein et al., 2011a]. As the number of observation grows, the complexity of both methods becomes unmanageable in practice [Ge and Collins, 2008].

The Hungarian algorithm [Kuhn, 1955] is another algorithm that can be used to find the best assignment of possible detection-track pairs in a runtime that is cubic in the number of targets. Stauffer [Stauffer, 2003] first obtained tracklets by performing a conservative frame-to-frame correspondence, and then associated these tracklets by the Hungarian algorithm. This approach was further extended [Kaucic et al., 2005] by introducing a segmentation based scene understanding module to estimate the locations of scene occlusions. Singh et al. [Singh et al., 2008] used a Multiple Hypothesis Tracker to grow tracklets before associating them by the Hungarian algorithm. Huang et al. [Huang et al., 2008] linked detections based on conservative affinity constraints to tracks. In a second level these tracks are then further associated based on more complex affinity measures, with the Hungarian algorithm, to form longer tracks. Some authors [Wu and Nevatia, 2007, Breitenstein et al., 2011b] used a greedy matching algorithms that achieves equivalent results to the Hungarian Algorithm but at lower computational complexity.

Another popular method is Markov Chain Monte Carlo Data Association (MCMCDA), which was first introduced for tracking a single or fixed number of targets [Pasula et al., 1999]. Oh et al. [Oh et al., 2004] extended the approach for general multiple-target tracking problems, in which unknown numbers of targets appear and disappear at random times. They presented a multi-scan MCMCDA algorithm that approximates the optimal Bayesian filter. Later, MCMCDA was adapted specifically for visual tracking by associating object detections resulting from background subtraction [Yu et al., 2007] and a boosted Haar classifier cascade [Liu et al., 2007b]. Ge and Collins [Ge and Collins, 2008] further developed this approach by using not only object detections but also tracklets, which were created by using a standard tracking algorithm for a short period after each detection.

Automatic parameter estimation was proposed in [Liu et al., 2007a] as a linear programming problem, but labelled sequences are required. Automatic parameter estimation from unlabelled data was proposed by Ge and Collins [Ge and Collins, 2008], their algorithm learns the model parameters by interleaving the MCMCDA sampling with a Gibbs sampler that updates the model parameters. Benfold and Reid [Benfold and Reid, 2011] combined asynchronous HOG detections with simultaneous Kanade-Lucas-Tomas (KLT) tracking in an accurate real-time algorithm. In addition, they presented a novel approach for false positive detection by creating a separate model for false positives and combining the identification of false positives with data association.

Several other methods have been proposed to find global optimum. Berclaz et al. [Berclaz et al., 2011] formulate multi-target tracking as an Integer Programming (IP) problem and demonstrated how the k-shortest paths algorithm can be used to solve this problem. Milan et al. [Milan et al., 2013] proposed a mixed discrete-continuous conditional random field that handles inter-object exclusions at two levels: at the data association level based on non-submodular constraints and at the track level. Brendel et al. [Brendel et al., 2011] divided the data association problem into disjoint subgraphs and then formulate associating object detections with tracks as finding the Maximum-Weight Independent Set (MWIS) of these graphs.

2.1.3 Fisheye Camera Multi-Target Tracking

One method proposed in this thesis involves tracking on cameras with fisheye lens, that is why this section gives a review about previous tracking methods on cameras with fisheye lens. A fisheye lens is an ultra wide-angle lens that produces strong visual distortion intended to create a wide hemispherical image.

One of the early works in human tracking in fisheye videos is done by Wang [Wang, 2006] who used motion history images to find a target and then tracked the target by using CamShift. Kubo et al. [Kubo et al., 2007] applied first background subtraction and then grouped the foreground pixels by clustering. Subsequently an ellipse as geometrical model was fitted to the groups. Saito et al.[Saito et al., 2010] introduced a probabilistic model to describe the wide variation of human appearance in hemispherical image. They model a human as variable shape features of body silhouettes and head and shoulder contours. Non linear template models are built by the combination of Principal Component Analysis (PCA) and Kernel Ridge Regression (KRR). Finally, the problem of human detection was formulated as a Maximum a Posteriori (MAP) estimation problem using the above model.

More recently, a two step system was proposed by Yuan et al. [Yuan et al., 2011]. In the first step they tried to detect possible head candidates, therefore they extracted edges of the input image using the Sobel operator then they computed the subtraction between the input edge image and a background edge image. Afterwards, they used an ellipse detector to find possible head candidates. In the second step, a bounding box was constructed depending on the position of the head candidates and then a classifier was applied to categorise the bounding boxes into humans and non humans.

Vandewiele et al. [Vandewiele et al., 2012] presented a system based on a network of fisheye cameras designed to track customers in a retail store. They detected moving objects with background subtraction and in order to detect people they fit an ellipse around each connected component of foreground pixels. The detections were then associated to a track using the position on the ground plane and an appearance model.

2.2 Face Recognition

The process of face recognition can be roughly divided into four parts: face detection, face alignment, feature extraction and recognition (see Figure 2.6). In the following, each of these topics are reviewed and in addition recent video based face recognition methods are reviewed explicitly. For recognition the review focuses on

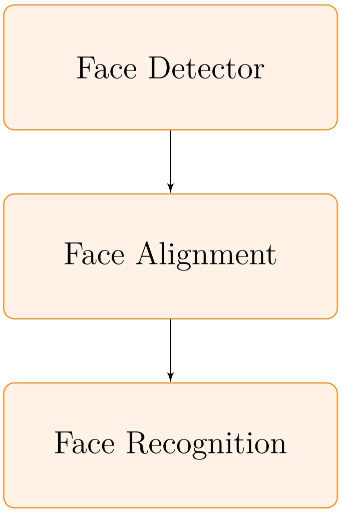


Figure 2.6: Face recognition framework overview

metric learning methods since that are the most related methods to the methods proposed in this thesis.

2.2.1 Face Detection

In section 2.1.2.1 methods for human detection were reviewed and the same methods are often used for face detection. That is why in this section only a short review is done on methods that were specially designed for face detection, a more detailed review can be found in [Zhang and Zhang, 2010].

Viola and Jones [Viola and Jones, 2002] proposed a method to combine integral image based Haar-like features, the Adaboost based classifier and cascade based fast inferences. The authors in [Lienhart and Maydt, 2002] proposed an extended set of Haar features for different views of faces. Zhang et al. [Zhang et al., 2004] used AdaBoost learning to select a set of local regions and their weights with respect to Local Binary Pattern (LBP) features for face detection. Furthermore, skin colour modelling has been proposed for face detection. Greenspan et al. [Greenspan et al., 2001] used Gaussian Mixture Models for modelling the skin colour distribution. Yan-Wen Wu et al. [Wu and Ai, 2008] used AdaBoost algorithm combined with skin colour segmentation. The segmentation is obtained by single Gaussian model fitting and morphological operations on binary images. Kai-Biao Ge et al.

[Ge et al., 2011] suggested an AdaBoost algorithm combined with skin segmentation and LBP based face description. Ban et al. [Ban et al., 2014] proposed a boosting-based face detection method using skin colour information without any parametric fitting or morphological operations.

Part based models have also been exploited for face detection. Early work has been proposed by Heisele et al. [Heisele et al., 2001] whose system consists of a two-level hierarchy of SVMs. On the first level component classifiers independently detect components of a face. On the second level the authors checked the geometrical configuration of the detected components. Cevikalp et al. [Cevikalp et al., 2013] combined a cascade of binary and one-class type classifiers for root detection and SVM like learning algorithm for part detection. Yan et al. [Yan et al., 2013] proposed a hierarchical part based structural model to explicitly capture large appearance variations such as pose and expression.

2.2.2 Face Alignment

A big challenge of unconstrained face recognition is the large amount of intra-class variability, due to factors such as illumination changes, pose variation, and perspective transformation. This intra-class variability can often be much larger than inter-class differences.

Removing undesired intra-class variability by aligning the faces to some canonical pose or configuration can lead to significant gains in recognition accuracy on unconstrained face recognition. This is true even for algorithms that were explicitly designed to be robust to some misalignments [Wolf et al., 2010].

Face alignment can be divided into two categories: fiducial point (or landmark-based) alignment and unsupervised alignment. Fiducial point alignment localizes facial feature points, such as corners of the eyes, mouth and the tip of the nose and then computes a similarity transformation which attempts to bring these points to a fixed location. Whereas in the unsupervised alignment case, a set of poorly aligned images (e.g. images from a detector) is taken and it is attempted to make

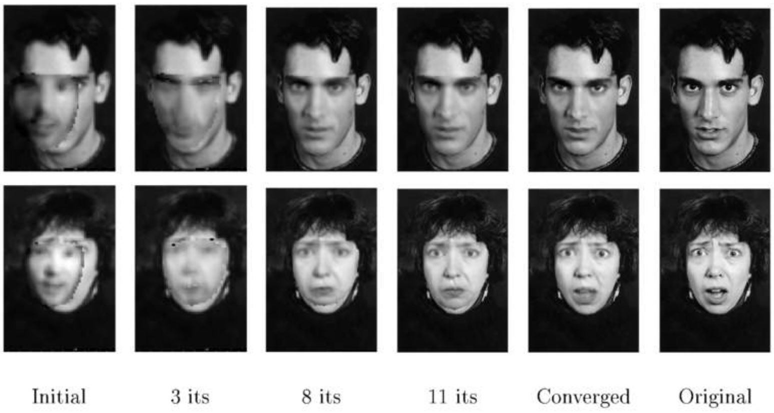


FIGURE 2.7: Multi resolution Active Appearance Model search for two faces, each starting with the mean model displaced from the true face centre (taken from [Cootes et al., 2001]).

the images more similar to each other by using a measure of joint similarity such as entropy [Huang et al., 2007, Huang et al., 2012b].

2.2.2.1 Deformable Model Fitting

One of the first most popular methods for fiducial point alignment is deformable model fitting. Deformable model fitting is the problem of registering a parametrized shape model to an image such that its landmarks correspond to consistent locations on the object of interest [Saragih et al., 2009]. The deformable model is first constructed from a set of training samples [Cootes et al., 2001] and then fits to the input image. Perhaps the most popular deformable models are the Active Appearance Models (AAM) [Cootes et al., 2001]. AAM combine a model of shape variation with a model of texture variation and then search a face by minimising the error between an input image and the model (Figure 2.7 shows an example of such an optimization). A face model that is similar in many ways to AMMs is the 3D Morphable Model (3DMM) [Blanz and Vetter, 1999]. The main difference between them is that the shape model of an AAM is 2D, whereas the shape model of a 3DMM is 3D.

Another successful deformable fitting model is the Constrained Local Model (CLM) [Cristinacce and Cootes, 2008, Saragih et al., 2009]. The CLM approach learns the variation in appearance of a set of template regions. The template regions

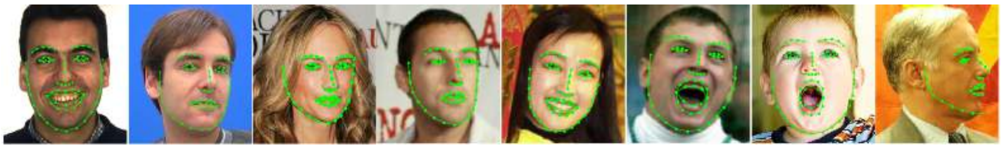


Figure 2.8: Selected landmark localization results from the work of Yu et al. (taken from [Yu et al., 2013]).

are then used as feature detectors in a local search, constrained by the full shape model.

Recently, regression-based approaches are getting more and more popular. This approaches learn a regression function that directly maps image appearance to the target output. Cao et al. [Cao et al., 2012] proposed a shape regression method that automatically encodes the shape constraint by jointly regressing the entire shape and minimizing the alignment error. Zhu and Ramanan [Zhu and Ramanan, 2012] proposed a tree structured part model of the face, which both detects faces and locates facial landmarks. Yu et al. [Yu et al., 2013] did speed up this approach by simplifying the mixture of parts for face detection and initial landmark localization (Figure 2.8 shows some results). Smith [Smith et al., 2014] further improved this approach by modelling the full interactions between each landmark and its surrounding local features.

Fiducial point detectors are another popular approach. The general idea is to train a classifier to respond to a specific fiducial point. Everingham et al. [Everingham et al., 2006] proposed a generative model of the fiducial point positions which is combined with a discriminative model of the fiducial point appearance. The probability distribution over the joint position of the fiducial point is modelled using a mixture of Gaussian trees. The appearance of each fiducial point is assumed independent of the other fiducial points and is modelled by a fiducial point/non-fiducial point AdaBoost classifier. In [Zhan et al., 2007] a Viola-Jones [Viola and Jones, 2002] detector is applied to detected fiducial point. Belhumeur et al. [Belhumeur et al., 2011] proposed an approach which combines the output of Support Vector Regressor (SVR) with a non-parametric set of global models for the part relative positions. Taigman et al. [Taigman et al., 2014] detected fiducial points with a SVR trained to predict point configurations from an image descriptor and then combined the output with a 3D shape model. The 3D shape model is then used to generate a 3D-aligned version of the face.

After finding the fiducial points, the faces can be warped to a conman face model using a similarity transformation. The similarity transformation defines the mapping between a fiducial point (u, v) and a model point (x, y) in terms of scaling s, rotation θ and translation t_x, t_y :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & -m_2 \\ m_2 & m_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2.1}$$

where $m_1 = s \cos\theta$ and $m_2 = s \sin\theta$. In order to find m_1, m_2, t_x and t_y a linear system can be defined:

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u_1 \\ u_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{bmatrix} \tag{2.2}$$

This linear system can be written as:

$$Ax = b \tag{2.3}$$

The least-squares solution for the parameters x can be determined by solving the corresponding normal equations:

$$x = (A^T A^{-1}) A^T b \tag{2.4}$$

which minimizes the sum of the squares of the distances from the projected model locations to the corresponding image locations. Once the transformation parameters m_1, m_2, t_x, t_y are obtained, the probe face can be transformed according to the 2D spatial similarity transformation.

2.2.2.2 Unsupervised Alignment

One of the early works of fully unsupervised face alignment from exemplars has been proposed by Frey and Jojic [Frey and Jojic, 1999]. They developed a method based on an Expectation-Maximization (EM) algorithm. The approach employed discrete hidden variables to model unwanted spatial variation. In [la Torre and Black, 2003] an extension of this approach was proposed, that learns a subspace, which is invariant to affine or higher order geometric transformations.

Learned-Miller [Learned-Miller, 2006] proposed a framework known as congealing that iteratively aligns an image set by minimizing the entropy function of the set. It was originally applied to joint alignment of binary images but was later applied to more complex object classes such as faces and cars [Huang et al., 2007]. Huang et al. [Huang et al., 2012b] incorporated deep learning into the congealing alignment framework in order to obtain features that can represent the image at different resolutions based on network depth and are tuned to the statistics of the specific data being aligned. Cox et al. [Cox et al., 2008] extended the congealing framework by introducing a sum of square differences cost function. The alignment was formulated under the Lucas-Kanade framework [Lucas and Kanade, 1981], and the optimization was iteratively solved by a Gauss-Newton gradient descent approach.

Zhu et al. [Zhu et al., 2009] extended this approach with a template based optimization scheme. They first fitted a set of template images to the input image by a deformable Lucas-Kanade fitting scheme and then the input face image was rectified into the canonical frontal view. Ni and Caplier [Ni and Caplier, 2011] applied a forward formulation of entropy function to estimate all transformation parameters at the same time rather than sequentially and later [Ni et al., 2012].

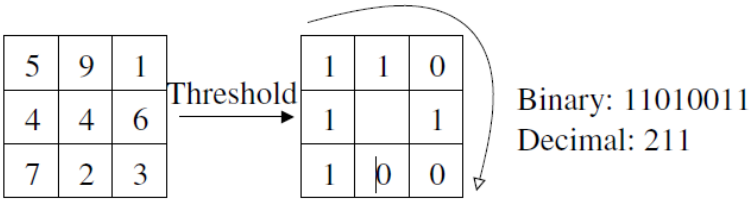


FIGURE 2.9: The basic LBP operator (taken from [Ahonen et al., 2004]) .

They improved this approach by switching the role of template and test data in order to reduce the computational cost in re-calculating the Jacobian and Hessian.

Peng et al. [Peng et al., 2010] proposed an approach called RASL that aligns images by sparse and low-rank decomposition. RASL is able to handle both spatial misalignment and corruptions (e.g., occlusion and shadows).

2.2.3 Feature Extraction

The simplest form of representing a face is to use the pixel intensity values directly [Turk and Pentland, 1991]. However, because of its simplicity, this method is unable to capture complex textures and to handle the large variations which are generally found in human faces. Another limitation is that the feature vector is often in a very high dimensional space.

One of most used features are Local Binary Pattern (LBP). They are based on the idea that small patterns of qualitative local gray-level differences contain a great deal of information about higher-level image content. The LBP operator was first introduced by Ojala [Ojala, 1996] as feature for texture classification. It takes a local neighbourhood around each pixel, thresholds the pixels in the neighbourhood by the value of the central pixel and considering the result as a binary number (Figure 2.9 shows an example). The original LBP operator was fixed to a 3×3 neighbourhood, but Ojala et al. [Ojala et al., 2002] proposed a multi resolution and rotation invariant version.

A limitation of LBP is that a small change in the input image can change the operator output and this is especially a problem in near uniform areas. In order to deal with this problem the authors in [Tan and Triggs, 2007] introduced a three-level operator called Local Ternary Patterns (LTP). In ternary encoding, the difference between the centre pixel and a neighbouring pixel is encoded by three values (1, 0 or -1) according to a threshold τ. The ternary pattern is divided into two binary patterns taking into account its positive and negative components. Hussain et al. [ul Hussain and Triggs, 2012, ul Hussain et al., 2012] proposed an extension of LTP called Local Quantized Patterns (LQP), that uses a lookup-table-based codebook to code larger or deeper patterns than other LTPs. The codebook is learned with a count-weighted version of k-Means. Several extensions have been proposed over the years with a more detailed review of LBP features and it extensions to be found in [Pietikaeinen et al., 2011].

Another method that has been successfully used in face recognition is Gabor filtering. Gabor filters are linear filters which allow description of spatial frequency structures in the image while preserving information about spatial relations which are known to be robust to some variations (e.g. pose and facial expression changes [Jin and Ruan, 2009]). Usually 40 filters (5 scales and 8 orientations) are used in face recognition applications. For each Gabor filter, one value is computed at each pixel position. In practice, the Gabor filters are often used as pre-processing step of LBP [Nguyen et al., 2009, Nguyen and Bai, 2010, ul Hussain et al., 2012] where LBP features are applied on the Gabor filtered images.

Scale-Invariant Feature Transform (SIFT) [Lowe, 1999] features are also often used. They were originally designed to detect and describe interest points in images. These features are invariant to image scale and rotation and robust to changes in illumination, noise and minor changes in viewpoint. SIFT features were directly used [Ozkan and Duygulu, 2006] to match interest points between faces and then used to compute a matching score by averaging the Euclidean distance between matched SIFT descriptors, as well as face descriptors [Cao et al., 2013] that combine several SIFT descriptors into a final face descriptor.

Recently, some algorithm have been proposed [Huang et al., 2012a, Sun et al., 2013, Taigman et al., 2014] that are different to hand-crafted features such as

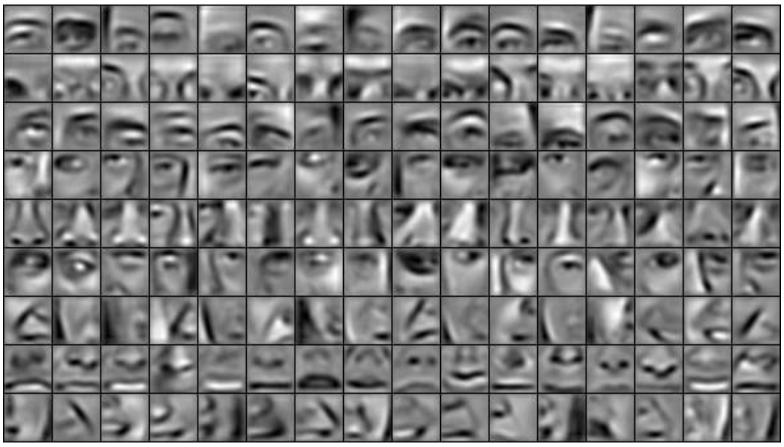


Figure 2.10: Visualization of sample filters from the second layer local CRBM. (taken from [Huang et al., 2012a]).

LBP, Gabor filter and SIFT try to learn a feature representations using deep learning (see Figure 2.10). Deep learning is a set of algorithms that attempt to model high-level abstractions in data by using architectures composed of multiple non-linear transformations.

2.2.3.1 Metric Learning

Several distance or similarity metric learning algorithms have been proposed over the last years, whose aim is to learn a metric so that the distance\similarity between positive face pairs is reduced\enlarged and that of negative pairs is enlarged\reduced as much as possible (see Figure 2.11 for a sample distribution of distances).

Most of the existing work on metric learning focus on the Mahalanobis distance learning, which is defined for any $x_1, x_2 \in \mathbb{R}^d$, by $d_M(x_1, x_2) = (x_1 - x_2)^T M(x_1 - x_2)$, where M is a positive semi-definite (p.s.d) matrix. One early work was presented by Xing et al. [Xing et al., 2002] which learns a Mahalanobis distance metric for k-means clustering. Their algorithm aims to minimize the sum of squared distances between similarly labelled inputs, while maintaining a lower bound on the sum of distances between differently labelled inputs. Goldberger et al. [Goldberger et al., 2004] proposed a method for learning a Mahalanobis distance metric for the K-Nearest Neighbour (KNN) classification algorithm. The algorithm directly

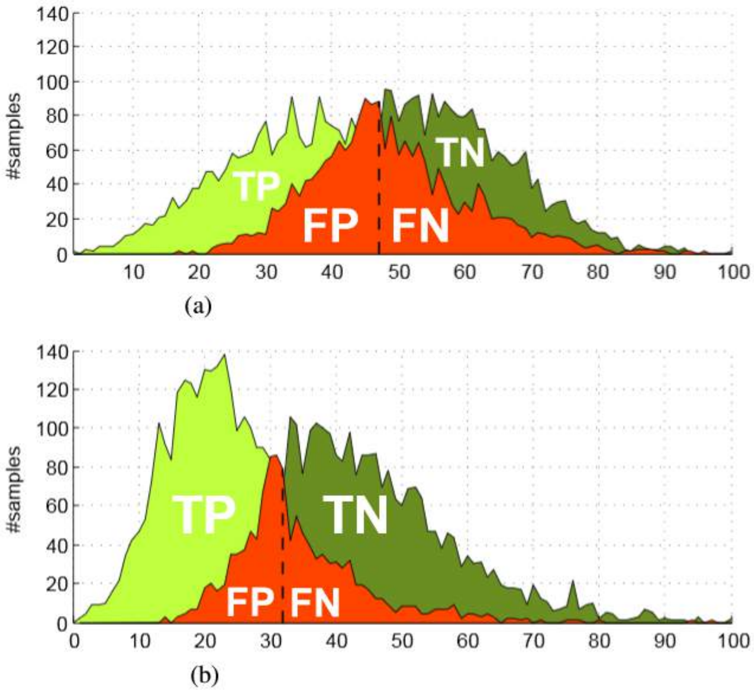


FIGURE 2.11: Distribution of distances before (a) and after (b) applying KISS metric learning (taken from: [Koestinger et al., 2012]).

maximizes a stochastic variant of the leave-one-out KNN score on the training set. Another algorithm that aims to improve KNN classification is presented by Weinberger et al. [Weinberger et al., 2006]. Their algorithm attempts to increase the number of neighbours sharing the same label by learning a linear transformation that minimises the distance between examples having matching labels and maximising the distance between examples with non-matching labels. Later [Weinberger and Saul, 2008], the method was extended in terms of scalability and accuracy.

Davis et al. [Davis et al., 2007] used an information-theoretic approach to learn a Mahalanobis distance metric. Their method exploits the relationship between multi-variate Gaussian distributions and the set of Mahalanobis distances. The general idea is to formulate the problem of learning an optimal distance metric as that of learning the optimal Gaussian with respect to an entropic objective. Guillaumin et al. [Guillaumin et al., 2009] used a probabilistic model to learn a Mahalanobis distance metric. They defined the a posteriori class probabilities as (dis)similarity measures and used maximum log-likelihood to optimize the parameters of the model. In [Koestinger et al., 2012], a method was presented to learn a Mahalanobis distance metric based on a statistical inference scheme.

A different metric has been proposed by Nguyen and Bai [Nguyen and Bai, 2010] that uses Cosine Similarity (GS) instead of Mahalanobis distance, where cosine similarity is defined as $CS_M = (x_1, x_2) = x_1^T M x_2 / (\sqrt{x_1^T M x_1}\sqrt{x_2^T M x_2})$. They proposed a gradient-based optimization algorithm for learning a distance metric based on the cosine similarity. Huang et al. [Huang et al., 2012b] showed how this metric can be learned with a linear SVM. Recently, Cao et al. [Cao et al., 2013] proposed a Generalized Similarity (GS) metric $f_{(M,G)}(x_1, x_2) = s_G(x_1, x_2) - d_M(x_1, x_2)$, where d_M is the Mahalanobis distance and s_G is the bilinear similarity function defined by $s_G = x_1^T G x_2$. They also proposed a regularization framework that learns the similarity metric over the intra-personal subspace.

2.2.4 Video based Face Recognition

Stallkamp et al. [Stallkamp et al., 2007] applied block-based Discrete Cosine Transform (DCT) to non-overlapping blocks and used KNN to determine the nearest neighbour in the training set. In order to get the final classification result, they combined the classification result of each frame by a weighted contribution function.

Zhao and Pietikainen [Zhao and Pietikainen, 2007] proposed a spatio-temporal representation of the LBP feature called Volume Local Binary Patterns (VLBP). The idea behind this feature is to combine motion with appearance features in order to be robust to grey scale changes, rotations and translations related to intra-personal variations. The VLBP feature looks at a face sequence as a rectangular prism (or volume) and defines the neighbourhood of each pixel in 3D space. In [Hadid and Pietikaeinen, 2009] the Extended Volume Local Binary Pattern (EVLBP) was proposed, which extends the VLBP by allowing a flexible number of points at each frame and by increasing the number of frames which are included from three to five. In addition, AdaBoost is used to automatically determine the optimal size and locations of the local rectangular prisms, and for selecting the most discriminative EVLBP patterns for recognition. Mendez-Vazquez [Mendez-Vazquez et al., 2013] further developed this approach by proposing an additional scale parameter which allows to encode the local spatio-temporal information by comparing neighbouring regions at different scales in neighbouring frames.

Cevikalp and Triggs [Cevikalp and Triggs, 2010] represented face feature vectors in a linear or affine feature space and characterize each set by a convex geometric region (the affine or convex hull spanned by its feature points). Set dissimilarity is measured by geometric distances (distances of closest approach) between convex models. Hu et al. [Hu et al., 2011] represented a face set as a triplet: a number of samples, their mean and an affine hull model. The dissimilarity of two sets is measured as the distance between their nearest points. In [Wu et al., 2012b], a set based discriminative ranking model was proposed which iterates between set-to-set distance finding and discriminative feature space projection.

Wolf et al. [Wolf et al., 2011] used a SVM based set-to-set similarity called Matched Background Similarity (MBGS). For each video, MBGS selects the nearest background feature samples from a set of background samples. Then a SVM is trained for each video where the video samples are defined as positive and the background samples as negative. Afterwards, each frame in the first video is classified by the SVM of the second video and vice versa. The mean classification source then gives the final result. This approach was further developed in [Wolf and Levy, 2013] with a SVM variant called SVM-minus, which tries to "unlearn" the separation induced by pose. Besides the MBGS, Wolf et al. [Wolf et al., 2011] also proposed several other methods such as mean Euclidean distance, distance between most frontal faces or maximum correlation, however none reached the same results as MBGS.

Li et al. [Li et al., 2013a] took a part based representation for a face track by extracting local features (e.g., LBP or SIFT) from densely sampled multi-scale image patches. Each local feature is augmented with its location, then a Gaussian mixture model (GMM) is trained to capture the spatial-appearance distribution of all face images in the training set. Zhu et al. [Zhu et al., 2013] proposed a set-to-set metric that models a set by an affine hull. They formulated the set-to-set metric learning problem as a sample set pair classification problem. Each sample pair is characterized by the covariance matrix of its two samples sets. A discriminative function is then proposed for sample pair classification, and finally the problem is solved by using a SVM model.

2.3 Open Research Issues

This chapter has discussed a range of methods that have been applied to tracking and face recognition. Some of the discussed methods have achieved promising results on challenging datasets. However, a number of problems and research challenges remain unsolved or unaddressed. A brief overview of these research issues and open problems is given in the following.

2.3.1 Tracking

Significant progress has been made in object tracking during the last few years. Several robust tracking systems have been proposed which can track objects even in more challenging scenarios. However, it is clear that several challenges still remain. One important challenge in tracking is to develop algorithms for tracking humans in crowded scenes. In these scenes there are usually severe occlusion, and people are only partially visible. Algorithm are needed that are able to handle partial occlusions in both the detection and the tracking stages.

Another challenge is the treatment of false-positive detections, especially in more challenging scenarios where different object types such as animals, cars or bicycles are present tracking can be difficult. One interesting solution in this context was presented by Benfold and Reid [Benfold and Reid, 2011] who approached this problem by creating a separate model for false-positives and combine the identification of false positives with the data association. However, they assumed that false positives are non-moving background objects which limit their approach.

In general, an important issue that has been neglected in the development of tracking algorithms is the high computational cost, most existing tracking-by-detection systems are unsuitable for real-time systems. Particularly in scenarios where high definition videos or multi-cameras are used is this a very challenging problem. The detection stage is the main bottleneck, and corresponding strategies are needed to address this issue.

2.3.2 Face Recognition

It can be seen that there are promising methods for face recognition. While the majority of approaches focuses on still image recognition, more video-based approaches are being proposed. Although the results provided by recent methods are improving, accurate video face recognition remains challenging. Videos are often of low resolution and contain faces in non-frontal pose or partly occluded. However, it also has the benefit of providing a setting in which weak evidence in a single frame can be integrated over multiple frames to achieve a more accurate result.

In order to exploit this additional information it is necessary to create novel strategies that are capable of comparing image-sets and to incorporate spatio temporal informations into the decision making process. In order to do so, it is not sufficient to merely improve the recognition task but it is also necessary to develop a robust framework that includes detecting, tracking and alignment. The most current work focuses only on one of these tasks and ignores the others.

In addition, several improvements for still-image recognition have been proposed such as novel features or novel metrics which possibly could also be exploited for video face recognition.

2.4 Metrics

This section describes the metrics that are used in this thesis during evaluation.

2.4.1 PASCAL measure

In order to judge if detections are true/false positives, the PASCAL measure [Everingham et al., 2010] is used. A detection is considered correct if the overlap ratio between a detection R^d and a ground truth detection R^{gt} exceeds a threshold τ:

$$a_o = \frac{|R^d \cap R^{gt}|}{|R^d \cup R^{gt}|} > \tau \tag{2.5}$$

Each detection may be matched at most once. Any assignment ambiguity is resolved by matching detections with highest confidence first [Dollár et al., 2012].

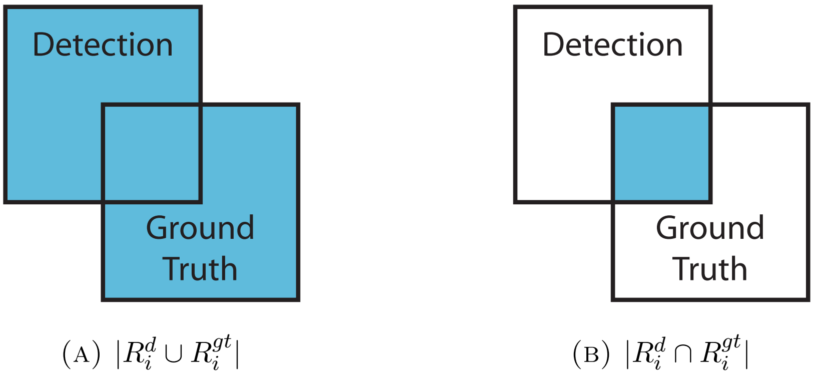
Following the methodology of [Dollár et al., 2012], the performance is then summarised using the log-average Miss Rate (MR) computed by averaging miss rate at nine False Positives Per Image (FPPI) rates evenly spaced in log-space in the range 10^{-2} to 10^0. For curves that end before reaching a given FPPI rate, the minimum miss rate achieved is used. The log-average miss rate is similar to the performance at 10^1 FPPI but in general gives a more stable and informative assessment of performance [Dollár et al., 2012].

2.4.2 CLEAR MOT metrics

The tracking performance is evaluated using the four CLEAR MOT metrics [Bernardin and Stiefelhagen, 2008], the Multiple Object Tracking Precision (MOTP), the Multiple Object Tracking Accuracy (MOTA) and the detection precision and recall. The MOTA is a combined measure which takes into account false positives, false negatives and identity switches:

$$MOTA = \frac{n_{fn} + n_{fp} + n_{mme}}{n_{gt}} \tag{2.6}$$

where n_{fn}, n_{fp}, n_{mme} and n_{gt} are the number of false negatives, of false positives, of mismatches and of ground truth detections. The MOTP measures the precision



(A) $|R_i^d \cup R_i^{gt}|$

(B) $|R_i^d \cap R_i^{gt}|$

FIGURE 2.12: The overlap between the detection and the ground truth rectangle is the union area (A) divided by the intersection area (B).

with which objects are located using the intersection of the estimated region R^d with the ground truth region R^{gt} (see Figure 2.12):

$$MOTP = \frac{1}{n_{tp}} \sum_{i=1}^{n_{tp}} \frac{|R_i^d \cap R_i^{gt}|}{|R_i^d \cup R_i^{gt}|} \tag{2.7}$$

where n_{tp} is the number of true positives. The precision is the fraction of retrieved instances that are relevant:

$$Prec = \frac{n_{tp}}{n_{tp} + n_{fp}} \tag{2.8}$$

and the recall is the fraction of relevant instances that are retrieved:

$$Rec = \frac{n_{tp}}{n_{gt}} \tag{2.9}$$

Chapter 3

Real-time Video-based Face Tracking and Recognition in Smart Homes

3.1 Introduction

The LPH smart home project focuses on the creation of an environment that acts as an intelligent agent, perceiving the state of the home and its dweller through sensors and acting upon the environment through device controllers. In order to allow comfortable usage and to treat every user individual, it is essential that the computer system follows human interaction patterns and simultaneously is able to identify the humans it is interacting with.

One way of identifying humans is face recognition. It allows identification without user interaction. Furthermore a camera system in which the camera locations are known allows the system to determine the location of a person. This allows a smart home to interact with a human in a natural way.

Face recognition from video has received extensive attention in recent years [Wang et al., 2009, Zhang et al., 2011]. Despite recent research, accurate face recognition remains challenging. Several problems still remain unsolved such as illumination changes, pose variation, and occlusion. Unlike still image face recognition, video-based face recognition provides a setting in which weak evidence in a single frame can be integrated over a set of frames in order to achieve more accurate result.

In this chapter, a novel face tracking and recognition framework is presented. Therefore, a new algorithm for building face tracks in real-life scenarios is proposed, that combines face detection and optical flow tracking to a face tracking algorithm. The temporal information which is provided by the face tracks is used to significantly improve the recognition rate of LBP face recognition [Ahonen et al., 2004] in video scenes.

The proposed approach is evaluated on the video face recognition dataset Honda/ UCSD Video Database (HUVD) [Lee et al., 2003, Lee et al., 2005] and a novel dataset that simulates the scenario of a smart display in a normal household. The test shows that the proposed approach operates in real-time and can handle illumination changes, occlusions, and out-of-plane rotations.

The rest of this chapter is organized as follows. The details of the proposed approach are provided in Section 3.2. In Section 3.3, experimental evaluation is presented. Section 3.4 concludes the paper with a brief summary and a discussion of the results.

3.2 Proposed Approach

The proposed probabilistic face recognition framework focuses on the utilization of information from a tracked face to infer the complete information of the tracked object. Instead of trying to recognise each frame separately, the most likely identity for each track is calculated. This is done by recognising each face image in a track with Local Binary Patterns (LBP) face recognition and then calculating the maximum over the track.

3.2.1 Tracking

Tracking is used to exploit the temporal information of a video scene for face recognition (see Section 3.2.3) and to estimate the position of a face when the face detector cannot detect it. In order to build face tracks $T = \{T_1, T_2, ..., T_i\}$, a face detector is first applied on individual video frames I_t and then the obtained detections $D_t = \{d_{t1}, d_{t2}, ..., d_{tj}\}$ are linked into tracks. The Viola Jones face detector [Viola and Jones, 2002] is used in order to detect faces in a frame (a face is described by a bounding box). In order to link the detections into face tracks, a visual tracker is utilised to estimate the location of each face bounding box from the last frame I_{t-1} in the current frame I_t. In particular, a tracker based on Lucas-Kanade optical-flow [Lucas and Kanade, 1981, Kalal et al., 2010] is chosen.

Therefore the face bounding box is initialized with a grid of points, then the location of each point in the next frame is computed with Lucas-Kanade optical-flow (see Figure 3.1). In order to be robust to outliers, the forward-backward error [Kalal et al., 2010] of the resulting points is calculated. The calculation of this error proceeds as follows: the resulting points are back-projected to the last frame I_{t-1} and the Euclidean distance between the original points and their back-projection is calculated. As an additional measurement, the Normalized Cross-Correlation (NCC) [Lewis, 1995] between the original points and their projection is calculated:

$$NCC(P_{t-1}, P_t) = \frac{\sum_{u,v}(P_{t-1}(u,v) \cdot P_t(u,v))}{\sqrt{\sum_{u,v} P_{t-1}(u,v)^2 \cdot \sum_{u,v} P_t(u,v)^2}} \tag{3.1}$$

where P_{t-1} and P_t are patches around the points with a fixed size of 15×15 pixel. Afterwards, all points that have a forward-backward error higher than the mean error are eliminated. The remaining points are then used to predict the position of the bounding box d_{t-1j} in I_t, with the predicted bounding box denoted as $\hat{d}_{tj}$.

In order to assign a face to a track, the overlap between the predicted bounding box $\hat{d}_{tj}$ of each track and each face detection $D_t = \{d_{t1}, d_{t2}, ..., d_{tj}\}$ is calculated. If the overlap is higher than a threshold τ_1, the face is considered as part of the track and the track bounding box is set to the face bounding box.

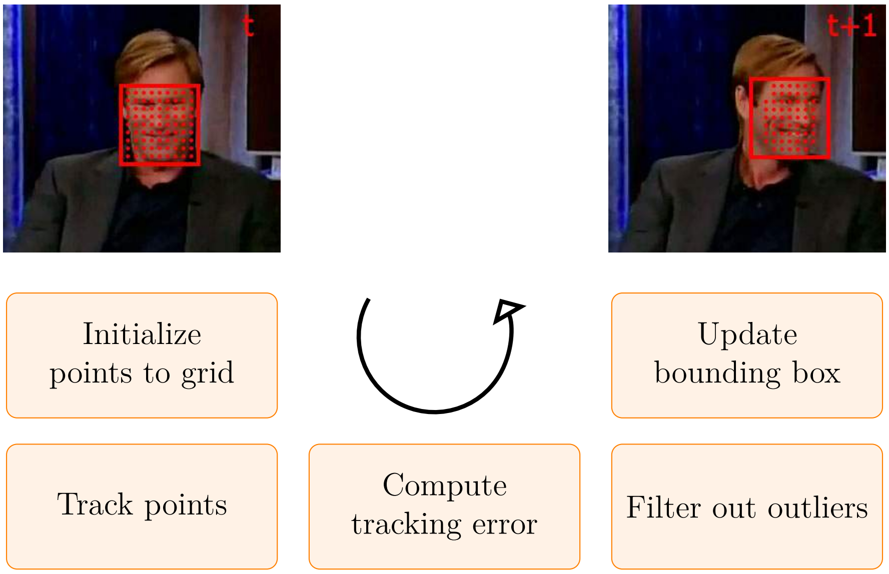


Figure 3.1: Lucas-Kanade optical-flow face tracking.

$$|\hat{d}_{tj} \cap d_{tj}| \geq \tau_1 \tag{3.2}$$

If a face cannot be assigned to a track and the face does not overlap with any track, a new track is initialised for the face. If a track has no overlap with a face and the mean NCC error of all points that were used to predict the bounding box is under a threshold τ_2, it is assumed that the tracker has failed and the track is defined as inactive. If a track is inactive for more than n frames, it is deleted.

3.2.2 Reinitialisation of inactive Tracks

If the tracker has failed, it is likely that the face has been occluded by other objects. In this section a novel approach is proposed in order to reinitialise tracks if the face is visible again, by linking them to detection responses based on a link probability. The link probability between the last detection of an inactive track T_i and a detection $d_{t,j}$ is defined as the product of two probabilities based on location $p_l(d_{t,j}|T_i)$ and appearance $p_a(d_{t,j}|T_i)$:

$$p(d_{t,j}|T_i) = p_l(d_{t,j}|T_i)p_a(d_{t,j}|T_i) \tag{3.3}$$

An inactive track T_i and a detection $d_{t,j}$ are linked if the detection is not assigned to an active track, if the link probability is higher than a threshold τ_3 and if the link probability is significantly higher than the link probability of any other track $(T \cup T_i)$ including active tracks:

$$p(d_{t,j}|T_i) > \tau_3 \tag{3.4}$$

$$p(d_{t,j}|T_i) - \underset{T_k \in T \cup T_i}{\arg\max}\, p(d_{t,j}|T_k) > \tau_4 \tag{3.5}$$

The location probability is based on the estimated location of the inactive track at the current frame I_t. In order to estimate the location, a recursive estimator is used. This means that only the last state from the inactive track is needed to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required. In what follows, the notation $\hat{x}_{t|m}$ represents the estimate of the location x at time t given observations up to, and including at time m:

$$\hat{x}_{t|m} = x_m + \Delta t \cdot v \tag{3.6}$$

with $\Delta t = t - m$. The velocity v has been estimate before, as long as the track was active, using an interpolate model that estimates the velocity using a weighted sum rule:

$$v = (1 - \alpha) \cdot v + \alpha \cdot v_t \tag{3.7}$$

with

$$v_t = x_t - x_{t-1} \tag{3.8}$$

and α being the update rate.

The difference between the predicted location $\hat{x}_{n|m}$ and the observed location x is assumed to follow a normal distribution with zero mean and variance of σ_l^2:

$$p_l(d_{t,j}|T_i) = e^{\frac{-||\hat{x}_{n|m} - x||^2}{\Delta t^2 \sigma_l^2}} \tag{3.9}$$

The appearance distance is defined as the distance between the face patch $s_{t,j}$ of the detection d_{tj} and the nearest neighbours of the set of face patches from the inactive track $S_i = \{s_{i1}, s_{i2}, ..., s_{in}\}$. The probability of the appearance distance is then modelled by a normal distribution with zero mean and variance of σ_a^2:

$$p_a(d_{t,j}, T_i) = \underset{s_{in} \in S_i}{\arg\max}\, e^{-\frac{f(s_{t,j}, s_{in})}{\sigma_a^2}} \tag{3.10}$$

where $f(\cdot, \cdot)$ is a distance function. In this thesis the Chi square statistic χ^2 is used (details about the Chi square statistic can be found in Section 3.2.3).

Since this approach is used in an online setting with a theoretically infinite number of face patches, it is impossible to store all these patches. Therefore it is assumed that only k patches can be retained. The best subset S^* of patches from a set S_i is selected by maximizing the feature space that is spanned by the subset:

$$S^*_{max} = \underset{\substack{S^* \subset S_i \\ |S^*| = k}}{\arg\max} \sum_{i=1}^{|S^*|} \sum_{j=i+1}^{|S^*|} f_{x^2}(s_i, s_j) \tag{3.11}$$

In order to update the subset sequentially at each frame with a new face patch s_{new}, the following online algorithm for selecting patches is proposed:

```
Input          : $S^*_{i,t-1}, s_{new}, k$
Output         : $S^*_{i,t}$
Initialization: $S^* \leftarrow S^*_{t-1} \cup s_{new}$
if $|S^*| > k$ then
    compute $S^*_{max}$ with Equation 3.11;
    $S^*_{i,t} = S^*_{max}$;
else
    $S^*_{i,t} = S^*$
end
```

Algorithm 1: Face patch set update algorithm.

The parameters used for tracking such as (τ_1 - τ_4, σ_l, σ_a) were estimated on the trainings set using Maximum-Likelihood Estimation [Myung, 2003].

3.2.3 Face Recognition

In this section a video face recognition algorithm is proposed. The algorithm extends the still image face recognition approach of Ahonen et al. [Ahonen et al., 2004] for video face recognition. Their approach first divides a face image into 4×4 regions from which the Local Binary Pattern (LBP) features are extracted and then concatenated into a single histogram (see Figure 3.2). The recognition is then performed by finding the nearest neighbour in a test set. To calculate the distance between two histograms x, y, Ahonen et al. [Ahonen et al., 2004] proposed the Chi square statistic χ^2:

$$\chi^2(x, y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i} \tag{3.12}$$

LBP face recognition was designed for still image face recognition. There are several disadvantages in its use for video-based face recognition. Firstly, faces extracted from videos have usually very low quality, the noise levels are high and

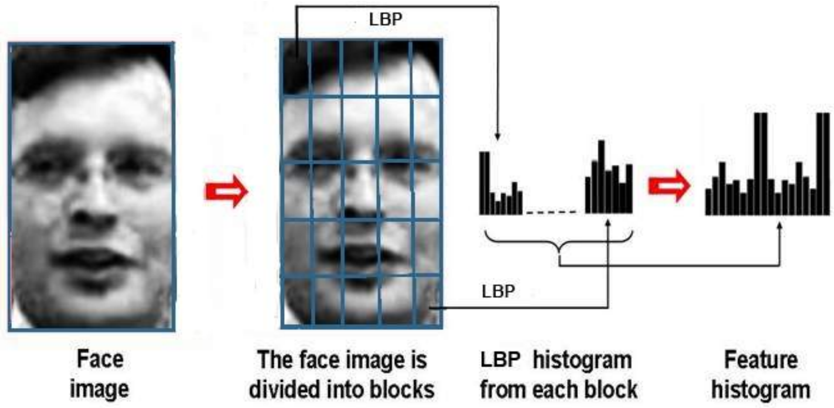

LBP
LBP
Face image
The face image is divided into blocks
LBP histogram from each block
Feature histogram

FIGURE 3.2: Example of an LBP based facial representation.

the illumination may change through the video. Secondly, face images variations such as expression, view and occlusion are higher. Furthermore the size of a face (in pixel) changes from frame to frame depending on the distance between the face and the camera. Despite these drawbacks videos have the advantage of providing temporal information. In order to exploit this for face recognition, an algorithm is proposed that maximises the recognition results over a track.

Suppose there is a dataset of LBP feature histograms $s_1, s_2, ..., s_m$ where each feature histogram is associated with an identity $ID_1, ID_2, ..., ID_n$ and a set of r faces that are assign to the track T_i. For each face in the track, the LBP feature histograms $f_1, f_2, ..., f_r$ are calculated. In order to determine the identity of the person represented in the track T_i, the identity of each face in the track is first determined by calculating the nearest neighbour in the database:

$$ID_{ij} = \underset{i=1,2...m}{\arg\min}\, \chi^2(f_j, s_i) \tag{3.13}$$

and then the identity of the track ID_i is calculated by determining the most frequently occurring ID in the track:

$$ID_i = \underset{i=1,2...n}{\arg\max} \sum_{j=0}^{r} S(ID_{Tj} = ID_i) \tag{3.14}$$

with

$$\begin{aligned} S(true) &= 1 \\ S(false) &= 0 \end{aligned} \tag{3.15}$$

In order to reduce false positives, the minimum length of a track is set to 5. If a track contains less than 5 faces, no recognition is performed. In order to compensate for the problem of different views, multiple training samples were used with a wide range of views. These views usually cover almost all possible views and an example can be seen in Figure 3.4. An overview of the proposed framework is shown in Figure 3.3.

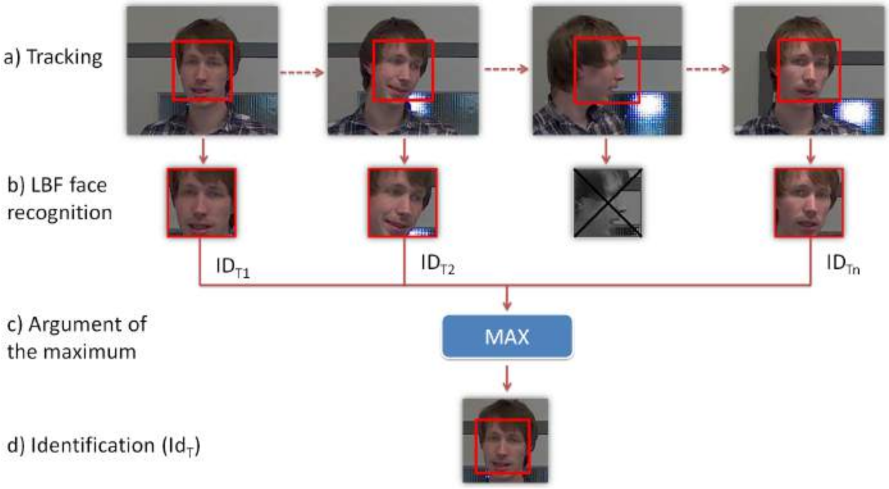


Figure 3.3: A overview over the proposed framework.

3.3 Experimental Setup and Evaluation

In this section, the experimental evaluation of the proposed approach is discussed in detail. The algorithm is first evaluated on the HUVD and then on a new dataset. In order to learn the tracking parameters leave-one-out cross-validation is used. One round of cross-validation involves partitioning the data into two subsets, performing the training on the first subset (called the training set) and the evaluation on the second subset (called the testing set). In order to reduce variability, multiple rounds of cross-validation are performed using different partitions and the evaluation results are averaged over the rounds. Leave-one-out cross-validation is a particular case of cross-validation where the validation set size is fixed to one. All tracking threshold and parameters are then learned from the trainings set.

3.3.1 Honda/UCSD Video Database (HUVD)

The HUVD is split into two subsets: the first contains 20 subjects [Lee et al., 2003] and the second contains additional 15 subjects [Lee et al., 2005]. Each video sequence is recorded indoors and contains about 300-600 frames with a resolution of 640 × 480 pixels per frame. Every individual is recorded in at least two video sequences. All the video sequences contain significant 2-D (in-plane) and 3-D (out-of-plane) head rotations. In addition, some of these sequences contain difficult events such as partial occlusion, the face partly leaving the field of view and large scale changes.

The training set is used to build a LBP feature histogram database for each subject. In order to extract the face image of a subject from the training sequences, a Viola Jones face detector [Viola and Jones, 2002] was applied to each frame. Figure 3.4 shows a sample of face images for a subject.

To determine the accuracy of the proposed algorithm, the number of frames in which a given face was correctly recognized is divided by the total number of frames in the test videos. The first 5 frames were ignored because the proposed algorithm needs at least a history of five frames to calculate an identity (see Section 3.2.3).

In order to show the advantage of using temporal information, the proposed algorithm (LBPFHist) was compared to an algorithm with a conventional LBP face recognition approach (LBPF). Therefore the same algorithm was used but the history length was set to 1. In addition the previous results of Mian [Mian, 2008] and Kim et al. [Kim et al., 2008] (see Table 3.1) were included.

The results show that the proposed algorithm has state-of-the-art performance. However, the originality compare to Mian [Mian, 2008] and Kim et al. [Kim

Table 3.1: Test results for Honda/UCSD Video Database

Approaches	**Dataset**	**Accuracy**
LBPF	[Lee et al., 2003] + [Lee et al., 2005]	76.34%
LBPFHist (proposed)	[Lee et al., 2003] + [Lee et al., 2005]	100%
Mian [Mian, 2008]	[Lee et al., 2003]	99.5%
Kim et al. [Kim et al., 2008]	[Lee et al., 2003]	100%

et al., 2008] is that a complete system was proposed that performs automatic detection, tracking and online recognition in real-time. In contrast Mian [Mian, 2008] algorithm ignores tracking and Kim et al. [Kim et al., 2008] algorithm does tracking but needs manual initialisation which makes both algorithm not applicable in real life sceneries.

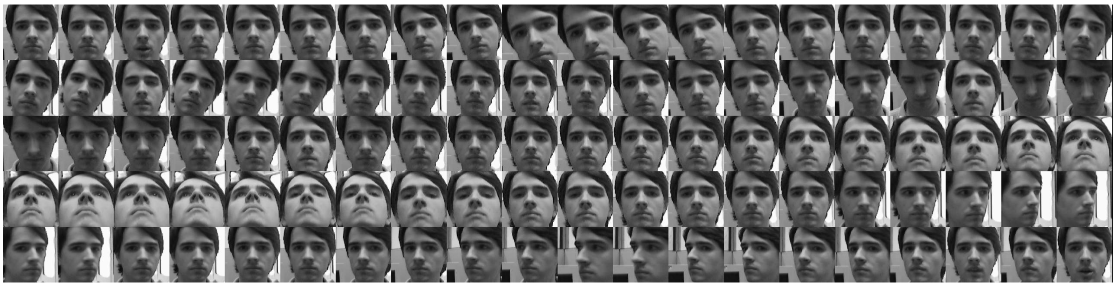


Figure 3.4: Sample faces of a subject taken for the Honda/UCSD Video Database

3.3.2 Smart Home Dataset

Since the proposed algorithm gives essentially perfect results on the HUVD, a new video face recognition dataset was created, called Smart Home Dataset (SHD). The dataset (SHD) simulates the scenario of a smart display in a normal household where people appear and disappear in front of the display. It contains 5 subjects and consist of a training and a test set. The training set consists of an approximately 300 frames frontal view video for each subject. In order to make the recognition task more challenging, the training dataset is combined with the HUVD, so that there are 40 subjects in the database. The test set consists of 7 videos with 1-4 different subjects, a resolution of 1280 × 800 pixels per frame and a length between 1800 to 5400 frames. For each subject, ground truth trajectories were manually annotated. The trajectories begin when a face becomes visible and end when it disappears. A face is defined as visible when the face is not occluded and the view differs not more than 80° from the frontal view. The challenges in this dataset include inter-subject occlusions, and significant 2-D (in-plane) and 3-D (out-of-plane) head rotations.

The evaluation was performed as follows. The proposed face tracking and recognition algorithm were used to calculate the track of the subject. Then all tracked and recognised frames were compared to the ground truth. The results were deemed correct if the bounding box of the ground truth and that of the algorithm overlap more then 25%. In addition the false positive rate was calculated. The results are presented in Table 3.2 and some sample images can be seen in Figure 3.5. It can be seen that the proposed algorithm LBPFHist (91.19% accuracy), which uses temporal information, significantly outperforms the algorithm LBPF (65.90% accuracy) which does not use this temporal information. The large difference between the results on different videos can be explained by different difficulty levels; for example in the video Office2 are significantly more occlusion then in the video Office1.

Table 3.2: Test results Smart Home dataset

Video	**LBPFHist (proposed) Accuracy**	**LBPF Accuracy**
Office1	97.88%	66.65%
Office2	77.22	52.18%
TV	99.62%	82.45%
Home1	93.68%	59.45%
Home2	96.76	77.47%
Home3	83.65%	57.21%
Overall	**91.19%**	**65.90%**

3.3.3 Performance

The test system is a desktop computer with an Intel Core i5-3470 CPU with 3.2 GHz and 8GB RAM. The algorithm ran under Windows and was written in C++. The OpenCV Viola Jones face detector implementation CascadeClassifier and the OpenCV Lucas-Kanade optical-flow implementation calcOpticalFlowPyrLK

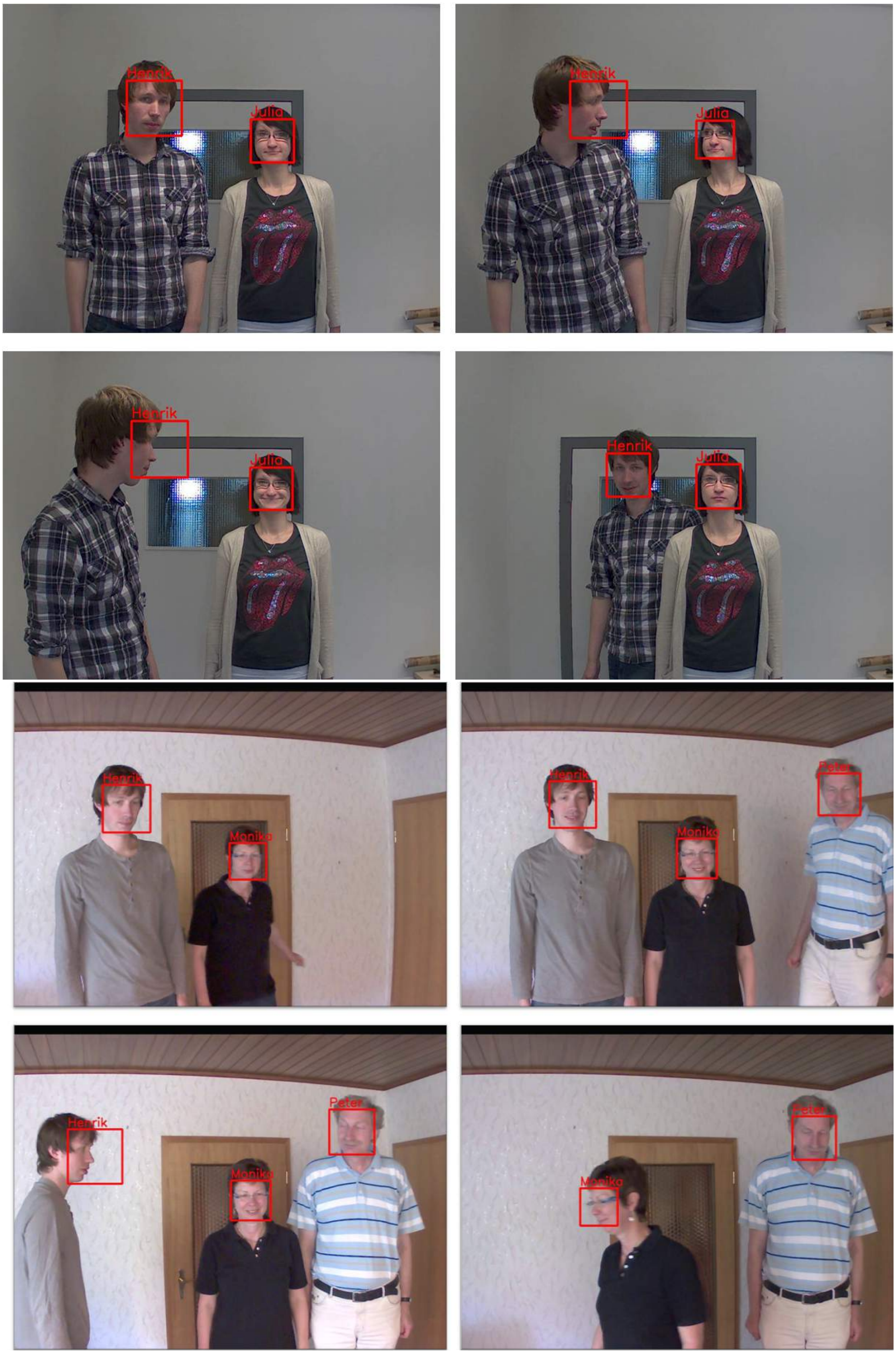


FIGURE 3.5: Sample results of the LBPFHist algorithm on the Smart Home dataset.

were used. On the test system, the test sequences of the HUVD ran with 25 fps and an average processor load of 26% over all 4 cores.

3.4 Conclusion

In this chapter a novel face tracking and recognition framework was presented. A tracking algorithm that combines face detection and optical flow tracking was proposed. The tracking results are used as the basis for LBP face recognition. The system significantly improved the recognition results by exploiting the temporal information provided through the tracking process (recognition rate 76% to 100% compared with a conventional LBP face recognition approach).

The evaluation was performed using the HUVD, where 100% recognition rate was achieved. The test showed that our approach operates in real time and is able to handle illumination changes, occlusions and out-of-plane rotations. Furthermore, the algorithm was evaluated on a new dataset that included multiple people, a wide range of views and constant appearing and disappearing of faces. On the second dataset the algorithm reached a recognition rate of 91%. The test showed that the proposed algorithm can be used for identification and tracking of people in smart homes.

The proposed approach reaches good results for a scenario such as a smart TV. However, in a smart environment such as the LPH a person should be tracked regardless whether the face is visible or not. Therefore, a head detector is presented in the next chapter that allows to detect a person from every view.

Chapter 4

Combining Detectors for Robust Head Detection

4.1 Introduction

Detecting humans is an important task for a wide range of applications, like surveillance, smart environments, or ambient assisted living. In indoor environments such as the LPH as well as in crowded outdoor scenes such as shown in Figure 4.1, only some humans are fully visible (left image); for many others, only the upper-body is visible, or even just the head (right image). Such impediments led previous works such as [Benfold and Reid, 2011] and [Rodriguez et al., 2011] to rely on head detection and ignore the rest of the body. However, robust head detection is difficult to achieve and our experiments indicate that head detection is not as reliable as full body detection (see Section 4.3). These observations motivated us to combine detectors to create a more robust head detector. The idea is not to rely on a single detector. Instead, a head, an upper-body and a body detector are used for decision making by combining their individual scores to derive a consensus decision.

The proposed detectors are based on three new detectors for head, upper-body, and body detection which were trained based on the Aggregated Channel Features

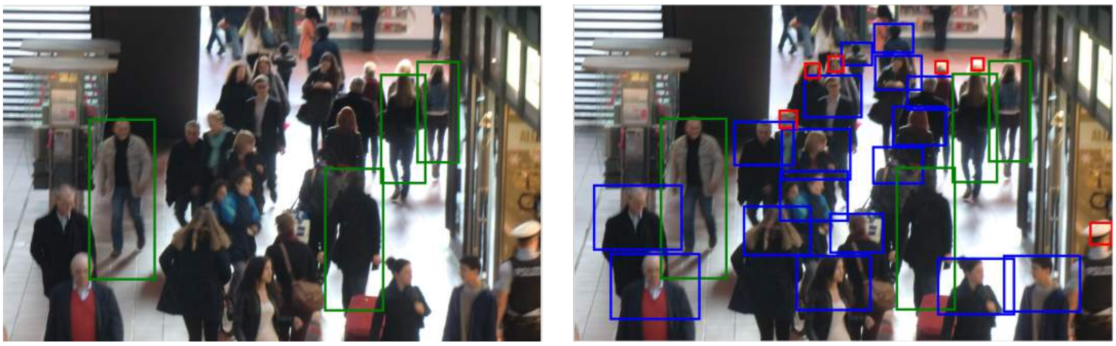


Figure 4.1: Example of a crowd scene.

(ACF) detector framework of Dollár et al. [Dollár et al., 2014]. For each detector, the head is defined as a point of reference that allows to combine the detectors by taking only geometric properties into account. The main principle of combination consists of estimating the head location of each part detector and then grouping detections into disjoint subsets. For confidence score combination, the maximum posterior probability over all parts is computed. In order to validate the findings, the combined detector is tested on the town centre dataset [Benfold and Reid, 2011]. Results showed an 18% reduction in the log-average miss rate of the proposed combined classifier. These results illustrate that combining head detectors may perform better than a single detector.

4.2 Combined Head Detection

The proposed approach consists mainly of three steps. The first step is to train each part detector separately. The second step is to apply the part detectors to a test image and to group the resulting detections. The final step is to compute the combined detection score. In the following section, each step is described in detail.

4.2.1 Detector

The proposed detector is based on three part detectors: a head, an upper-body and a body detector. A separate detector is trained for each part using the ACF detector framework of Dollár et al. [Dollár et al., 2014], which has shown high accuracy on the related task of full-body pedestrian detection.

The ACF detection framework first smooths an input image I with a [1 2 1]/4 triangle filter and then computes several channels $C = \Omega(I)$. The channels are divided into blocks and pixels in each block are summed. The resulting channels are smoothed again with a [1 2 1]/4 triangle filter. Features are single pixel lookups in the aggregated channels. Boosting is used to train and combine decision trees over these features (pixels) in order to distinguish objects from the background. For multi-scale detection a feature pyramid is built. At each scale, a sliding-window approach is then used to detect objects.

In order to train the detector, a novel dataset was created that contains 650 overhead person images (plus horizontal mirror images) from different indoor and outdoor locations. The people are usually standing but appear in any orientation and against a wide range of backgrounds. The head locations were manually annotated (see the red box in Figure 4.2), while the upper-body (green box) and body locations (blue box) were estimated by a fixed ratio (see Figure 4.2). Head images have a size of 16×16 pixels without and of 32×32 pixels with padding, upper-body images a size of 30×41 pixels without and of 60×64 pixels with padding and body images a size of 41×100 pixels without and of 64×128 pixels with padding. In order to align the manual annotated heads, the automatic alignment algorithm of Huang et al. [Huang et al., 2007] was used. This algorithm takes a collation of images from a particular class and automatically aligns these images. Since heads appear in reality under different rotations, alignment was limited to x and y translation and scaling. As negative training set, background images from the INRIA dataset [Dalal and Triggs, 2005] were used. By estimating the upper-body and body locations based on the head location, the resulting detectors are aligned at the head location. That allows more accurate head location estimation

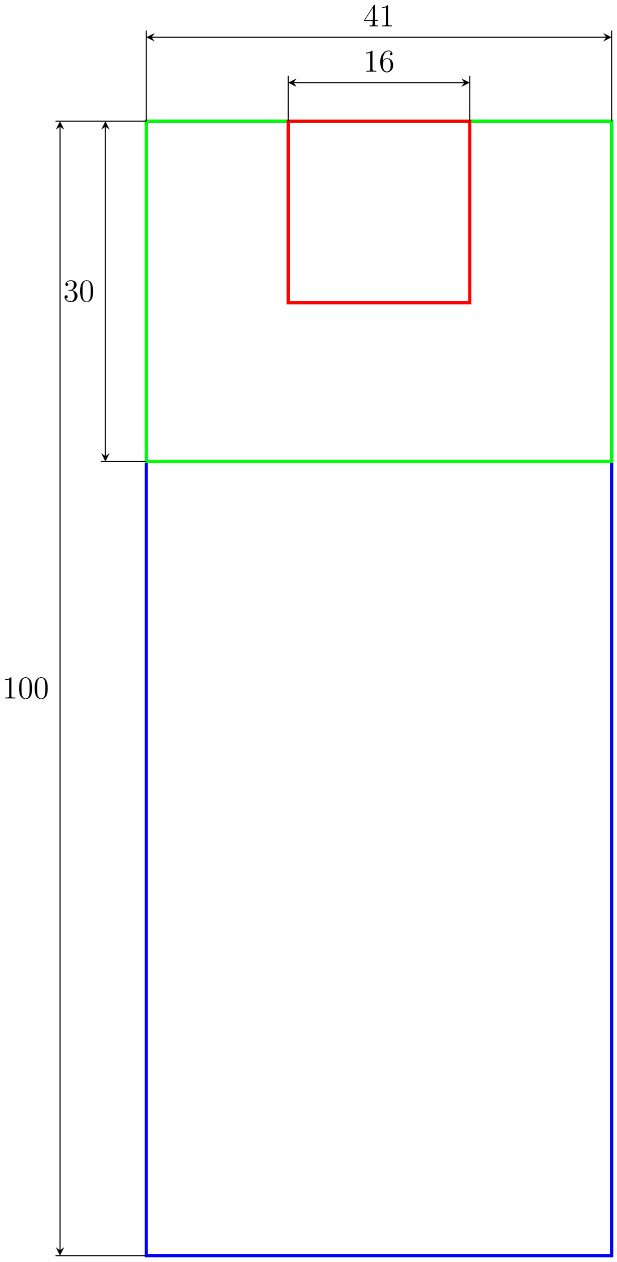


FIGURE 4.2: Annotations estimation ratios.

based on this detector output rather than if the training set were centred at the body or upper-body location as is commonly done [Dalal and Triggs, 2005, Dollár et al., 2012].

For each detector the same configuration was used. Ten feature channels were used: normalised gradient magnitude, HOG (6 channels) and LUV colour channels; the block size was set to 4×4. AdaBoost was used to train and combine 2048 depth-two trees over the candidate features (channel pixel lookups) in each window. The step size of the detectors is set to 4 pixels and 8 scales per octave.

4.2.2 Combining Part Detectors

The final part detectors are then applied across a test image. The same feature pyramid is used for each detector, which speeds up the detection process (see Section 4.3). Then the head location is estimated for upper-body and body detections using the same ratio that was used to build the training set (see Section 4.2.1). In order to return one final detection, detections of different parts have to be combined into a single detection.

In part-based models such as [Felzenszwalb et al., 2010b], the geometric relationship between parts is modelled explicitly. In the developed scheme, this is not necessary since the part detectors are aligned at the head location, so detections can be combined in a very simple way.

The set of all part detections is first partitioned into disjoint subsets. Two detections are in the same subset if their bounding regions overlap more than 50 %. Each partition yields a single final detection. The final bounding box is the bounding box of the most confident head detection (see Section 4.2.3) and if the partition does not include a head detection, then the bounding box of the most confident detection of the remaining parts is used. In order to be able to compute the combined confidence for each detection (see Section 4.2.3), the most confident detection of each part in a partition is saved. It is worth nothing that it is not required that all parts exist, if a partition does not include detections of all parts, the confidence scores of the missing parts is set to zero.

4.2.3 Confidence Combination

The confidence scores of a part detection can be obtained from the boosted classifier H, which consists of K weak classifiers:

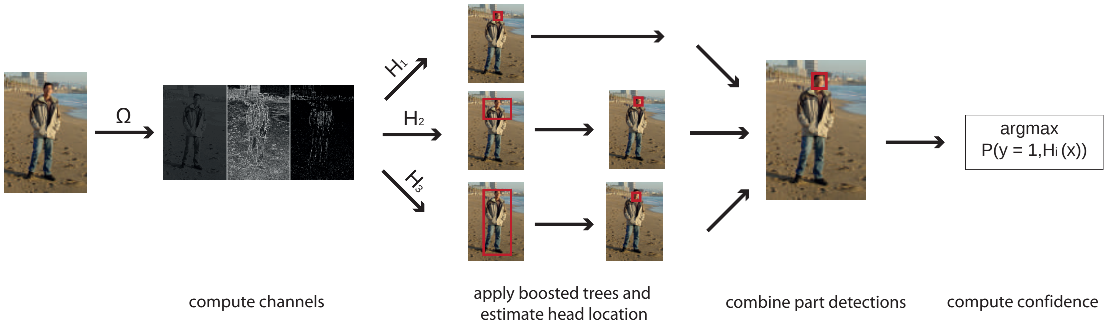


FIGURE 4.3: Overview of the combined head detector.

$$H(x) = H_K(x) = \sum_{i=1}^{K} \alpha_i h_i(x) \tag{4.1}$$

where each h_i is a weak classifier (with output -1 or 1) and α_i is its associated weight; x is classified as positive if $H(x) > 0$ and $H(x)$ serves as a score. When a person is not occluded, our experiments have shown that a body detector is more reliable then a head detector. However, if a person is partly occluded, a head detector is significantly more reliable than a full body detector. In order to address this problem, occlusion information is inferred from the scores of the part detections by selecting the part which maximises the detection score:

$$score(x) = \arg\max_{1 \leq i \leq 3} P(y = 1, H_i(x)) \tag{4.2}$$

where $P(y = 1, H_i(x))$ is the posterior probability of the i-part being a true positive. In this work the posterior is defined as a sigmoid function of the score $H_i(x)$:

$$P(y = 1, H_i(x)) = \frac{1}{1 + exp(A_i H_i(x) + B_i)} \tag{4.3}$$

The sigmoid model is equivalent to assuming that the detection score is proportional to the log odds of a positive example. The parameters A and B are learned for each part separately from the training set (see Section 4.2.1) by the sigmoid fitting approach proposed in [Platt, 1999]. An overview of the proposed combined head detector is shown in Figure 4.3

4.3 Experimental Setup and Evaluation

In order to evaluate the ability of the detector to distinguish between heads and all other objects, experiments were done on the town centre dataset [Benfold and Reid, 2011] which was chosen because it was designed for evaluation head detection and head tracking. This dataset is a high definition video (1920x1080 pixels/25fps) of a shopping street that has a ground truth consisting of 71500 hand labelled head and body locations. Following the methodology described in Chapter 2.4.1 the performance is summarised using thee MR and FPPI. For body regions the PASCAL measure [Dollár et al., 2012] is employed, which states that their area of overlap must exceed $\tau = 0.5$ (see Chapter 2.4.1). Since head regions are considerably smaller than full body regions, any error in the location has a greater impact on the performance measures. This is why the measure of Benfold and Reid [Benfold and Reid, 2011] is employed for heads, who define that the area of overlap must exceed $\tau = 0.25$.

Tests were performed with all three detectors separately and with the proposed combined detector; in addition, two detectors provided by Dollár et al. [Dollár et al., 2014] were tested. The first was trained on the INRIA [Dalal and Triggs, 2005] and the second was trained on the Caltech [Dollár et al., 2012] dataset.

Results are reported for head and body regions in Table 4.1 and some examples of detection results are shown in Figure 4.5. In addition, in Figure 4.4 the detectors are compared by plotting the MR against FFPI (using log-log plots) by varying the threshold of detection confidence (details can be found in [Dollár et al., 2012]). This is preferred to precision recall curves for certain tasks, e.g. tracking applications, as typically there is an upper limit on the acceptable FPPI rate independent of pedestrian density [Dollár et al., 2012]. The body region for the combined, the head and the upper-body detector as well as the head region for the upper-body, the body, the AcfInria and the AcfClatech detector are estimated using the same ratios used in Section 4.2.1 to create the training set.

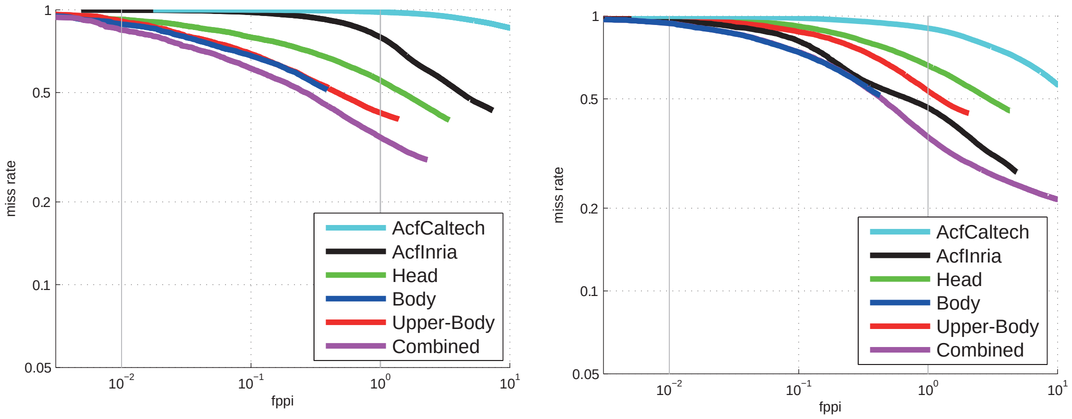


FIGURE 4.4: Log-log plots miss rate against false positives per image. Left: Head region. Right: Body region.

The results show (see Table 4.1) that the combined head detector outperforms all other detectors and reduced the MR for head detections from 76% (Head-Detector) to 58% (Combined-Detector). Even in the case of the full body region, the combined detector achieves the best result, which is particularly remarkable since using the head is often not discriminative in various tasks. In case of the body location, the proposed body detector and the AcfInria detector achieve similar results, but in case of the head location the proposed body detector archives a 28 % better result. This shows the advantage of defining the head as point of reference for the trainings image instead of the full body.

4.3.1 Performance

The combined detector needs 360 ms to process a 1920 × 1080 pixels image on the test machine, a desktop computer with an Intel Core i5-3470 CPU with 3.2 GHz and 8GB RAM. A single detector needs 260 ms. That the combined detector is only 38% slower is due to the fact that the most time-consuming process, the features computation, only has to be done once.

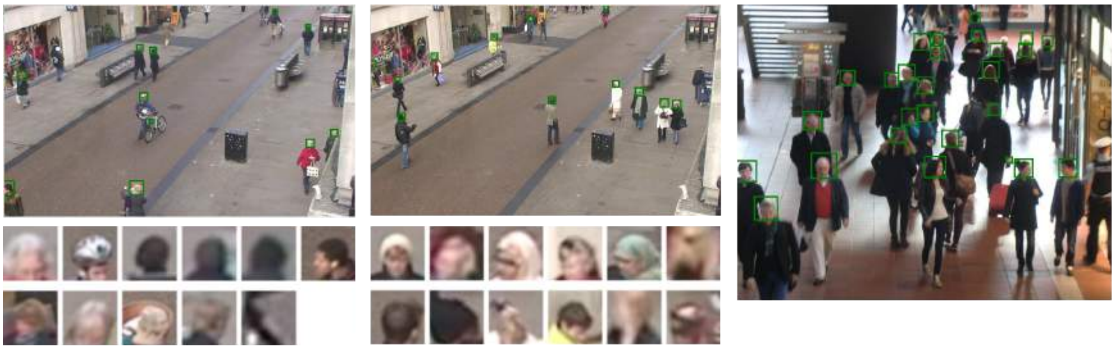


Figure 4.5: Some examples of detections on test images (1 + 2 town centre, 3 test image from the training set) for the final person detector.

Table 4.1: Performance on the town centre dataset

Method	MR - Head	MR - Body
AcfClatech [Dollár et al., 2014]	99%	96%
AcfInria [Dollár et al., 2014]	95%	72%
Head	76%	87%
Body	67%	70%
Upper-Body	66%	81%
Combined (proposed)	58%	66%

4.4 Conclusion

In this paper, a novel method was developed to combine different detectors for head detection. Three separate detectors for head, upper-body and body detection were trained based on the ACF detector framework of Dollár et al. [Dollár et al., 2014]. An algorithm was proposed to combine part detections that first estimates the head location of each part detector and then groups detections by partitioning them into disjoint subsets. The final confidence score is then calculated by maximising the detection score over all parts. In order to validate the findings, the performance of the detection system was examined on the town centre dataset. The results showed that combing a head, an upper-body and a body detector gives very good results for head detection by reducing the MR by 18%.

In the following chapter the proposed head detector is used as detector for multi-target tracking by detection system.

Chapter 5

Multi-Target Tracking

5.1 Introduction

In this chapter, a new system for real-time multi-target tracking is presented and analysed. Markov Chain Monte Carlo Data Association (MCMCDA) [Pasula et al., 1999, Yu et al., 2007, Liu et al., 2007b, Ge and Collins, 2008, Benfold and Reid, 2011] is adapted to estimate a varying number of trajectories given a set of detections extracted from a video sequence. The system focuses on head detection, because heads are hardly obscured from overhead surveillance cameras. The purpose of the system is to provide stable head location estimation in real-time for surveillance cameras. Therefore, the approach of Benfold and Reid [Benfold and Reid, 2011] was adapted, however several improvements were made with the aim to reduce the number of false positive heads and to increase the number of corrected tracked heads.

The first contribution involves the head detection. In order to be more robust to false positive detections and to minimise the number of missed detections, a novel schema is proposed which include the identification of true positives with the data association instead of using the internal decision making process of the detector. The assumption is that the average detection score of a true positive track is higher than the average detection score of a false positive track.

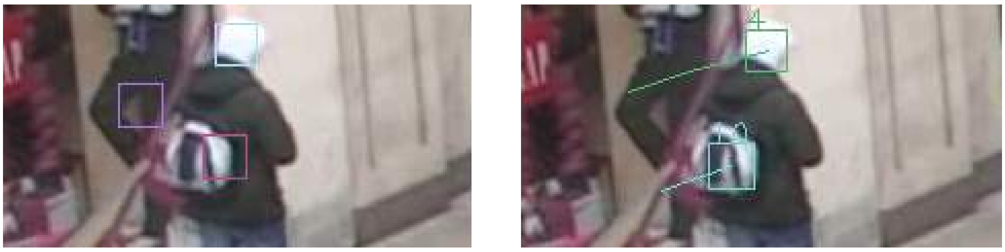


FIGURE 5.1: Results from the work of Benfold and Reid [Benfold and Reid, 2011].

The next contribution, involves the treatment of false positive detections. False positives are a frequent problem in multi-target tracking, and they either occur on background objects or as part of a foreground object. False positives in background regions are stationary and often repeatedly occur on the same position. Benfold and Reid [Benfold and Reid, 2011] have shown that such false positives can be filtered out by creating a separate model for false positives and then combine the identification of false positives with the data association. However, different to background false positives, foreground false positives are the result of incorrect detections on a foreground objects, for example incorrect head detections on other body parts such as shoulders or on bags.

It has been noticed that such incorrect detections have the same motion model as true positives and therefore can not be detected by the model of Benfold and Reid [Benfold and Reid, 2011]. Figure 5.1 shows an example where the model of Benfold and Reid [Benfold and Reid, 2011] failed to detected foreground false positive. The left image shows the raw detections with a true positive (blue rectangle), a background (purple rectangle) and a foreground false positive (red rectangle). The right image shows the results after data association where the algorithm of Benfold and Reid [Benfold and Reid, 2011] was able to remove the background false positive but not the foreground false positive (blue rectangle). However experiments showed, that even when the motion model of foreground false positive is the same as of true positives, they have mostly properties which label them as false positives. In order to filter out foreground false positives the approach of Benfold and Reid [Benfold and Reid, 2011] was expanded with separate models for background and foreground false positives rather than background false positives only.

The framework was evaluated on the town centre benchmark where a MOTA of 81.55% was achieved and on the Parking Lot benchmark where a MOTA of 79.71% was achieved. Additional experiments assess various sub-parts of the system that might affect the performance.

The rest of this chapter is organised as follows. A description of the system employed can be found in Section 5.2. Experimental results are presented in Section 5.3. Conclusions are presented in Section 5.4.

5.2 Multi-Target Tracking

In agreement with the majority of recent multi-target tracking methods [Liu et al., 2007b, Yu et al., 2007, Ge and Collins, 2008, Benfold and Reid, 2011], tracking by detection is pursued. Targets (pedestrians) are separated from the background in a preprocessing step and form a set of target hypotheses, which are then used to infer the target trajectories. In order to estimate the location of pedestrians in the current frame and to ensure that data associations can be made correctly, a tracker is initialised to track the relative motion for a period of time d (in our case d = 75 frames). In order to achieve real-time performance, a multi-threaded approach is used in which one thread produces asynchronous detections while a second thread applies the tracking algorithm and a third thread performs data association. Figure 5.2 illustrates the relation between each thread.

5.2.1 Data Association

Assuming there is a set of detections $D = \{D_1, D_2, ..., D_\tau\}$ in the time interval $[1, \tau]$, where $D_t = \{d_{t1}, d_{t2}, ..., d_{tn}\}$ are the detections obtained at the frame t. For each detection, a tracker is initialised to track the relative motion for a period d (in our case d = 75 frames). The aim is to find the hypothesis, H_i, that divides the detections into a set of target trajectories $T = \{T_1, T_2, ..., T_j\}$, so that each track

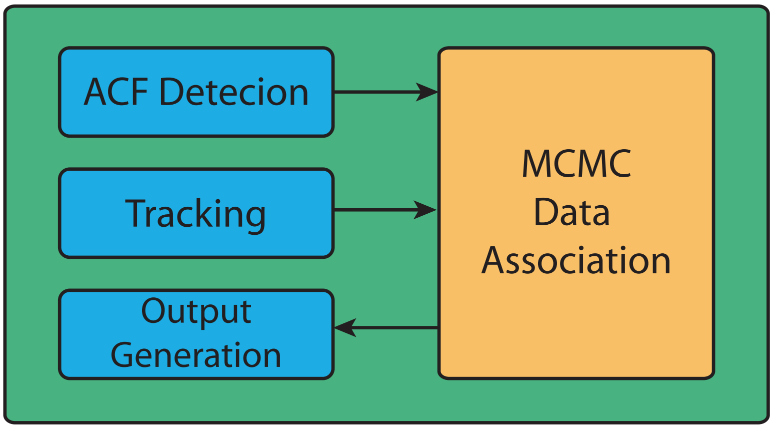


Figure 5.2: System overview.

contains all observations of a single person. In order to represent false positive trajectories, each track has a type c_j that can take the values c_p for true positive, c_{f1} for foreground false positive and c_{f2} for background false positive trajectories. Each observation is constrained to be associated with at most one track, and only one detection can be associated to a track at each time step.

The tracking problem is then formulated as a Bayesian problem and then the Maximum a Posterior (MAP) estimator of the posterior distribution is taken as the optimal solution for the hypothesis H:

$$H^* = \arg\max(p(H|D)) = \arg\max(p(D|H)p(H)) \tag{5.1}$$

where $p(D|H)$ is the likelihood function that models how well the hypothesis fits the detections and $p(H)$ expresses the prior knowledge about desirable properties of good trajectories. As prior the function of [Benfold and Reid, 2011] was adapted:

$$p(H_i) = J! \prod_{T_j \in H_i} \left(\frac{|T_j|}{|D|} \right)^{|T_j|} p(c_j) \tag{5.2}$$

where $p(c_j)$ is a prior over the different track types in case of this thesis c_p, c_{f1} and c_{f2}. $|D|$ and $|T_j|$ are the cardinalities of the sets D and T_j. The factor of $J!$ arises because the ordering of the subsets is not important, so the first detection in any track may be encoded with any of up to J identifiers which have not already been used [Benfold and Reid, 2011].

In order to represent both background and foreground false positives, the following likelihood function is proposed:

$$p(D|H_i) = \prod_{T_j \in H_i} \left[p(d_1^j|c_j) \prod_{d_n^j \in T_j / d_1^j} p(d_n^j|d_{n-1}^j, c_j) \right] \tag{5.3}$$

where d_n^j is the n detection in a track T_j, with n indicating only the order within the track. The link probability between two detections is then defined as the product of four probabilities, namely the probability for size s, location x, motion m, and detection score r.

$$p(d_1^j|c_j) = p(s_1|c_j)p(x_1|c_j)p(m_1|c_j)p(r_1|c_j) \tag{5.4}$$

$$p(d_n^j|d_{n-1}^j, c_j) = p(s_n|s_{n-1})p(x_n|x_{n-1}, c_j)p(m_n|c_j)p(r_n|c_j) \tag{5.5}$$

The proposed link probability is designed to represent the correct data association and track types and allows in contrast to the link probability of [Benfold and Reid, 2011] to make a distinction between true positives and foreground false positives. In the following, each probability will be explained in detail.

5.2.2 Feature Extraction and Modelling

The features are designed to represent the correct data associations and track types. For each detection, the size of the detection s_n, the location x_n, the motion vector m_n and the detection score r_n with $n \in [1, |T_j|]$ is encoded.

5.2.2.1 Detection Score

Object detection algorithms like the Histograms of Oriented Gradients (HOG) based detection algorithm used in [Benfold and Reid, 2011], utilise only the information present in a single frame to decide if a possible head candidate is a true positive. However, in the case of multi-target tracking, each true detection is part of a track of true detections. This additional information can be used to increase the accuracy at each frame.

In order to exploit this additional knowledge, a novel scheme is proposed that instead of using the internal decision making process of the detector, it includes the identification of true positives with the data association. The assumption is that the average detection score of a true positive track is higher than the average detection score of a false positive track. This has two advantages: firstly the recognition rate of false positives improves and secondly true positive detections with low confidence are included which would otherwise be removed.

The general idea is that a head detector is trained in such a way that all possible head candidates are returned, the ones with low confidence which are most likely false positives. Then a probability is assigned to each detection that describes how certain it is that this detection is a true positive and then this probability is included in the data association process. The combined detector proposed in Chapter 4 is used, but each part detector is retrained such that 99% of the true detection are retained without taking false positives into account. This results in a detector that returns almost all true positives and an acceptable number of false positives (see Figure 5.3). A example can be seen in Figure 5.3. On the left the result of detector trained with default configuration are shown. It can be seen that two heads were no detected (marked with a circle). On the right the result of a detector trained with proposed configuration, all head were detected but three false positive detections (marked with a circle).

The probability of each detection to be a true positive is then modelled in the same way as in Chapter 4:

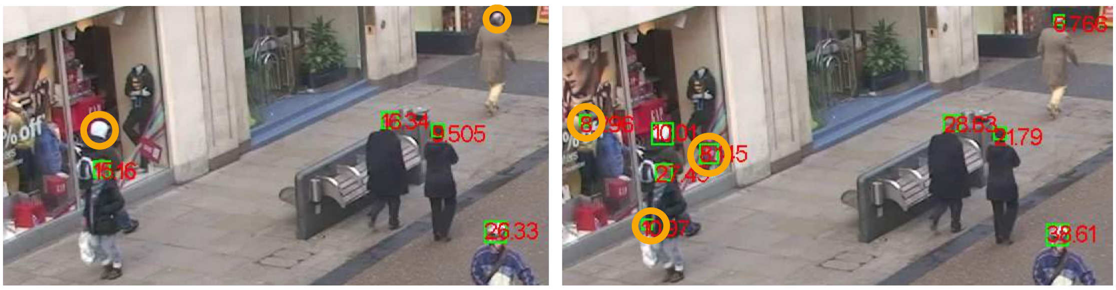


FIGURE 5.3: Detector results with default and proposed configuration.

$$f(r_n) = \arg\max_{r_n} \frac{1}{1 + exp(A_i r_{ni} + B_i)} \tag{5.6}$$

where $r_n = \{H_1(x), H_2(x), H_3(x)\}$ is the confidence vector that containers the confidence of each part detector. The probability that a detection is a true positive detection is then:

$$p(r_n|c_p) = f(r_n) \tag{5.7}$$

and a false positive detection:

$$p(r_n|c_{f1}) = 1 - f(r_n) \tag{5.8}$$

$$p(r_n|c_{f2}) = 1 - f(r_n) \tag{5.9}$$

5.2.2.2 Detection Size

Benfold and Reid [Benfold and Reid, 2011] assumed that the size of the first detection has a global prior log-normal distribution that is independent of the image location. However, in most scenarios that were examined the size of a true positive detection strongly correlates with the image location x. For example, in the town centre scene the average head size of a person on the left side is 50.3 pixels whereas the average size on the right side is 18.7 pixels. Figure 5.4 illustrates the relation of image location (x and y axis) to average head size (z axis) in the town centre scene. This relation is modelled with a probability map that depends on the image location x_n, y_n and assume a normal distribution:

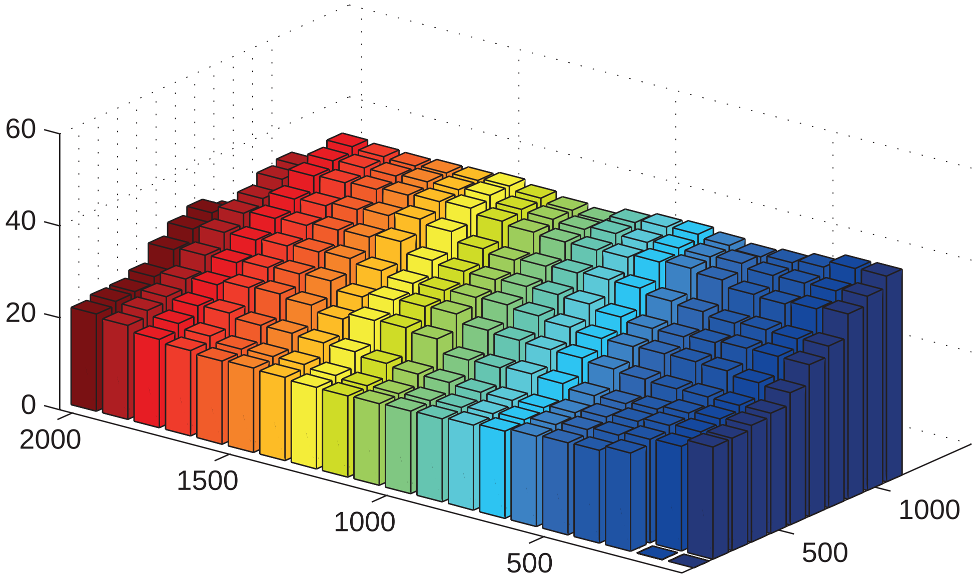


Figure 5.4: Mean detections size in the town centre scene summed over 100 × 100 blocks.

$$ln(s_1) \sim N(\mu_{map}(x_1, y_1), \sigma^2_{map}(x_1, y_1)) \tag{5.10}$$

In contrast, the size of the foreground and background false positives is uniformly distributed over the set of possible sizes S:

$$p(s_1) = \frac{1}{|S|} \tag{5.11}$$

For the following detections in the track, the size is then encoded by the ratio to the previous detection:

$$ln\frac{s_n}{s_{n-1}}\bigg|c_p \sim N(0, \delta_t \sigma^2_{sp}) \tag{5.12}$$

$$ln\frac{s_n}{s_{n-1}}\bigg|c_{f1} \sim N(0, \delta_t \sigma^2_{sf1}) \tag{5.13}$$

$$ln\frac{s_n}{s_{n-1}}\bigg|c_{f2} \sim N(0, \delta_t \sigma^2_{sf2}) \tag{5.14}$$

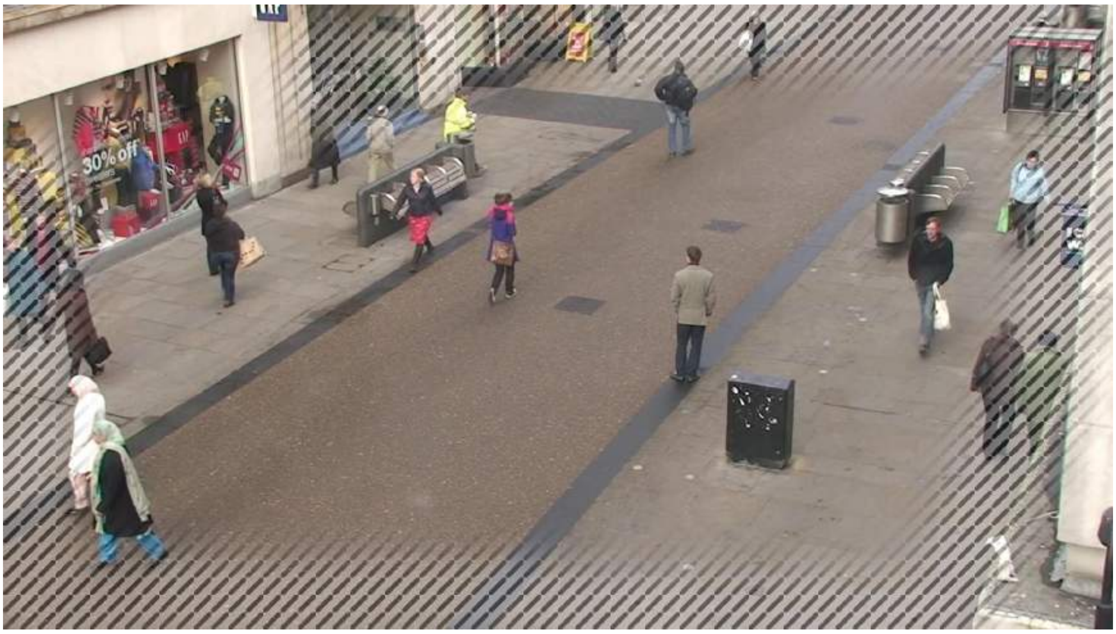


Figure 5.5: Entry map in the town centre video.

where δ_t is the time difference between the frames in which the detections were made.

5.2.2.3 Detection Location

Previous approaches [Ge and Collins, 2008, Benfold and Reid, 2011] have assumed that the locations of both pedestrians and false positives are uniformly distributed around the image; however in the case of a stationary camera this is not true for pedestrians. A pedestrian always has to enter the scene at some point and therefore the first detection has to be next to an entry point. In order to model this fact, an entry map is built. Figure 5.5 shows an example of such an entry map. The hatched area defines the possible entry points. The probability that a track is a true positive track depends on the distance from the first detection x_n to the next entry point ep divided by the detection size s_1:

$$\left.\frac{||x_1 - ep||}{s_1}\right|c_p \sim N(0, \sigma_p^2) \tag{5.15}$$

where $||\cdot||$ is the Euclidean distance.

For false positives, it is assumed that the location of the first detection is uniformly distributed around the image, therefore the probability density of x_1 is proportional to the square of the object size s (in pixels), divided by the image area α:

$$p(x_1) = \frac{s_1^2}{\alpha} \tag{5.16}$$

For the following detections, the probability depends on the track type. For true positives and foreground false positives, the probability depends on the estimated location x_{est} of the previous detection x_{n-1} at the time t, where t is the time at which the following detection x_n was made. In order to estimate the location x_{est}, the tracker that was proposed recently in [Ben-Ari and Ben-Shahar, 2013] is used. This tracker combines template matching and an adaptive Kalman filter and is able to deal with temporary occlusions. A normal distribution is assumed:

$$||x_n - x_{est}||\Big|c_p \sim N(0, \sigma_{lp}^2 + 2\sigma_d^2) \tag{5.17}$$

$$||x_n - x_{est}||\Big|c_{f1} \sim N(0, \sigma_{lf1}^2 + 2\sigma_d^2) \tag{5.18}$$

where $2\sigma_d$ is an additional uncertainty that models the error in the two detection locations. Background false positives are the result of background objects, and therefore the location is assumed stationary:

$$||x_n - x_{n-1}||\Big|c_{f2} \sim N(0, 2\sigma_d^2) \tag{5.19}$$

5.2.2.4 Motion Vector

As the last feature, a motion vector histogram similar to [Benfold and Reid, 2011] is used. This histogram included to distinguish between background false positives which are expected to have no movement and true positives which are excepted to have at least a small amount of movement. The motion vector histogram has four bins with boundaries representing movement of $\frac{1}{8}, \frac{1}{4}, \frac{1}{2}$ pixels per frame, where the motion vector is calculated from the result of the tracking in the first five frames immediately after the detection. A multinomial distribution is then used to model the probability:

$$m_n|c_p \sim Mult(m_p) \tag{5.20}$$

$$m_n|c_{f1} \sim Mult(m_{f1}) \tag{5.21}$$

$$m_n|c_{f2} \sim Mult(m_{f2}) \tag{5.22}$$

The multinomial distribution is a generalization of the binomial distribution. The parameters m_p, m_{f1} and m_{f2} define for each track type the event probability of each histogram bin.

5.2.3 Markov Chain Monte Carlo Data Association

Evaluating the space of hypotheses is extremely challenging and a closed form solution is usually not available in practise. That is why Markov Chain Monte Carlo Data Association (MCMCDA) [Ge and Collins, 2008, Benfold and Reid, 2011] is used to estimate the best hypotheses H^*.

The MCMCDA algorithm is based on the Metropolis Hastings algorithm. Metropolis Hastings is a method for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult. The Metropolis Hastings algorithm generates a sequence of sample values in such a way that as more and more sample values are produced, the distribution of values more closely approximates the desired distribution $P(H)$. At each iteration, the algorithm picks a candidate for the next sample value based on the current sample value and a proposed distribution $q(H_i \rightarrow H^*)$. Then an acceptance function defines the likelihood with which the proposal should be accepted:

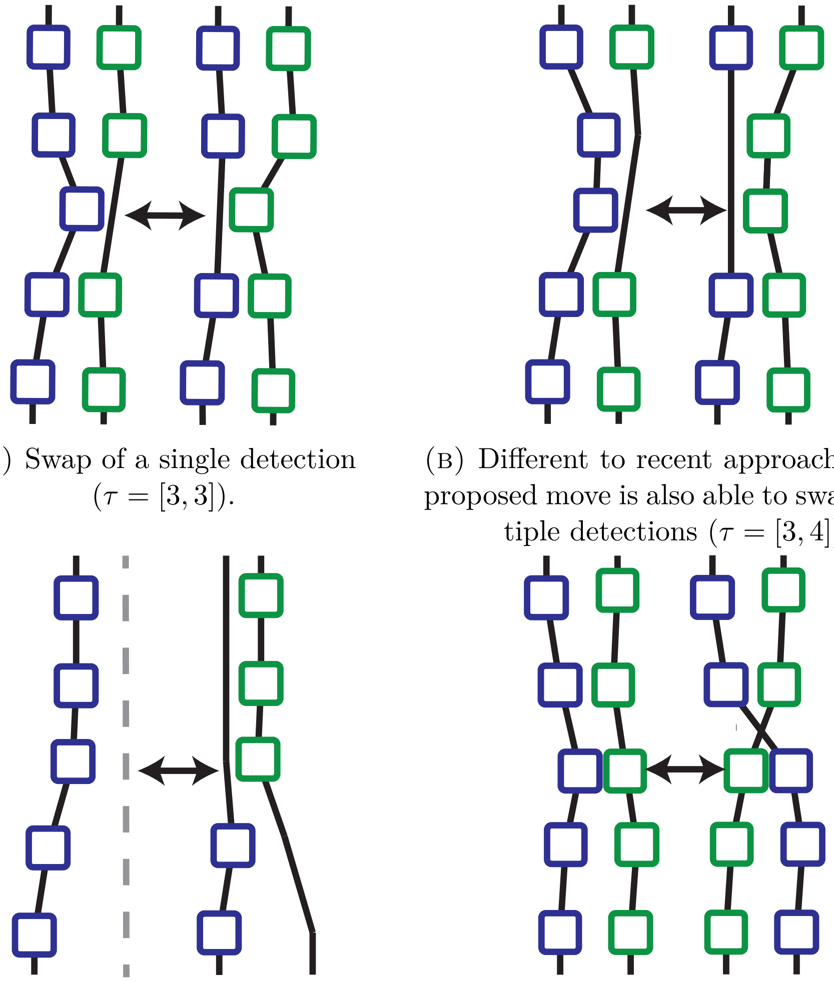
$$p(H_{i+1} \rightarrow H^*) = \min\left(1, \frac{p(H^*)}{p(H_i)} \frac{q(H_i \rightarrow H^*)}{q(H^* \rightarrow H_i)}\right) \tag{5.23}$$

In case of MCMCDA, Metropolis Hastings sampling is used to explore the space of data associations by generating a sequence $H_0, H_1, H_2, ..., H_n$ of sample hypotheses. The aim is to find the hypotheses H^* with the highest probability, hence the algorithm keeps track of the most likely hypothesis H_{max} since the last received hypothesis is not guaranteed to be the most likely. The complete algorithm is summarized in Algorithm 2.

In this work, the MCMCDA algorithm is used within a temporal sliding window [Stalder et al., 2010, Song et al., 2010, Benfold and Reid, 2011] representing the most recent d frames that have been received. Doing so allows the algorithm to recover from false associations and makes it robust against inaccurate detections or tracking errors.

Different to recent approaches [Ge and Collins, 2008, Benfold and Reid, 2011], this thesis proposes to use a proposal distribution that consists only of two moves. The first move can change the data association and the second move can change the type of a track. In the first move, a pair of trajectories is randomly selected (T_i, T_j), as well as a switch range $\tau = [t_1, t_2]$ within the sliding window W. Then all the detections in this range τ are switched between the two the trajectories. This move substitutes all data association moves of recent approaches. Figure 5.6 compares the proposed move with the moves of [Ge and Collins, 2008] and [Benfold and Reid, 2011]. If a move creates a new empty track, then this empty track is removed and when one or multiple detections are added to the empty track, a new empty track is added to the set of trajectories.

In second move, a random track T_j is selected and then the type c_j of this track is randomly switched to one of the remaining types.



(A) Swap of a single detection ($\tau = [3, 3]$).

(B) Different to recent approaches, the proposed move is also able to swap multiple detections ($\tau = [3, 4]$)

(C) The empty track allows to split and merge tracks ($\tau = [3, 5]$).

(D) By setting t_2 to the end of the window the move creates the same result as the switch move in [Ge and Collins, 2008, Benfold and Reid, 2011] ($\tau = [4, 5]$)

FIGURE 5.6: Examples of the first move used for MCMCDA, the rectangles in the same colour belong to the same track

5.2.4 Assignment of Detections

Recent approaches [Benfold and Reid, 2011] add a set of new detections $D_t = \{d_{t1}, d_{t2}, ..., d_{tj}\}$ to data assignment by creating a new track for each detection that only contains this single detection. However, experiments showed that this is not the optimal solution since most of the new detections belong to existing tracks. Finding the optimal solution for such a problem is known as assignment problem, which can be formulated as follows.

```
Input          : $D, T, H_0, n_{mc}, W$
Output         : $H_{max}$
Initialization: $H \leftarrow H_0, H_{max} \leftarrow H_0$
for $i \leftarrow 1$ to $n_{mc}$ do
    sample a move $m$ from the distribution $p_H(m, W)$;
    propose $H'$ from the move specific proposal $p_m(H)$;
    sample $U \sim Uniform(0, 1)$;
    if $U \leq p(H_{i+1} \rightarrow H^*)$ then
        $H \leftarrow H'$;
        if  $p(H) > p(H_{max})$ then
            $H_{max} \leftarrow H$;
        end
    end
end
```

Algorithm 2: Markov Chain Monte Carlo Data Association

Given an $m \times n$ cost matrix C let m be the number of tracks, n is the number of detections and $C_{i,j}$ the cost of assigning j-th detection to the i-th track. The aim is to select n elements of C, so that exactly one track is assigned to one detection and the sum of the corresponding costs is minimum.

$$C_{m,n} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{12} & \cdots & C_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{32} & \cdots & C_{1n} \end{pmatrix} \tag{5.24}$$

where m is the size of all recent tracks plus the number of new detections $m = |T| + |D_t|$ and n is the size of all new detections $n = |D_t|$. The components of the cost matrix are then defined as:

$$C_{i,j} = \begin{cases} 1 - p(T_i | d_{tj}, c_i) & \text{if } i \leq |T| \\ 1 - p(d_{tj} | c_{max_{tj}}) & \text{if } i > |T| \text{ and } i = j \\ \infty & \text{otherwise} \end{cases} \tag{5.25}$$

where $p(T_i|d_{tj}, c_i)$ is the link probability between the last detection in a track T_i and a new detection d_{tj} (see Equation 5.5). $p(d_{tj}|c_{max_{tj}})$ is the maximum of the first detection likelihood (see Equation 5.4) over the set of track types $c_j = \{c_p, c_{fp1}, c_{fp2}\}$:

$$c_{max_{tj}} = \arg\max(p(d_{tj}|c_{tj})) \tag{5.26}$$

In this work, the Hungarian method [Kuhn, 1955, Munkres, 1957, Miller et al., 1997] also known as Kuhn-Munkres algorithm or Munkres assignment algorithm, is used to find the optimal assignment for the given cost matrix.

5.2.5 Parameter Estimation

The optimal model parameters are likely to depend on the tracking scenarios. Ge and Collins [Ge and Collins, 2008] have proposed an approach for automatic parameter estimation by interleaving the MCMCDA sampling with an additional Metropolis Hastings update for the parameters. In this thesis, the same approach is used to estimate the parameters, however as pointed out in [Benfold and Reid, 2011] parameter updates take considerably longer then data association updates because the likelihood must be recalculated for all the data. In the case of a live system, the parameters can be learned over several hours but since most of benchmarks are only some minutes long, the videos are slowed down for training so that there is enough time to learn the parameters. The entry points for the entry map were manually defined.

5.3 Experimental Setup and Evaluation

In this section the performance of the proposed multi-target tracker is evaluated on two challenging image sequences. The model parameters were learned as described in Section 5.2.5. Since detections do occur delayed and not in every frame, the current location is estimated with the tracker of the last detection in each track.

TABLE 5.1: Town centre benchmark head results.

Exp.	Method	MOTA	MOTP	Prec.	Rec.	False Pos.	Missed
1	Benfold [Benfold and Reid, 2011]	45.40%	50.80%	73.80%	71.00%	18374	20427
	Proposed	81.55%	64.51%	90.87%	91.35%	6500	6127
2	Test 2A	74.61%	64.53%	91.29%	82.71%	5586	12247
	Test 2B	76.53%	64.35%	85.63%	92.73%	11021	5147

5.3.1 Town Centre Benchmark

The town centre benchmark is a high definition video (1920x1080 pixels/25 fps) of a shopping street that has a ground truth consisting of 71500 hand labelled head locations. In Table 5.1 the results of the head location estimation are compared to the work of Benfold and Reid [Benfold and Reid, 2011]. Some example tracking results are presented in 5.7. The results show the advantages of the proposed approach. It can be seen that the number of missed detections and simultaneously the number of false positive detections are reduced. The improvement in the proposed method is a result of two main factors: Firstly, the combined detector produce more accurate head detections. Secondly, the schema that distinguishes between foreground and background false positives is more robust against false positive.

The next experiment (2) shows the impact of the different contributions for false positive detection. Therefore, two tests are done; Test $2A$ illustrates the advantage of including the identification of true positives with the data association by showing the recognition rate of the combined detector with default configuration (see Section 5.2.2.1) and without including the score probability. The results show that the number of missed detections increases from 6127 to 12247 and that the number of false positives drops slightly from 6500 to 5586. This is due to the fact that the proposed detector is explicitly trained such that all possible head candidates get returned, even those with low confidence scores (see Section 5.2.2.1). As a consequence, the number of missed detections drops but also some additional false positives appear that could not be filtered in the data association step.

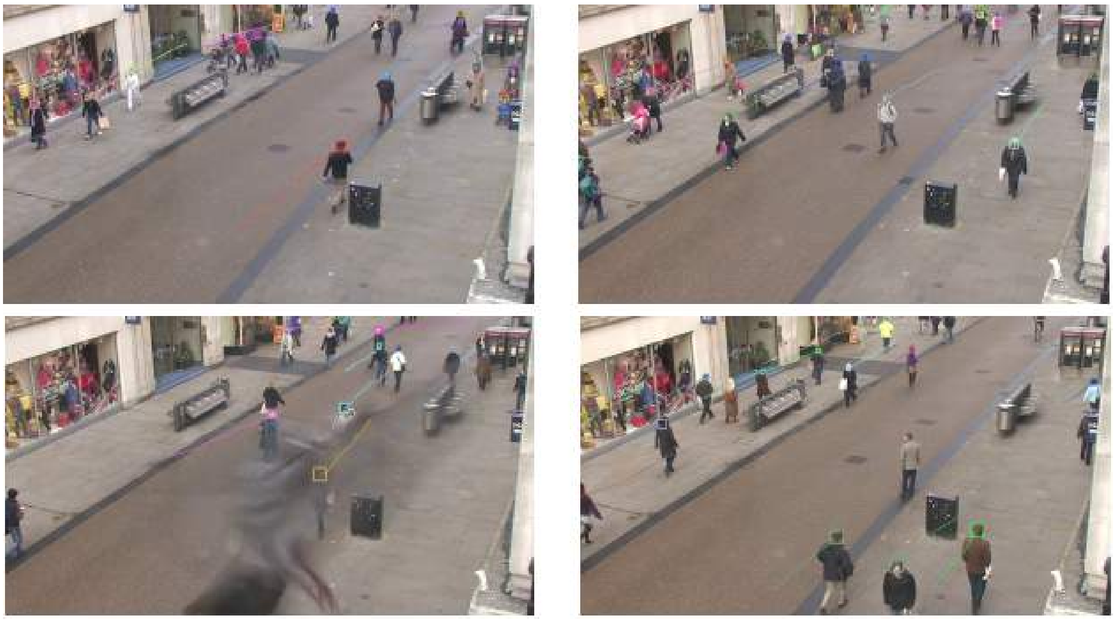


FIGURE 5.7: Sample video frames from the town centre sequence.

Test $2B$ shows the advantage of creating a separate model for background false positives by testing the proposed schema without background false positives. The resulting schema is then similar to the schema of [Benfold and Reid, 2011]. In this test, the number of false positives increases from 6500 to 11021, possibly because foreground false positives get classified as true positives since they have the same motion model. Simultaneously the number of missed detections drops 6127 to 5147, probably because true positive tracks can not be erroneously classified as foreground false positives.

5.3.2 Parking Lot Benchmark

The parking lot benchmark is a high definition video (1920x1080 pixels/29 fps) of a parking lot in which a group of 14 pedestrians are walking through the scene. It has a ground truth consisting of 2500 hand labelled body locations. The challenges in this benchmark include long-term inter-objects occlusions, camera jittering and similarity of appearance among the humans in the scene [Shu et al., 2012]. Since the ground truth is for body locations and the proposed algorithm only tracks heads, the full body regions were estimated using the fixed ratio as in 4.2.1. The

Table 5.2: Parking Lot benchmark results.

Method	MOTA	MOTP	Prec.	Rec.	False Pos.	Missed
Shu [Shu et al., 2012]	79.3%	74.1%	91.3%	81.7%	n/a	n/a
Proposed	79.71%	73.53%	95.49	91.4%	189	212

results are shown in Table 5.2. The proposed algorithm achieves similar results to [Shu et al., 2012], however their approach is not real-time.

5.3.3 Performance

The bottleneck of most tracking-by-detection systems is the detections step, especially in HD videos. In the proposed approach this issue is addressed by a multi-threaded approach in which one thread produces asynchronous detections while a second thread applies the tracking algorithm and a third thread performs MCMCDA. MCMCDA has the additionel advantage that at any instant in time it can report its current best estimate of all target tracks. This architecture ensures that the proposed approach runs always in real-time. However, this also means that the accuracy of system is directly dependent on the performance of each part. For example if the detection step would need 10 minutes, a correct data association would be almost impossible. That is why the performance of all parts will be evaluated individually in the following. The test system is a desktop computer with an Intel Core i5-3470 CPU with 3.2 GHz and 8GB RAM.

Benfold and Reid [Benfold and Reid, 2011] used a GPU implementation of a HOG detector that needed on their system 1200 ms, no details are given about what kind of GPU was used. On the test system, which is used in this thesis, the OpenCV CPU implementation (`HOGDescriptor::detectMultiScale`) of a full body HOG detector needs on average 2500 ms when using only one core. Shu et al. [Shu et al., 2012] use in their approach deformable part-based model for human detection similar to the approach of Felzenszwalb et al. [Felzenszwalb et al., 2010b]. Shu et al. [Shu et al., 2012] do not provide source code but the approach of Felzenszwalb et al. [Felzenszwalb et al., 2010b] is public available, which needs on the test system used in the thesis 6000 ms. As mention in Chapter

4 the proposed detector needs only 360 ms, which is significantly faster then the approaches used by Shu et al. [Shu et al., 2012] and Benfold and Reid [Benfold and Reid, 2011].

Benfold and Reid use as tracking algorithm Kanade-Lucas-Tomasi (KLT) with 4 points, the OpenCV implementation of this algorithm (calcOpticalFlowPyrLK) needs on the test system per frame and detection 0.1 ms. The proposed system uses tracking algorithm of Ben-Ari and Ben-Shahar [Ben-Ari and Ben-Shahar, 2013] which needs on the test system 0.5 ms per frame and detection. Shu et al. [Shu et al., 2012] do not use a tracking algorithm.

For data association the proposed system as well as the system of Benfold and Reid use MCMCDA. Experience showed that generating 5000 sample values is sufficient reach correct data association for which the system needs average 35 ms. Shu et al. [Shu et al., 2012] algorithm needs on a conventional desktop computer between 200 ms and 1000 ms.

5.4 Conclusion

This chapter presents an approach for real-time multi-target tracking that effectively deals with false positive detections. In order to achieve this, a novel motion model was built that treats false positives on background objects and false positives on foreground objects separately. The novel model makes the tracker robust against false positives and simultaneously reduces the number of missed detections. In addition, a novel detector was proposed that combines head, upper body and body detection. The results showed that this approach is superior to earlier approaches. For the town centre dataset the number of missed detections decreases from 18374 to 6500 and the number of missed detections decreases from 20427 to 6127.

In the next chapter the proposed multi-target tracking system is used to build face tracks which are then used for video face recognition and in Chapter 7 it is extended for multi-target tracking for fisheye cameras.

Chapter 6

Face Recognition in Videos via Generalized Similarity

6.1 Introduction

Face recognition is an important and popular computer vision topic. Applications of face recognition can be found in surveillance and security, human-computer intelligent interaction and smart environments. Recently, video-based face recognition has become more and more popular, where the problem becomes more challenging due to illumination changes, pose variation, and occlusion. However, it also has the benefit of providing a setting in which weak evidence in a single frame can be integrated over a set of frames to achieve a more accurate result.

The video-based face recognition problem depends on two tasks: accurate face tracking and recognition of the tracked data. Face tracking is a critical step that first detects the face in video frames and then associates the detections to a track, from which feature descriptors can be extracted and fed as input data to the face recognizer. This chapter introduces a novel fully automatic framework for video face recognition, which includes face detection, multi-target face tracking, face alignment, and video face recognition.

The contributions of this chapter are summarized as follows: The first contribution is a novel face detector that combines a general classifier with five view-specific classifiers. The detector uses boosted classifiers and pixel lookups in aggregated channels [Dollár et al., 2014] as features. The detector reaches state of art results and yields considerable speed-up compare to detector that uses only view-specific classifiers.

Furthermore two novel set-to-set similarity measures, the Generalized Matched Background Similarity (GMBGS) and the Mean Vector Generalized Similarity (MVGS), are presented. These similarity measures are designed for comparing the frames of two face videos in order to determine whether the faces appearing in the two sets are the same subject. The GMBGS method is built upon the Matched Background Similarity (MBGS) [Wolf et al., 2011] and utilizes the Generalized Similarity (GS) for sets. Whereas the MBGS uses a linear combination of feature vectors from a set in order to use the GS directly. In addition, a complete framework is presented that uses the face detector and the similarity measures in order to create a robust video-based face recognizer. Finally, the algorithms are evaluated on three existing datasets (YouTube Faces, YouTube Celebrities, and ChokePoint), where recognition rate of 83.44%, 84.07% and 100% were achieved.

6.2 Proposed Approach

In this section, the proposed approach is described in detail. Firstly, the proposed face detector is described followed by the face tracking algorithm based on multi-target tracking. Finally, details of the proposed set-to-set similarity measures are given.

6.2.1 Face Detection

Lots of work has been done in object and face detection in order to increase the detection accuracy, however, improved detection accuracy has been accompanied by increased computational costs. As pointed out in [Dollár et al., 2014], the Viola and Jones detector [Viola and Jones, 2002] ran at 15 frames per second (fps) over a decade ago, on the other hand, most recent detectors require multiple seconds to process a single image as they compute richer image representations. The aim of this thesis is to achieve real-time performance, therefore a novel face detector is proposed called ACFFace-5 that focuses on speed while maintaining accurate.

The ACFFace-5 was inspired by the SquaresChnFtrs-5 detector proposed in [Mathias et al., 2014]. There are two main differences between the proposed detector and the SquaresChnFtrs-5 detector. Firstly, the SquaresChnFtrs-5 trains classifiers for five different views that are later eventuated individually. The proposed detector first evaluates a general classifier that was trained on all views and then evaluates view-specific classifiers only on promising windows. In doing so the evaluation cost is reduced, since for all non promising results only one classifier has to be evaluated instead of five.

Secondly, the SquaresChnFtrs-5 detector depends on the Integral Channel Features (ICF) [Dollár et al., 2009, Dollár et al., 2010] framework while the proposed detector depends on the Aggregated Channel Features (ACF) [Dollár et al., 2014] framework, which achieves slightly better results (see [Dollár et al., 2014]). Both, ACF and ICF, use the same channel features and boosted classifiers; the key difference between the two frameworks is that ACF uses pixel lookups in aggregated channels as features while ICF uses sums over rectangular channel regions (computed efficiently with integral images) [Dollár et al., 2014]. Both frameworks have similar accuracy [Dollár et al., 2014] but ICF is slower than ACF (in [Dollár et al., 2014] 16 fps versus 30 fps) due to construction of integral images and more expensive features.

The detector was trained on the AFLW database [Koestinger et al., 2011] that consists of 26000 annotated faces. A frontal face detector (yaw angle ±20 degrees) and four side views (20 to 60, 60 to 100, -20 to -60, -60 to -100 degrees) were trained using 5886, 3700, 2131, 3424 and 1741 samples respectively. Pitch and roll were kept between 22.5 degrees. Each face was resized to 80×80 pixels with an addition padding of 8 pixels to each side. As the negative training set, background images from the INRIA dataset [Dalal and Triggs, 2005] were used.

The classifiers were trained jointly in multiple rounds of bootstrapping to make sure that no additional false positives could be found in the negative training set. In each round, the general classifier was trained first and then the component classifiers. For each detector the same configuration was used. Ten feature channels were used: normalised gradient magnitude, histogram of oriented gradients (6 channels) and LUV colour channels; the block size was set to 4×4. AdaBoost was used to train and combine depth-two trees over the candidate features (channel pixel lookups) in each window. For the general classifier $N = 512$ trees were used and for the view classifiers $K = 1536$ trees were used. This values were determined empirically. The resulting classifier H has the flowing form:

$$H(x) = \sum_{i=1}^{N} \alpha_i h_i(x) + \arg\max_{1 \leq v \leq V} \sum_{i=1}^{K} \alpha_i^{(v)} h_i^{(v)}(x) \tag{6.1}$$

where each h_i is a weak classifier (with output -1 or 1), α_i is the associated weight and V are the number of components. If $H(x) > 0$, then x is classified as positive and $H(x)$ serves as the confidence score. The final confidence score is the confidence score of the general classifier plus the maximum confidence score of the view-specific classifiers. During evaluation, a rejection threshold τ is used after evaluation of every weak classifier and if $H(x) < \tau$ computation stops. After evaluation, the detected bounding boxes are concatenated and if their overlap is greater than 0.3 then the bounding boxes with the lower scores are suppressed. Figure 6.1 summarizes the complete evaluation process.

6.2.2 Face Tracking

In order to achieve reliable face tracking, the real-time multi-target tracking system presented in Chapter 5 is extended for face tracking. Therefore, the combined head detector that was used in Chapter 5 and presented in Chapter 4 was extended

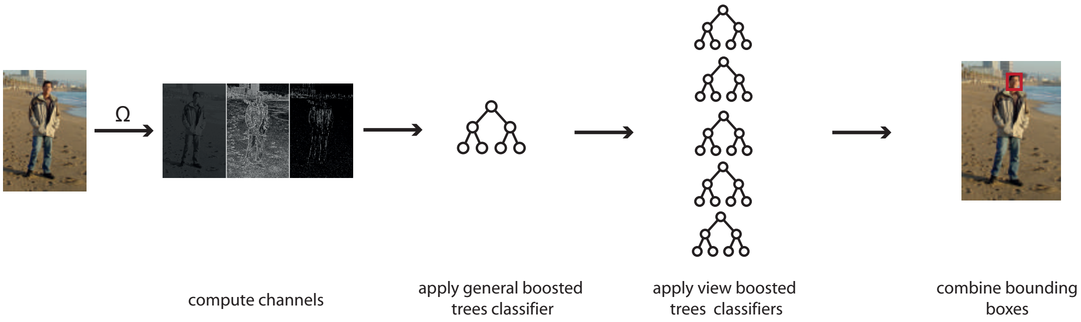

Ω
compute channels
apply general boosted trees classifier
apply view boosted trees classifiers
combine bounding boxes

FIGURE 6.1: Overview of the ACFFace-5 detector.

with the ACFFace-5 detector as an addition part detector. Compared to other approaches that rely only on face detections [Kim et al., 2008, Ortiz et al., 2013] a combination of head and face detection has the advantage that a person (head) can be tracked regardless whether the face is visible or not. For each face, fiducial points are localized using the approach proposed in [Asthana et al., 2014a] that is public available under [Asthana et al., 2014b]. The points are then used to align the faces with a similarity transformation, as described in Chapter 2.2.2.

6.2.3 Feature Extraction

Local Quantized Patterns (LQP) [ul Hussain and Triggs, 2012] are a generalization of local pattern features (such as LBP [Ojala, 1996]), that provide good results for face recognition [ul Hussain et al., 2012]. Local pattern features, based on the idea that small patterns of qualitative local gray-level differences, contain a great deal of information about higher-level image content. LQP uses a lookup-table-based vector quantization to code larger or deeper patterns. For example Hussain et al. [ul Hussain et al., 2012] used a LQP pattern that involved 24 ternary pixel comparisons whereas LBP only uses 8 binary pixel comparisons.

In the current study, LQP are used as features for face representation. Similar to [ul Hussain et al., 2012], a disk layout is used to sample pixels from the neighbourhood, to generate a pair of binary codes using ternary splits (see Equation 6.3) and to quantize each one using separately learned codebooks. The codebooks are learned by applying k-means clustering to all possible feature vectors (see [ul Hussain and Triggs, 2012] for details).

Hussain et al. [ul Hussain et al., 2012] distinguish in their work between binary and ternary coding. In the binary coding, the pixel values p are compared to the centre c and set to 1 if $p \geq c$ and to 0 otherwise:

$$f_i = \begin{cases} 1 & \text{if } p \geq c \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

Whereas in the ternary coding, a zone of width $\pm\tau$ around c is quantized to 0 with pixels above the threshold quantized to +1, and pixels below the threshold to -1, that is, the indicator $f(i)$ is replaced by a 3-valued function:

$$f_i = \begin{cases} 1 & \text{if } p \geq c + \tau \\ -1 & \text{if } p \leq c - \tau \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

In this thesis LQP features with an inner and outer disk are used. The LQP geometry is described by a notation such as LQP_3^1, where the subscript indicates the outer neighbourhood radius (here 3 pixels) and the superscript indicates the inner radius (here 1 pixels).

A feature vector of a face image is built by dividing the image into square blocks. Each pixel contributes its vote to the histograms of the four nearest blocks, using bilinear interpolation. By using bilinear interpolation the histogram archives robustness against slight changes in imaging conditions [ul Hussain and Triggs, 2012]. LQP histograms are extracted from all blocks (in this paper into 15×8 blocks) and, then, concatenated into a single histogram. The histogram entries within each block are normalized using the L1-Sqrt norm; i.e., each histogram is normalized to the sum of one and then the square-root of each value is computed. Finally, the histogram is normalised to a length of 1 (Figure 6.2 illustrates the

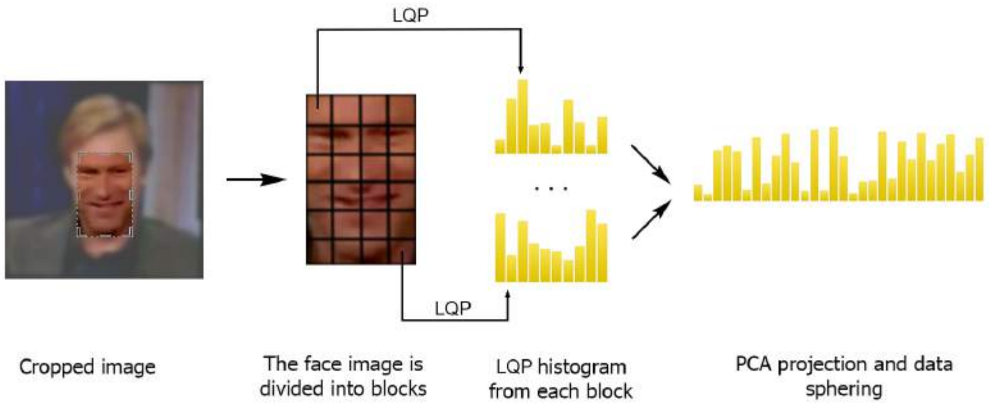

LQP
LQP
Cropped image
The face image is divided into blocks
LQP histogram from each block
PCA projection and data sphering

FIGURE 6.2: Overview of the face recognition framework.

process). This leads to a feature vector that represents both the statistics of the facial micro-patterns and their spatial locations.

The feature histogram has a very high dimensionality; in this work, a dimension of 18,000 (8 blocks × 15 blocks × 150-word codebook). Using a high-dimensionality feature slows down the matching process and always includes the risk of overfitting. Hence, before applying any learning method, Principal Component Analysis (PCA) is used to reduce the dimension of the original feature vector to a more tractable number. PCA is an orthogonal linear transformation that transforms the feature vector to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance, on the second coordinate and so on.

PCA strongly favours the dimensions with high variance by weighting more heavily the components corresponding to large eigenvalues. In the case of face representation, high variance mainly results from illumination and expression changes rather than from other discriminating information [Deng et al., 2005]. A way of reducing the influence of large eigenvectors is the whitening transformation, which normalises the PCA-based feature. Specifically, the PCA-based feature u is subject to the whitening transformation and yields yet another feature set w:

$$w = \Lambda_M^{-\frac{1}{2}} u \tag{6.4}$$

where $\Lambda_M^{-\frac{1}{2}} = diag\{\lambda_1^{-\frac{1}{2}}, \lambda_2^{-\frac{1}{2}}, ..., \lambda_M^{-\frac{1}{2}}\}$. Different authors have shown that applying first PCA and then the whitening transformation improve the recognition result [Deng et al., 2005, Nguyen et al., 2009, Nguyen and Bai, 2010, ul Hussain et al., 2012]. Therefore, all the principal components are divided by the square-roots of their corresponding eigenvalues to have the projected features with the same variance.

6.3 Set-to-Set Similarity

The effect of the whitening process is a change of the projected scale on each component. As pointed out by different authors [Deng et al., 2005, ul Hussain et al., 2012, Nguyen and Bai, 2010, Nguyen et al., 2009], this reduces the accuracy of distance measures like L1, L2 or Mahalanobis distance. It was observed [Cao et al., 2013] that similarity functions, such as bilinear similarity function or the cosine similarity, are not effected by the whitening process. Motivated by these observations, Cao et al. [Cao et al., 2013] combined the bilinear similarity $x_i^T x_j$ and the Mahalanobis distance $(x_i - x_j)^T(x_i - x_j)$ and proposed a Generalized Similarity (GS) to measure the similarity of a feature pair (x_i, x_j):

$$GS(x_i, x_j) = x_i^T x_j - (x_i - x_j)^T(x_i - x_j) \tag{6.5}$$

This section describes two novel set-to-set similarities that utilise the GS for sets. The first is called Generalized Matched Background Similarity (GMBGS) and is built upon the Matched Background Similarity (MBGS) [Wolf et al., 2011]. It is a set-to-set similarity metric designed for comparing the frames of two face videos in order to determine whether the faces appearing in the two sets of faces represent the same subject. The second is called Mean Vector Generalized Similarity (MVGS), and creates a single feature vector for all frames and then use the GS directly.

6.3.1 Generalized Matched Background Similarity (GMBGS)

Given two face videos, X_1 and X_2, where $X_i = \{x_{i1}, \dots , x_{in}\}$ and x_{in} is the n-frame of the i-face-video encoded using a feature transform, their score is computed as follows.

Assume a set $B = \{b_1, \dots, b_n\}$ of background feature samples. This set of samples contains items that are different from both X_1 and X_2, and that are otherwise unlabelled. Firstly, a set of background samples B_1 is determined for X_1 by finding for each member of X_1, its nearest-neighbour in B. If the size of the resulting set of nearest feature vectors is below a predetermined size C, the second nearest neighbour is used, and so on until that size is reached. Afterwards, the set is trimmed such that exactly C frames are collected, hence $|B| = C$. Then a Support Vector Machine (SVM) is trained to distinguish between the two sets X_1 and B_1.

The trained SVM is then used to classify all members of X_2 as either belonging to X_1 or B_1. For each member of X_2 a classification confidence is obtained, reflecting the likelihood that this member represent the same person appearing in X_1. Then the mean (or alternatively, the median, the minimum, or the maximum) is obtained of all members of X_2. This provides a global estimate of the likelihood that the person in the first video X_1 is the same person as in the second video X_2. Afterwards the same procedure is carried out, reversing the roles of X_1 and X_2. The final score is then determined averaging the two scores.

SVM is a large margin classifier that tries to maximize the margin around a separating hyperplane. The SVM takes in labelled training data $\{x_i, y_i\}$, where x_i represent the features and y_i the class label, that could be either 1 or -1. The SVM approach aims at construction a classifier of the form:

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b \tag{6.6}$$

In the dual formulation of the training problem, the coefficients α_i are obtained by minimizing a convex quadratic objective function under constraints:

$$\begin{gathered} max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(x^{(i)}, x^{(j)}). \\ s.t.\ \alpha \geq 0,\ i = 1, \ldots, m \\ \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \end{gathered} \tag{6.7}$$

where α is the Lagrange multiplier, and $K(x_i, x_j)$ is the kernel function (more details about the optimisation process can be found in [Burges, 1998]).

In order to utilise the GS for SVM classification, we incorporate the Generalized Similarity into the kernel function and define the resulting kernel function as:

$$K(x_i, x_j) = x_i^T x_j - (x_i - x_j)^T (x_i - x_j) \tag{6.8}$$

Recent studies [Nguyen and Bai, 2010, Cao et al., 2013] observed that metric learning can improve still image face verification. Motivated by these observations, the similarity metric learning approach proposed by Cao et al. [Cao et al., 2013] was adapted as a pre-processing step for SVM classification. Cao et al. approach learns two positive semi-definite (p.s.d.) matrices M and G for the generalized similarity so that the similarity between positive face pairs is enlarged and that of negative pairs is reduced as much as possible. The kernel becomes now:

$$K_{(M,G)}(x_i, x^j) = x_i^T G x_j - (x_i - x_j)^T M (x_i - x_j)$$

Algorithm 3 summarises the Generalized Matched Background Similarity with metric learning.

```
Function GMBGS($X_1, X_2, B, M, G$)
    Sim1 = OneSideGMBGS($X_1, X_2, B, M, G$);
    Sim2 = OneSideGMBGS($X_2, X_1, B, M, G$);
    Similarity = (Sim1+Sim2)/2;
    return Similarity;
End
Function OneSideGMBGS($X_1, X_2, B, M, G$)
    $B_1$ = FindNearestNeigbors($X_1, B$);
    Model1 = SVMTrain($X_1, B_1$);
    Confidences = SVMPredict($X_2$, Model1);
    Sim = Mean(Confidences);
    return Sim;
End
```

Algorithm 3: GMBGS algorithm with metric learning

6.4 Mean Vector Generalized Similarity (MVGS)

Given a face track $X_i = \{x_{i1}, \dots, x_{in}\}$, the assumption of the Mean Vector Generalized Similarity (MVGS) is that all faces in a face track belong to the same person. Under this assumption it can be expected that there is a high degree of correlation amongst the feature vectors of a track. In fact, with enough similarity between the faces in a track, nearly the same feature vector can be expected. This fact can be exploited by modelling each face track by a linear combination of the feature vectors in order to create a single feature vector for all frames. Mathematically, this means that a track is represented by the mean feature vector:

$$\bar{X}_i = \frac{1}{|X_i|} \sum_{n=1}^{|X_i|} x_n \tag{6.9}$$

The advantage of a single feature vector is that the GS can be exploited directly resulting in the following formulation for the MVGS:

$$MVGS(\bar{X}_i, \bar{X}_j) = \bar{X}_i^T \bar{X}_j - (\bar{X}_i - \bar{X}_j)^T (\bar{X}_i - \bar{X}_j) \tag{6.10}$$

The similarity metric learning approach proposed by Cao et al. [Cao et al., 2013] can then also be used for the MVGS:

$$MVGS_{(M,G)}(\bar{X}_i, \bar{X}_j) = \bar{X}_i^T G \bar{X}_j - (\bar{X}_i - \bar{X}_j)^T M (\bar{X}_i - \bar{X}_j) \tag{6.11}$$

6.5 Experimental Setup and Evaluation

In this section the performance of the proposed face recognition framework is evaluated on three benchmarks: the YouTube Face Database [Wolf et al., 2011], the YouTube Celebrities Dataset [Kim et al., 2008] and the ChokePoint Dataset [Wong et al., 2011]. In addition the ACFFace-5 face detector is evaluated on the Face Detection Dataset and Benchmark [Everingham et al., 2010] and the Annotated Faces in the Wild Dataset [Mathias et al., 2014].

6.5.1 Face Detection Evaluation

The evaluation of the ACFFace-5 detector is done on two datasets. The first is the Face Detection Dataset and Benchmark (FDDB), which is a dataset of face regions designed for studying the problem of unconstrained face detection. The dataset contains the annotations for 5171 faces in a set of 2845 images taken from the Faces in the Wild dataset. The evaluation results are shared using ROC curves, which are computed using the Pascal VOC protocol [Everingham et al., 2010]. In the Pascal VOC protocol a binary match/non-match label is computed for each detection, where a match label requires at least 50% overlap ratio of the intersection of two regions against the union of the two regions. The ground truth of the FDDB are elliptical regions, while the output of the proposed method are rectangles. Changing the output format form rectangles to elliptical regions immediately increased the overlap and thus this procedure was applied on the detector output rectangles. Figure 6.3 shows the ROC curve generated by ACFFace-5 detector in comparison to available results on the benchmark. The related papers to each algorithm can be found at [Jain and Learned-Miller, 2010].

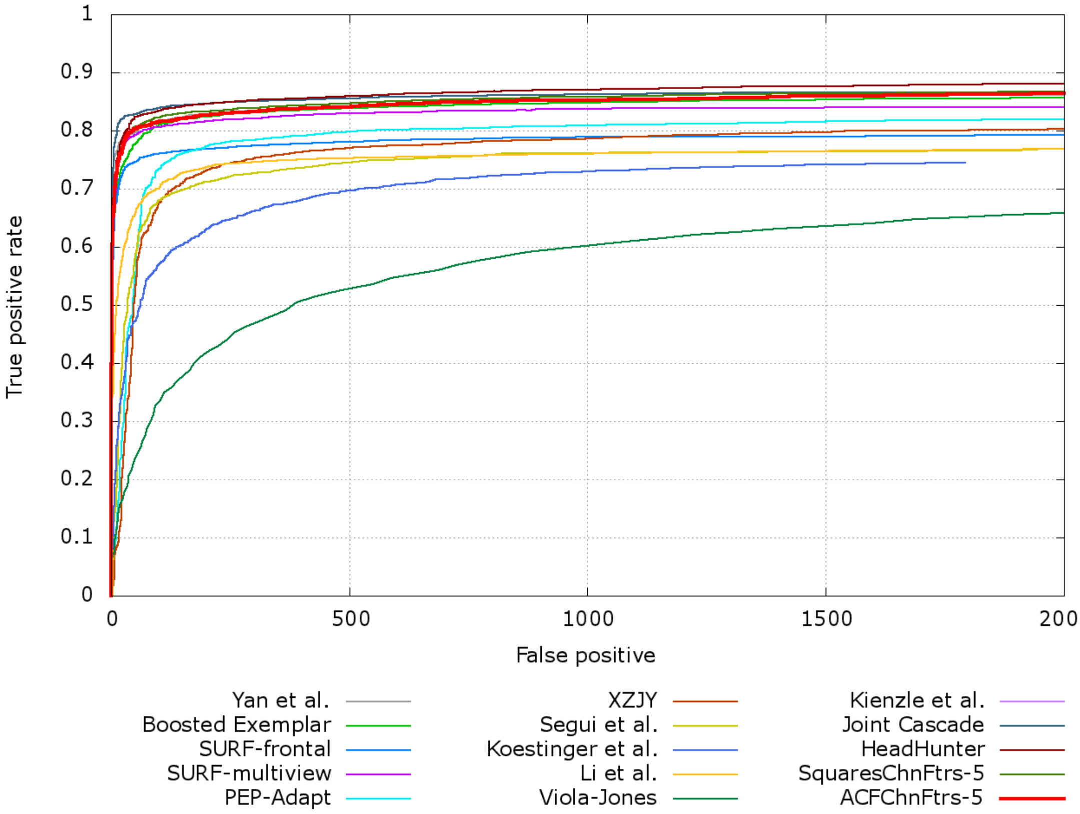


Figure 6.3: Discrete score ROC curves for different methods on the FDDB dataset.

The second dataset is the Annotated Faces in the Wild (AFW), which consists of 205 images with 1025 annotated faces. For evaluation, the evaluation toolbox as well as the annotations provided by Mathias et al. [Mathias et al., 2014] were used. The results are evaluated using also the Pascal VOC criterion [Everingham et al., 2010], and quality is summarised using the Average Precision (AP). Figure 6.4 shows the ROC curve and the AP generated by ACFFace-5 detector in comparison to available results on the benchmark. Sample detection results are shown in Figure 6.5. The results on both datasets show that the ACFFace-5 detector reaches state-of-the-art performance and achieves similar results to the SquaresChnFtrs-5 [Mathias et al., 2014]. However, the aim of the proposed detector was not to increase accuracy but speed.

Since the aim of the proposed detector was not to increase accuracy but to develop

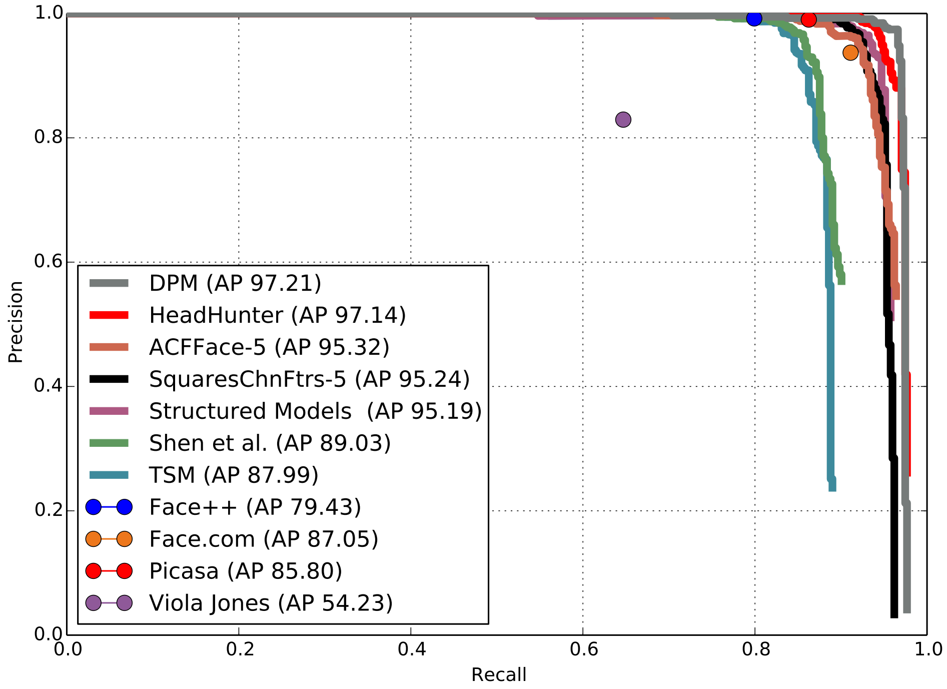


Figure 6.4: Results on the AFW dataset.

a detector with low computational costs, further comparison were made of detection speed between the ACFFace-5 detector, the Viola Jones face detector and the ACFFace-5-view detector. The ACFFace-5-view[1] was trained the way as the ACFFace-5 but only view-specific classifiers were trained that are later eventuated individually. The ACFFace-5-view detector was included in the results in order to show the speed advantage of the combination of a general classifier with view-specific classifiers to view-specific classifiers only. The ACFFace-5 is implemented using Matlab and therefore the Matlab 2014a (vision.CascadeObjectDetector) implementation of the Viola Jones detector was used.

The detectors were tested on three video clips. The first is the video 0098_03_006_al_gore from the YouTube Celebrities Dataset (see Section 6.5.2.1) which has 233 frames with a resolution of 320×240 pixels, the second is the video AVSS_AB_Easy_Divx from i-Lids dataset [AVSS, 2007] which has 5474 frames with a resolution

[1] The ACFFace-5-view detector was not include in the detection rate evaluation because both detectors achieve approximately the same results.

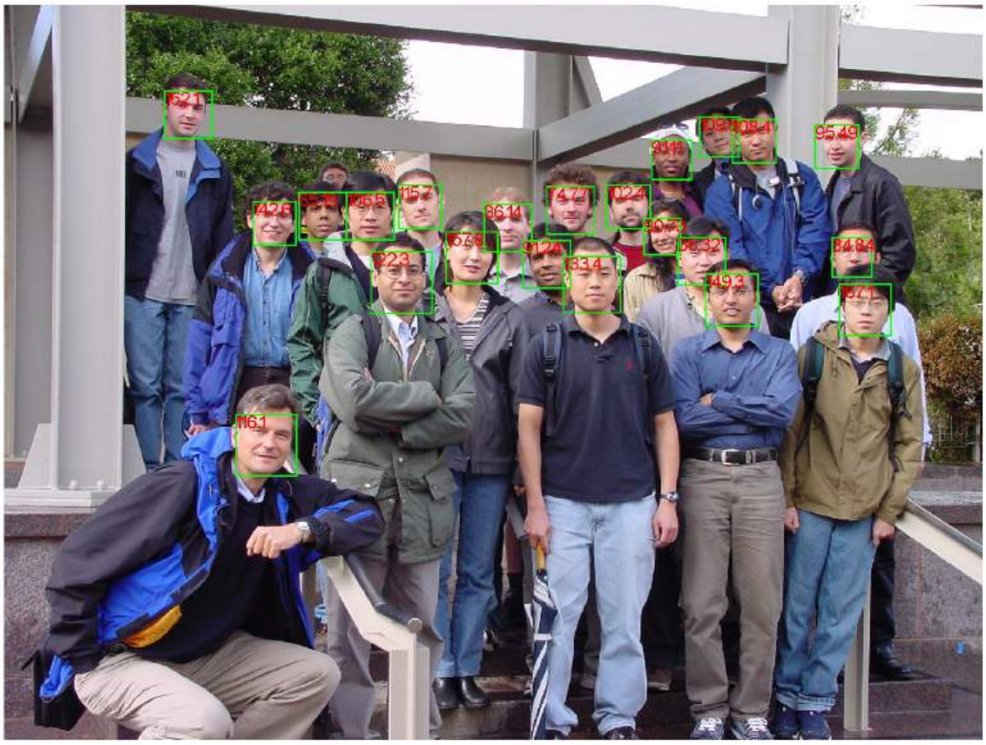


FIGURE 6.5: Sample result of the ACFFace-5 detector.

of 720 × 576. The third is a video downloaded from [Gallery, 2013] called James Nares STREET which has 4126 frames with resolution of 1920 × 1080 pixels. The test machine was a desktop computer with an Intel Core i5-3470 CPU with 3.2 GHz and 8GB RAM.

The results are presented in Table 6.1. It can be seen that the ACFFace-5 detector is on average twice as fast as the ACFFace-5-view detector. This shows that the proposed combination of a general classifier and view-specific classifiers improves speed while maintaining high accuracy. Compared to the Viola Jones face detector, the ACFFace-5 is on average slightly slower, however since the ACFFace-5 has a 41.09% higher AP this trade-off is acceptable (see Figure 6.3 and 6.4).

Table 6.1: Speed comparison of the face detector.

Video	ACFFace-5	ACFFace-5-view	Viola Jones
0098_03_006_al_gore (320 × 240)	108.39 fps	59.95 fps	106.47 fps
AVSS_AB_Easy_Divx (720 × 576)	27.84 fps	14.82 fps	38.32 fps
James Nares: STREET (1920 × 1080)	5.28 fps	2.99 fps	6.22 fps

6.5.2 Face Recognition Evaluation

The GMBGS and MVGS were evaluated on three datasets. Unless otherwise described, for all tests the parameters used are described in the following: All the images in the face sets were histogram equalized and cropped to have dimensions of 80×150 pixels. Tests were made with 80×140 pixels, 90×140 pixels, 80×150 pixels, 90×150 pixels, and 90×160 pixels, where 80×150 pixels achieved the best results. The original and the flipped versions of the face images were used.

For all experiments, the total number of PCA components has been fixed to 300 as suggested in [Cao et al., 2013]. As a default, a 150-word-codebook-based LQP_3^1 descriptor with a tolerance value of 7, i.e. $\tau = 7$ (see Figure 6.6 for a comparison of different τ values on the YouTube Face Database) and a block size of 10 × 10 pixels is used. This results in a feature dimensions without PCA of 36,000 (8 blocks × 15 blocks × 2 × 150-word codebook). For selecting the background sets for the MVGS, the nearest neighbours were used and the maximum background set size was set to 250. In order to be comparable with other methods, results are reported using aligned version of the face images that are provided by the dataset if available otherwise the face images are aligned as described in Section 6.2.2.

6.5.2.1 YouTube Face Database

The YouTube Face Database [Wolf et al., 2011] is a database designed for studying the problem of unconstrained face verification in videos. The dataset contains 3,425 videos of 1,595 different people and some sample frames can be seen in Figure 6.7. All the videos were downloaded from YouTube. The average length of a video clip is 181.3 frames. In addition, the dataset provides for every video frame the face position, the three rotation angles of the head and an aligned version of

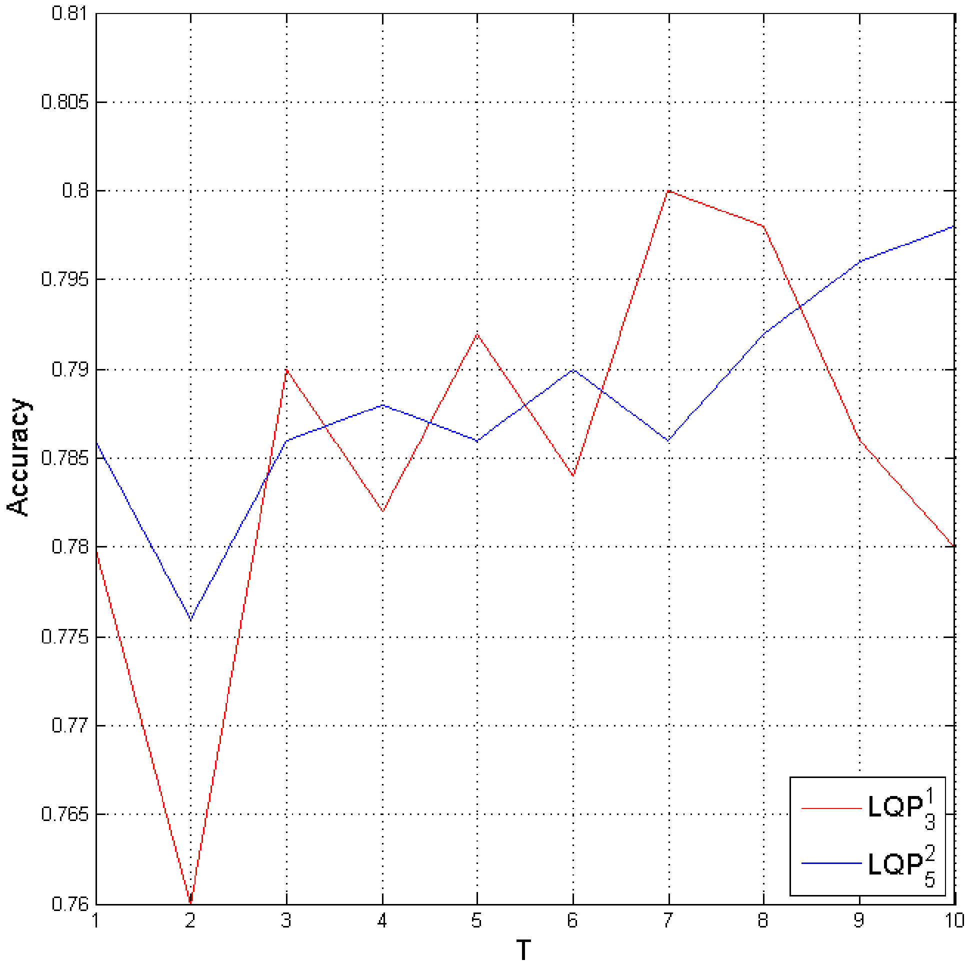


FIGURE 6.6: Recognition rate of the LQP_3^1 and LQP_5^2 with $\tau = \{1 - 10\}$ on the YouTube Face Database.

the face. The quality of the videos varies widely in the dataset: some videos are of high quality, while others are very blurred.

The dataset provides a ten-fold, cross-validation, pair-matching (same/not-same) test. The test is divided into 10 splits; each split contains 250 same and 250 not-same video pairs. The splits are independent from each other, thus, each person appears only in one split. In all experiments the aligned face images are used.

FIGURE 6.7: YouTube Face Database sample frames.

TABLE 6.2: YouTube Face Database results.

	Accuracy ± *SE*	AUC	ERR
MBGS L2 mean, LBP [Wolf et al., 2011]	76.4 ± 1.8	82.6	25.3
MBGS+SVM- [Wolf and Levy, 2013]	78.9 ± 1.9	86.9	21.2
APEM-FUSION [Li et al., 2013a]	79.1 ± 1.5	86.6	21.4
STFRD+PMM [Cui et al., 2013]	79.5 ± 2.5	88.6	19.9
VSOF+OSS (Adaboost) [Mendez-Vazquez et al., 2013]	79.7 ± 1.8	89.4	20.0
DDML (combined) [Hu et al., 2014]	82.3 ± 1.5	90.1	18.5
DeepFace-single [Taigman et al., 2014]	91.4 ± 1.1	96.3	8.6
GMBGS (proposed)	81.60 ± 1.0	88.8	19.0
$GMBGS_{(M,G)}$ (proposed)	82.0 ± 1.1	89.4	18.6
MVGS (proposed)	78.36 ± 2.0	85.8	22.4
$MVGS_{(M,G)}$ (proposed)	83.44 ± 1.8	91.60	16.8

The final results are presented in Table 6.2. ROC curves are presented in Figure 6.8. A ten-fold, cross-validation, pair-matching (same/not-same) test was used to obtain these results. Each time, nine sets were used for training, and tested on the tenth set. Similar to Wolf et al. [Wolf et al., 2011], results are reported by constructing a ROC curve for all splits together, by computing statistics of the ROC curve (area under curve and equal error rate) and by recording the standard errors for the average recognition for the 10 splits. For the sake of clarity only the curves of the best performing methods were plotted.

The best result on the YouTube Face Database is achieved by the method DeepFace-single [Taigman et al., 2014], however the evaluation of DeepFace-single was not performed with the aligned images provided by the dataset. Instead they used a novel 3D-warped version of the face. Unfortunately neither the algorithm, nor the

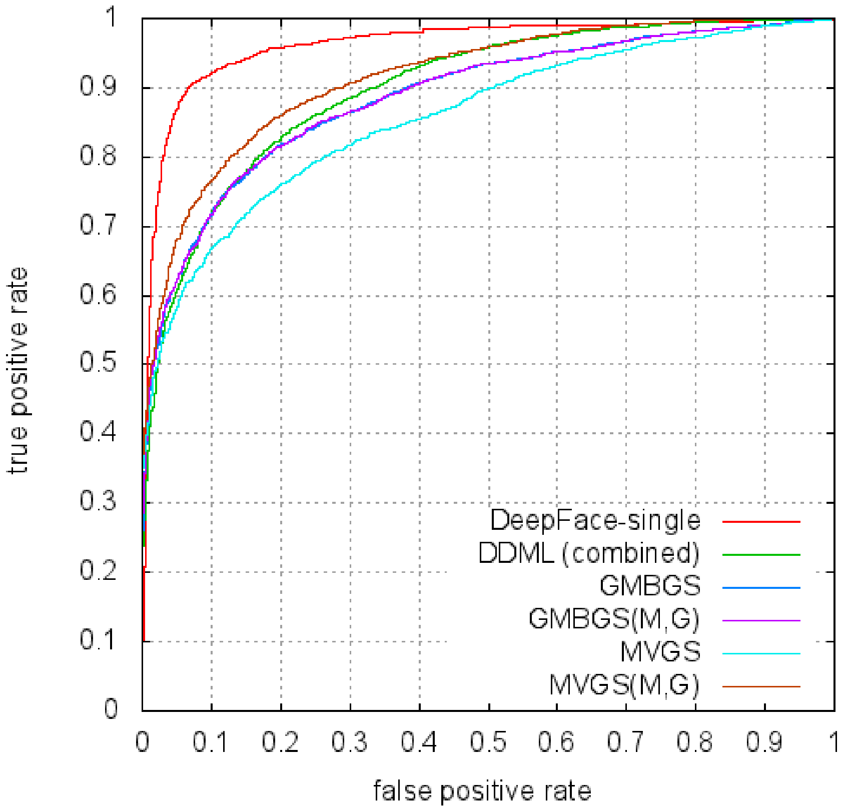


Figure 6.8: ROC curves averaged over 10 folds from the YouTube Face Database.

warped images used are publicly available. Since alignment can have large impact on the recognition rate, the results should not be compared directly.

In both cases learning a metric improves the accuracy. However, for the GMBGS schema only a small improvement of 0.4% was reached, while the MVGS schema showed a significant improvement of 5.08%. The accuracy of the MVGS with a learned metric gives the best result on the original dataset as provided by [Wolf et al., 2011] and improves the recognition rate by 1.14% compared to the second best method DDML (combined) [Hu et al., 2014].

6.5.2.2 YouTube Celebrities Dataset

The YouTube Celebrities Dataset [Kim et al., 2008] consists of 47 celebrities (actors and politicians) in 1910 video clips downloaded from YouTube and manually segmented into sequences where the celebrity of interest does appear. The segmented dataset consists of about 1500 video clips: each one containing hundreds of frames. Some sample frames can be seen in Figure 6.9. The frame sizes range from 180 × 240 pixels to 240 × 320 pixels. The dataset is challenging due to pose, illumination, and expression variations, as well as low resolution and high compression rates.

The proposed tracking framework successfully tracked 96% of the videos as compared to 92% tracked in the paper of Ortiz et al. [Ortiz et al., 2013] and 80% tracked in the paper of [Kim et al., 2008]. The standard experimental setup selects 3 training clips, 1 from each unique video and 6 test clips, 2 from each unique video per person. A metric was not learned for this dataset because the number of subjects was not sufficient in order to train a metric that did not overfit. Table 6.3 summarizes the evaluation results on YouTube Celebrities. Both algorithm achieve better results then the state-of-the-art algorithm, MVGS achieved the best results, it improves the recognition rate by 3.32%.

Table 6.3: YouTube Celebrities Dataset results.

Method	Accuracy
HMM [Kim et al., 2008]	71.24
MDA [Wang and Chen, 2009]	67.20
SANP [Hu et al., 2011]	65.03
COV+PLS [Wang et al., 2012]	70.10
UISA [Cui et al., 2012]	74.60
MSSRC[Ortiz et al., 2013]	80.75
GMBGS (proposed)	81.74
MVGS (proposed)	84.07

6.5.2.3 ChokePoint Dataset

The ChokePoint Dataset [Wong et al., 2011] was created using an array of three cameras placed above two portals (natural choke points in terms of pedestrian traffic) to capture subjects walking through each portal in a natural way. The

FIGURE 6.9: YouTube Celebrities Dataset sample frames.

dataset consists of 25 subjects (1 male and 6 female) in portal 1 and 29 subjects (23 male and 6 female) in portal 2. The recording of portal 1 and portal 2 are one month apart. The dataset has frame rate of 30 fps and the image resolution is 800×600 pixels. In total, the dataset consists of 48 video sequences and 64,204 face images. Some sample frames can be seen in Figure 6.10. In all sequences, only one subject is presented in the image at a time.

The dataset provides a verification protocol. In this protocol, video sequences are divided into two groups (G1 and G2), where each group played the role of development set and evaluation set in turn. In each group, all possible genuine and imposter pairs were generated. Parameters and thresholds are first learned on the development set and then applied on the evaluation set. The average verification rate (accuracy) is used for reporting results [Wong et al., 2011]. More details about the dataset and the verification protocol can be found in [Wong et al., 2011].

Furthermore, a metric was not learned in this dataset since the number of subjects was not sufficient in order to train a metric that did not overfit. All results are presented in Table 6.4. The results show the advantage of the proposed methods which both achieved an accuracy of 100%.

FIGURE 6.10: ChokePoint Dataset sample frames.

TABLE 6.4: ChokePoint Benchmark results.

Method	Accuracy
Asym_shrp [Wong et al., 2011]	75.4
Gabor_asym [Wong et al., 2011]	84.0
DFFS [Wong et al., 2011]	83.4
Patch-based [Wong et al., 2011]	86.7
LBP + LASSO [Fusco et al., 2013, Zini et al., 2014]	93.1
LBP + MC-GrpLASSO [Fusco et al., 2013, Zini et al., 2014]	96.9
GMBGS (proposed)	100.0
MVGS (proposed)	100.0

6.6 Conclusion

This paper has presented a complete framework for video face recognition, which includes face detection, multi-target face tracking, face alignment and video face recognition. The contributions include a novel face detector called the ACFFace-5 detector which combines a general classifier with five view-specific classifiers. The detector was evaluated on two datasets, where it reached state-of-the-art results. In addition, the computation cost was compared with a face detector that only uses view-specific classifiers and showed reductions of approximately 50%.

Furthermore, two novel set-to-set similarity measures, the Generalized Matched Background Similarity and the Mean Vector Generalized Similarity were presented. The GMBGS similarity measure uses a SVM in order to distinguish between same and not same face sets and defines the GS as kernel function. Whereas the MBGS uses a linear combination of feature vectors in order to use the GS directly. In both cases, LQP histograms, on which whitened PCA is applied, are used as feature vectors. Finally, the algorithms were evaluated on three existing datasets (YouTube Faces, YouTube Celebrities, and ChokePoint), where recognition rates of 83.44%, 84.07% and 100% were achieved.

Chapter 7

Human Tracking and Recognition in the Context of a Fisheye Camera

7.1 Introduction

One of the main difficulties of human tracking systems, particularly in indoor environments like the LPH, is the limited field of view of a single camera. A solution for this problem is to use multiple cameras to monitor a wider field of view. However, using multiple cameras has its own challenges. For example, in order to compute the topology of a camera network, a precise calibration step is necessary. Likewise if humans are observed in different camera views, an association step has to be applied [Wang, 2013]. It is noted that it is not possible in every environment to install multiple cameras.

Another solution to address this problem is to use fisheye cameras rather than conventional perspective cameras. A fisheye lens is an ultrawide-angle lens that produces strong visual distortion intended to create a wide panoramic or hemispherical image (see Figure 7.1). A single fisheye camera mounted on the ceiling is able to capture the entire room whereas a perspective camera with a field of

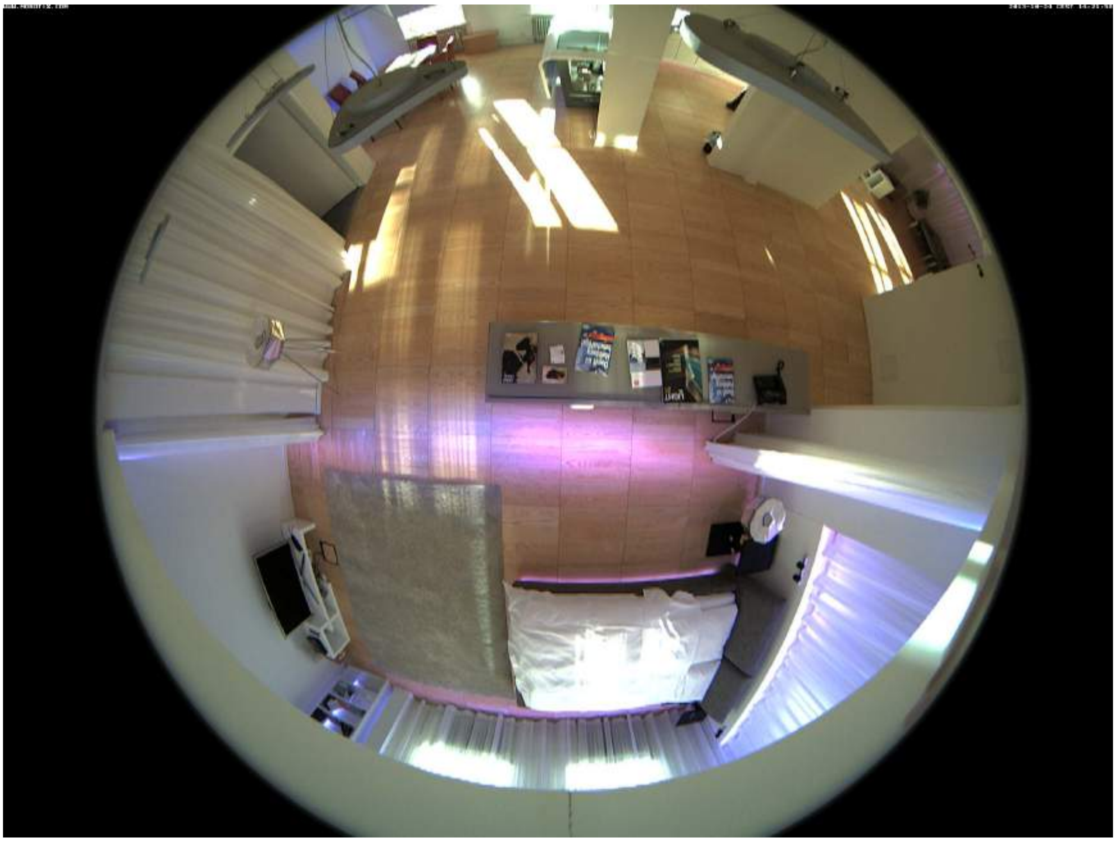


Figure 7.1: A fisheye image example

view of 45 degree is only able to capture a small part of the room. Figure 7.2 illustrates the problem. The left image shows perspective cameras which are only able to capture a limited field of view, even two cameras are not able to capture a entire room. The right image shows a single fisheye camera mounted on the ceiling which is able to capture the entire room. However, such cameras have the drawback that captured images suffer from strong distortion and perspective effects. Thus, for such cameras non-standard algorithms for human detection and tracking are required.

Recent methods [Wang, 2006, Kubo et al., 2007, Saito et al., 2010, Yuan et al., 2011, Vandewiele et al., 2012] have focused on applying detection and tracking algorithms directly on the fisheye images. However, doing so has several drawbacks such as the fact that fisheye images suffer strong distortion and perspective effects. That is why the recent approaches mostly rely on simple detection algorithms like blob detection with background subtraction or ellipse fitting. Unfortunately, in real-life scenarios where illumination changes occur and objects other than humans

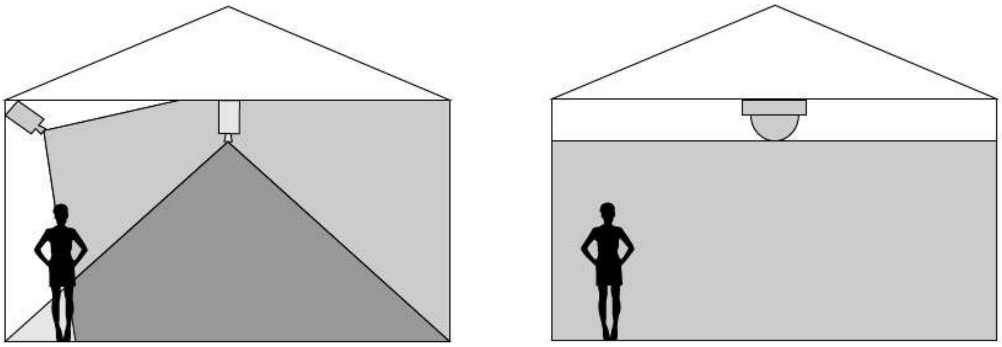


Figure 7.2: Perspective cameras and fisheye camera field of view.

move through the scene, such simple models tend to fail. That is why this thesis proposes a novel approach that first projects the fisheye image into a set of perspective images and then applies standard human detection on these projections.

Firstly, a camera model is described that allows fast projection between a fisheye image and a corresponding set of perspective images. This allows us to apply a standard detection algorithm. Then an algorithm is proposed that automatically generates from annotated heads in fisheye images, sets of aligned heads in perspective images that are then used to train a combined multi-view detector. In order to achieve such conditions, a novel combination of unsupervised alignment and unsupervised subcategory learning based on SVMs is proposed.

Finally, a system is proposed for tracking humans on a calibrated fisheye camera. Therefore, multi-target tracking is applied. In order to evaluate the algorithm a novel dataset for tracking humans in fisheye videos is created on which a MOTA of 69.53 % is achieved. In addition, evolution is performed on the Bomni-DB which is an omnidirectional video tracking database where a MOTA of 78.55% is achieved.

7.2 Camera Model

This work is based on tracking by detection. In general, human detectors are obtained by learning differences between human and not human. For the human body, constructing such a classifier using hemispherical images becomes extremely difficult due to the strong distortion and the complexity of articulated motion during walking. Hence, the detector is not applied on the fisheye image itself but instead on perspective projections. Therefore, the fisheye image is first projected into an unit sphere that is then used to construct virtual multi-camera system containing 9 perspective images with view of 45 degree. In this section the details are explained.

7.2.1 Centre Perspective Projection vs. Fisheye Projection

In the central perspective projection model the angle of incidence of the ray from an object point α_1,β_1 is equal to the angle between the ray and the optical axis α_2,β_2 within the image space (grey line Figure 7.3) [Schwalbe, 2005].

In order to realise a wider opening angle for a lens, the principle distance (distance between lens and image plane) has to be shortened. This can only be done to a certain extent in the central perspective projection model. At an opening angle of 180 degrees, a ray with an incidence angle of 90 degree would be projected onto the image plane at infinite distance to the principle point (red ray in Figure 7.3) independently of how short the principle distance is. In order to enable a complete projection of the hemisphere onto the image plane with a defined image format, a different projection model becomes necessary such as the fisheye projection model [Schwalbe, 2005].

The mathematical model of a fisheye projection assumes that the distance between an image point and the principle point is linearly dependent on the angle of incidence of the ray from the corresponding object point (see Figure 7.4). Thus the incoming object ray is refracted in the direction of the optical axis [Schwalbe, 2005]. This principle allows the fisheye camera to capture images with a field of view of 180 degree. In order to realise this, a system of lenses is used as shown in Figure 7.5.

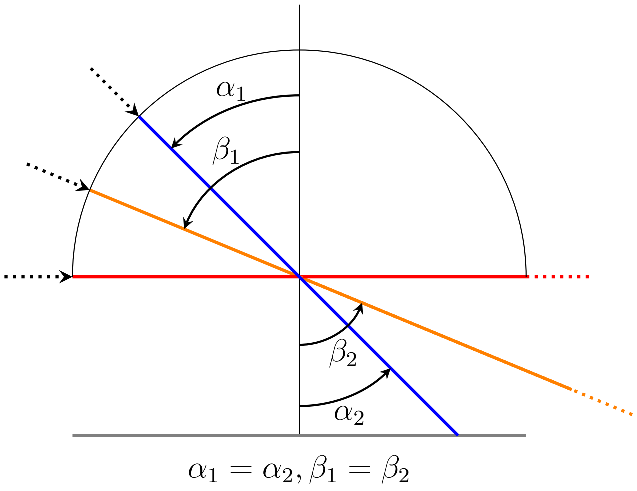


FIGURE 7.3: Central perspective projection.

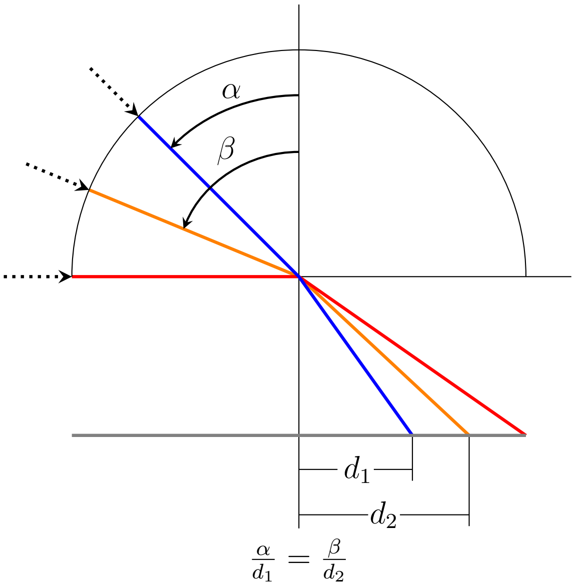


FIGURE 7.4: Fisheye projection model.

7.2.2 Reconstruction of the 3D Vector Space

In order to construct a perspective projection from a fisheye image, the 3D vector space has be to reconstructed. This can be done by projecting the fisheye image into a sphere (Figure 7.6 illustrates the relationship). Starting with the point p on the image plane with the coordinate (u, v) with respect to the centre of the fisheye image, the aim is to find the corresponding 3D vector P with the coordinates (x, y, z). If the axes of the camera and the lenses are perfectly aligned, then x and y are proportional to u and v respectively.

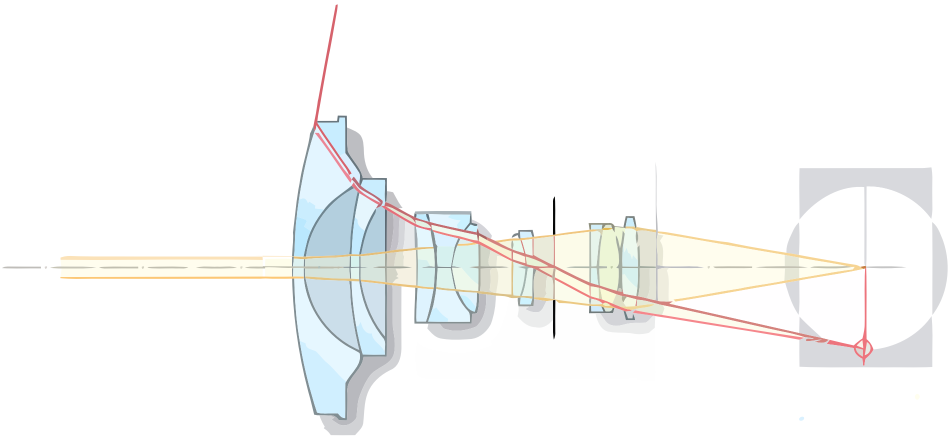


FIGURE 7.5: A fisheye lens system of the AF DX Fisheye-Nikkor 10.5mm f/2.8G ED (taken from: [Toscani, 2015]).

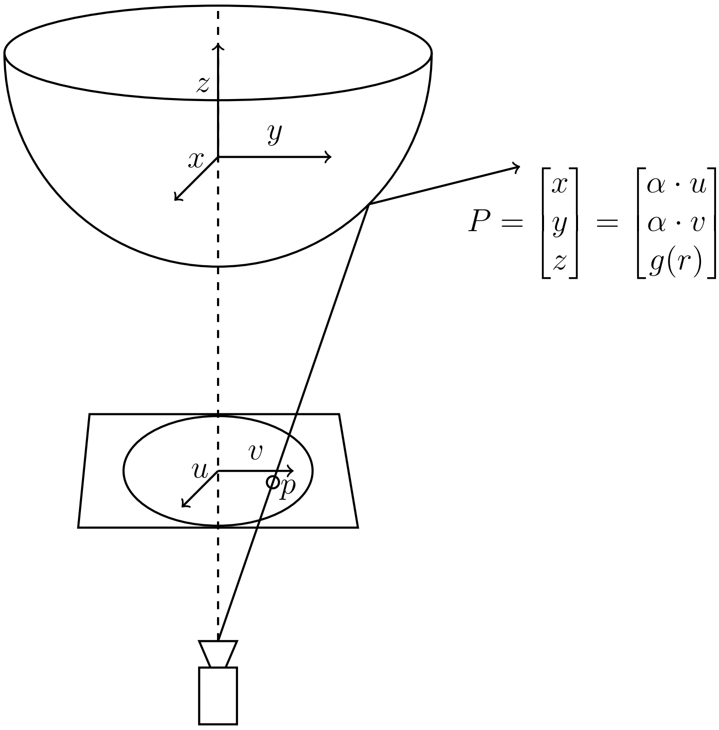

z
y
x
$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \\ g(r) \end{bmatrix}$
v
u
p

FIGURE 7.6: Relationship between fisheye image and a sphere.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \end{bmatrix}, \alpha > 0 \tag{7.1}$$

Then the mapping between the 3D vector $P = (x, y, z)^T$ and an image point $p = (u, v)^T$ is given by:

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \\ g(r) \end{bmatrix} \tag{7.2}$$

where r is the distance of the point p to the image centre $r = \sqrt{u^2 + v^2}$ and g is a mapping function that depends on the type of fisheye lens. Since P is a vector the vector α can be included into the function g (see [Scaramuzza et al., 2006] for proof):

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ g(r) \end{bmatrix} \tag{7.3}$$

In this thesis the generalized mapping function described by Scaramuzza et al. [Scaramuzza et al., 2006] is used, which approximates the mapping by a polynomial of degree 4:

$$g(r) = a_0 + a_1 \cdot r + a_2 \cdot r^2 + a_3 \cdot r^3 + a_4 \cdot r^4 \tag{7.4}$$

where the coefficients $a_0 - a_4$ are parameters that have to be determined by a calibration step which in done in this work with the Omnidirectional Camera Calibration Toolbox[1].

Natural errors in the camera settings, as well as misalignments between the camera and the lenses, may still cause undesired distortions. In order to model these imperfections, an affine transformation is used that describes the relation between real distorted coordinates (u', v') and ideal undistorted ones (u, v):

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} a & d \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} xc' \\ yc' \end{bmatrix} \tag{7.5}$$

[1] https://sites.google.com/site/scarabotix/ocamcalib-toolbox

where a, d, e, xc' and yc' are parameters that are also determined by in the calibration step. More details of the point to 3D mapping and the calibration step can be found in [Scaramuzza et al., 2006].

Once the mapping function $g(r)$ is estimated, the mapping between a point on the fisheye image plane and the corresponding 3D vector can be calculated with Equation 7.3. In order to create a perspective image from the fisheye image, a virtual perspective camera plane is created with the negative z-axis of the sphere as optical axis. The centre point of this camera is then:

$$C = \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} \tag{7.6}$$

where f is the the focus length. A point $p = (u_p, v_p)$ on the perspective image plane can then be defined by a 3D vector on the virtual perspective camera plane:

$$V_p = \begin{bmatrix} u - c_x \\ v - c_y \\ f \end{bmatrix} \tag{7.7}$$

where $c_x = \frac{w}{2}$ and $c_y = \frac{h}{2}$ and w, h are the width and the height of the image plane. This vector can now be used to find the corresponding image point in the fisheye image plane by using the inverse of Equation 7.3. The mapping function can be used to map points between fisheye image plane and the perspective image and vice versa.

The resulting image shows the view from a perspective camera at the centre of the sphere looking down through the south pole. Other view can be created by rotation of the virtual perspective camera plane in different directions:

$$P_p^* = RV_p \tag{7.8}$$

where R is a 3×3 rotation matrix:

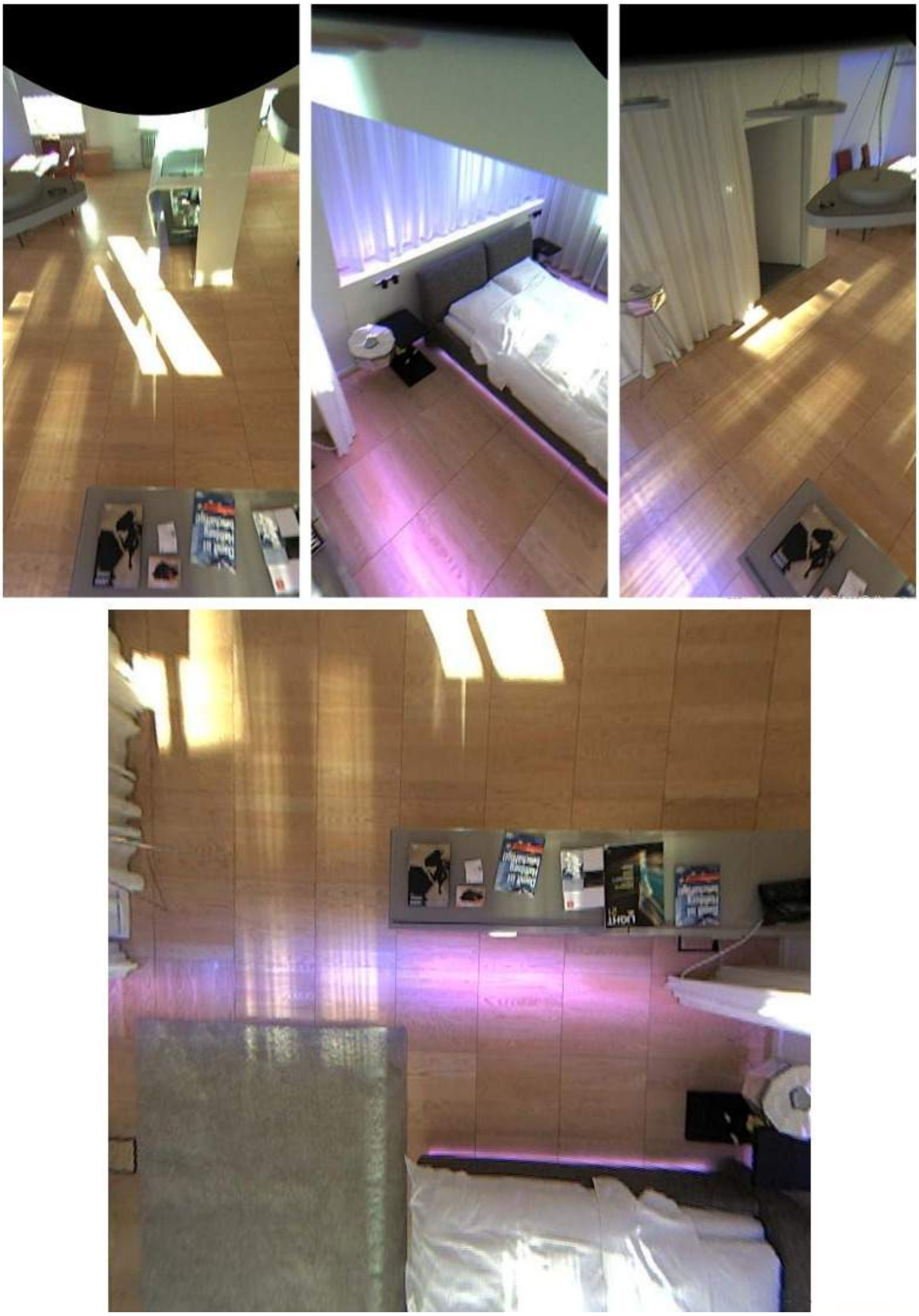


Figure 7.7: Sample perspective projections from the fisheye image in Figure 7.1.

$$R = R_x(\gamma)R_y(\beta)R_z(\alpha) \tag{7.9}$$

with

$$R_x(\gamma) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{pmatrix} \tag{7.10}$$

$$R_y(\beta) := \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \tag{7.11}$$

$$R_z(\alpha) := \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7.12}$$

A projected perspective image has the same limitation as an image taken with a perspective camera, thus allowing only a small field of view. Therefore the fisheye image model by 9 perspective images, 8 side view images and a top down image is used in this thesis. Figure 7.7 shows some example; the examples were created from left to right and top to bottom with the following angles: $1 : (\gamma = 0°, \beta = 52°, \alpha = 0°), 2 : (\gamma = 0°, \beta = 52°, \alpha = -90°), 3 : (\gamma = 0°, \beta = 52°, \alpha = 45°), 4 : (\gamma = 0°, \beta = 0°, \alpha = 0°)$.

The mapping between the original image and the perspective image can be easily stored in a Lookup Table (LUT). A projection takes then less than a 1 ms on our test system (see Section 7.5 for experimental settings).

7.3 Detector

Due to the orientation of the ceiling mounted camera and the wider opening angle for the lens, humans are visible under a wide range of views. For reliable human detection, it is necessary to create detectors under different camera views. Recent work [Kim and Cipolla, 2008] has shown that dividing the classification problem into sub-categories (views) allows better modelling and improves classification. These sub-categories are often specified manually (e.g. frontal or profile faces as in Chapter 6), but can also be determined automatically using unsupervised clustering. In this section a combined head detector for perspective projection of fisheye images is proposed, that automatically generates sets of head and upper-body sub-categories detectors that are then combined with a body detector in order to create a robust head detector for perspective projection of fisheye images.

Starting from a set of annotated heads $D^{(F)} = \{d_1^{(F)}, d_2^{(F)}, ..., d_n^{(F)}\}$ in fisheye images (see first and third image in Figure 7.8), the aim is to project these heads into perspective images and then align them and cluster them into sub-categories. The creation of the perspective projections of the head detections is done as follows: Firstly each fisheye image is projected into a set of perspective images as described in Section 7.2 and then for each head annotation the four corner points are projected into the perspective images using the inverse of Equation 7.8. Finally the enclosing rectangles of the corner points are computed, resulting into a set of perspective projected detections $D^{(P)} = \{d_1^{(P)}, d_2^{(P)}, ..., d_n^{(P)}\}$.

Figure 7.8 illustrates the process. The first and the third image are annotated heads in a fisheye image. The second and the fourth are perspective projections of this annotations. The black points are the projected corners. It can be seen that the projection in the first pair is as expected whereas in the second pair a margin appears around the head. In order to remove this undesired variability, a novel algorithm is introduced that clusters the head images into sub-categorises while simultaneously aligning them.

Suppose there exist N heads. Let x_i be a feature vector describing the i head under the transformation $t^i = \{t_x^i, t_y^i, t_s^i\}$, where t_x is the x-translation, t_y the y-translation and s the scaling (uniform in x and y). Given this set of head examples and a set of negative examples, the alignment algorithm first trains a SVM to discriminate heads form the negative examples. Once the SVM has been trained, a probabilistic output can be computed using the sigmoid function:

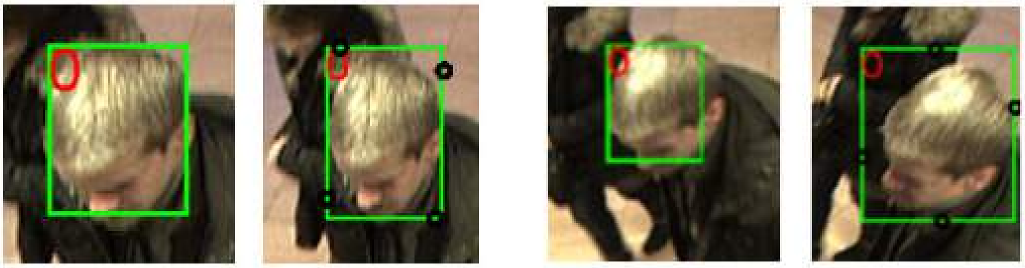


FIGURE 7.8: Rectangle projection.

$$P(y = 1|f(x)) = \frac{1}{1 + exp(Af(x) + B)} \tag{7.13}$$

where $f(x)$ is the output of the SVM:

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

where α is the Lagrange multiplier, and $K(x_i, x_j)$ is the kernel function (more details about the optimisation process can be found in [Burges, 1998]). The parameters A and B are learned by the sigmoid fitting approach that was proposed in [Platt, 1999].

In order to align the head images for each head image, the transformation that maximizes the probability is computed. In this work, this maximization is done by hill climbing over the transformations. At each iteration, the transformation is selected that increases the probability until conversion. In order to align the images further, the process is repeated with the aligned images for N iterations. In order to not get stuck in a local minima, the algorithm is restarted on multiple starting points. The complete algorithm is shown in Algorithm 4 and some sample images can be found in Figure 7.9.

In order to simultaneously cluster the images into sub-categories, the alignment algorithm is incorporated into the sub-categories clustering algorithm that was proposed by Hoai and Zisserman [Hoai and Zisserman, 2013]. Given a set of positive and negative examples of a category, their approach simultaneously determines

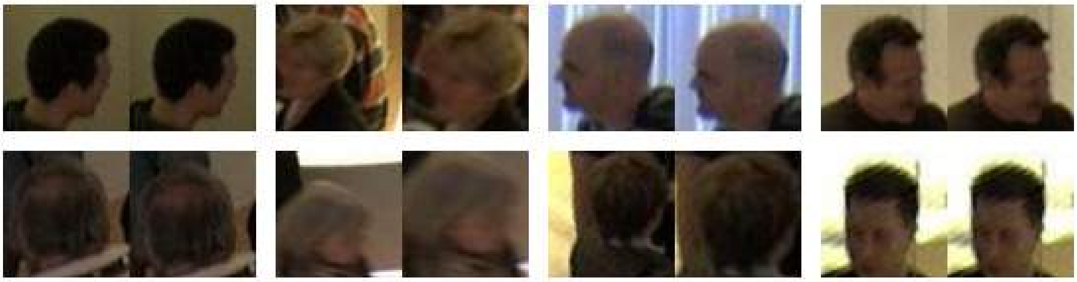


FIGURE 7.9: Sample pairs of original (left) and aligned head images (right).

the cluster label of each positive example, whilst learning a SVM for each cluster, discriminating it from the negative examples. Both algorithms are combined as follows: first an initial alignment is done on all images as it was described above. Then sub-categories are computed on the aligned images using Hoai and Zisserman method. Afterwards the SVM that was trained while clustering is used to align each sub-categories among themselves. Then the algorithm starts again with sub-category clustering and runs until conversion.

The learned sub-categories are then used to train a detector that combines a general classifier with a multi component classifier. This is done in the same way as it was done for the ACFFace-5 detector described in Chapter 6. Ten feature channels were used: normalised gradient magnitude, histogram of oriented gradients (6 channels) and LUV colour channels; the block size was set to 4 × 4. AdaBoost was used to train and combine 2048 depth-two trees over the candidate features (channel pixel lookups) in each window. The trainings window size was set to 28 × 28 pixels with an addition padding of 8 pixels to each side.

After evaluation the resulting detections are back projected in the fisheye image.

```
Input: $I$ – Set of positive images
       $I_{neg}$ – Set of negative images
       $N$ – Number of images in I
       $NIterations$ – Number of iterations
Result: Set of aligned images $I$
for $i \leftarrow 1$ to $N$ do
    $t_k^i \leftarrow [0, 0, 0]^T$;
end
for $n \leftarrow 1$ to $NIterations$ do
    SVMModel ← TrainSVM(PositiveImageSet, NegativeImageSet);
    for $i \leftarrow 1$ to $N$ do
        $P \leftarrow$ compute Probability using Equation 7.13;
        $P^* \leftarrow 0$;
        for $k \leftarrow 1$ to 3 do /* For each transformation parameter */
            while $|P^* - P| > 0$ do
                $P^* \leftarrow P$;
                $t_k^{i*} \leftarrow t_k^i + \delta(k)$;  /* Try new value for transform.  */
                $I_i^* \leftarrow I_i + t_k^{i*}$;  /* Compute I from new transform.  */
                $P_{new} \leftarrow$ compute Probability using Equation 7.13;
                if $P_{new} > P$ then
                    $P \leftarrow P_{new}$;
                    $t_k^i \leftarrow t_k^{i*}$;
                    $I_i \leftarrow I_i^*$;
                end
            end
        end
    end
end
```

Algorithm 4: Unsupervised alignment algorithm.

7.4 Tracking

In order to achieve reliable human tracking, the real-time multi-target tracking system presented in Chapter 5 is extended for fisheye tracking. Therefore, the combined head detector that was presented in the last section is used to generate detections. These detections are then used to form a set of target hypotheses, which are then used to infer the target trajectories in the same way as it was presented in Chapter 5.

In order to estimate the location of pedestrians in the current frame and to ensure that data associations can be made correctly, a tracker is initialised to track the relative motion for a period. Due to the significant change between different locations, the tracker used in Chapter 5 can not be used and the Lucas-Kanade optical-flow based approach, that was proposed in Chapter 3, is used instead.

7.5 Experimental Setup and Evaluation

In this section the performance of the tracking framework is evaluated on two benchmarks: A novel benchmark called the Living Place Fisheye Dataset (LPFD) and another benchmark called Bomni-DB [Demiroz et al., 2012].

7.5.1 Living Place Fisheye Dataset (LPFD)

The dataset includes videos captured by three fisheye cameras. The cameras are mounted on different locations on the ceiling of the LPH [Place, 2012]. Videos have been acquired using Mobtix Q24 cameras, which are 360 degrees IP cameras. Camera calibration parameters were computed with the Omnidirectional Camera Calibration Toolbox[2]. The videos were captured using 2048×1536 pixels resolution and a frame rate of 8 fps. The dataset is divided into two separate sets. The first set is used to train the detector and contains 500 images from two videos with

[2] https://sites.google.com/site/scarabotix/ocamcalib-toolbox

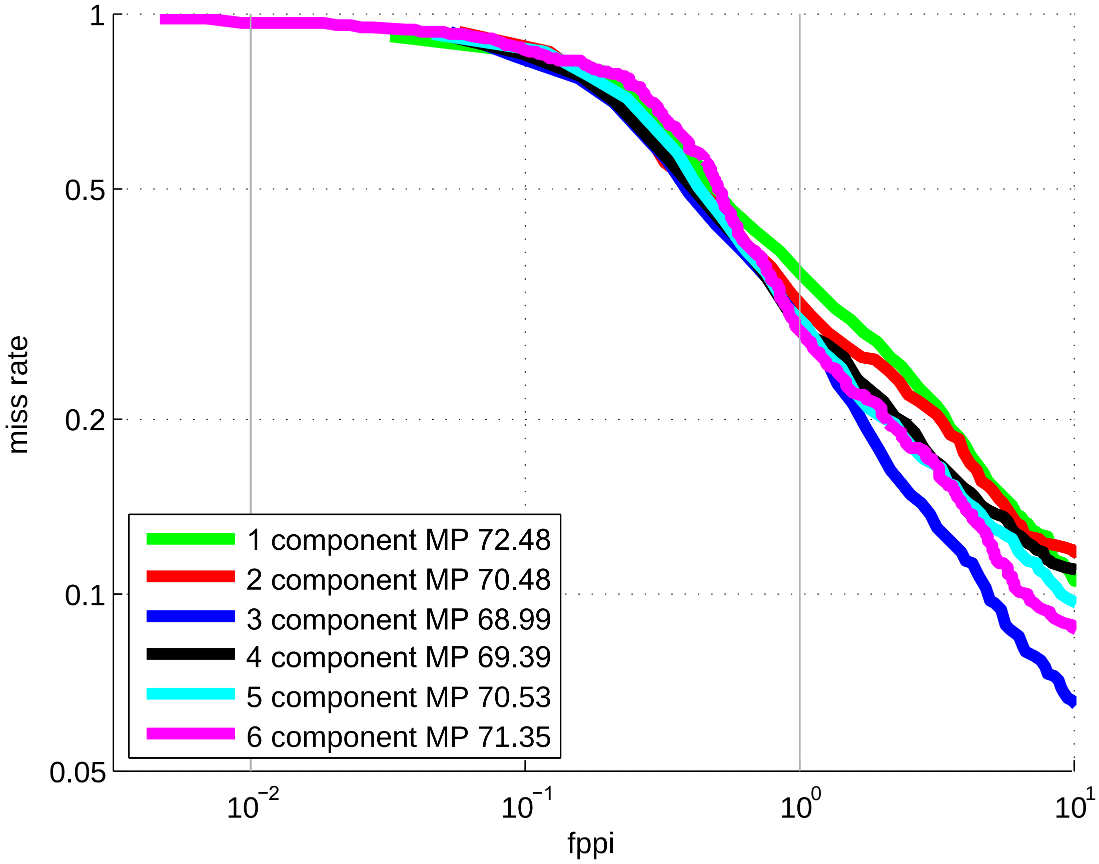


FIGURE 7.10: Fisheye head detector results with different number of components.

3039 annotated heads from approximately 200 different subjects. The second set contains 10 videos. The videos have ground truth consisting of 19000 hand labelled head locations. The challenges of the dataset are low frame rate, frame drops, occlusions and noise.

The number of components used to create the detector has impact on the accuracy of the detector. In order to find the optimal number of components, detectors with different number of components were compared. Figure 7.10 shows the results. It can be seen that the MR does not drop any further past three components. It can be assumed that the reason for this is that the cluster become to small (three components cluster size: (562/817/1660), four components size: (259/569/856/1355)).

Finally, the quality of the proposed multi-target tracking algorithm is evaluated using CLEAR MOT metrics as described in Chapter 5. Since there are no similar

Table 7.1: Living Place Fisheye Dataset results.

Method	MOTA	MOTP	Prec.	Rec.	FP	Missed
Head detections	47.32%	52.74%	74.21%	73.02%	23.71%	26.83%
Proposed schema	69.53%	59.41%	82.57%	79.31%	12.89%	15.02%

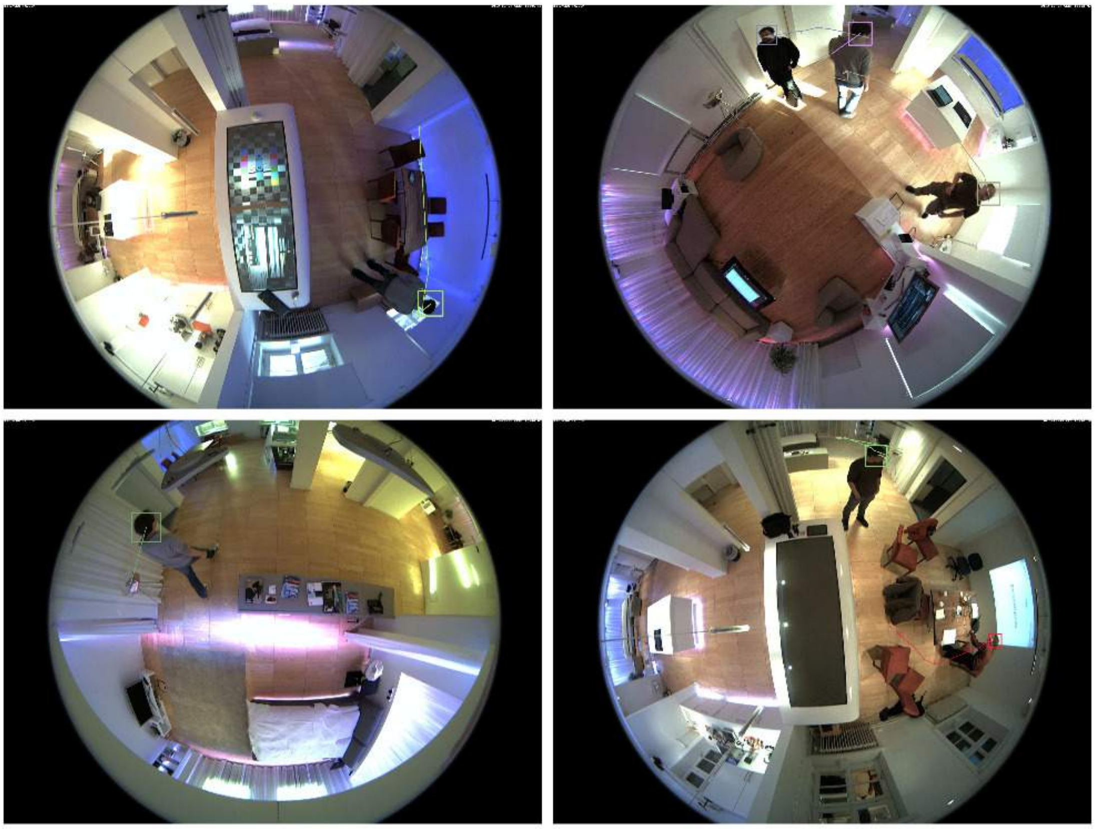


Figure 7.11: Tracking results on the Living Place Fisheye Dataset

head tracking results to compare with, baseline results from raw head and detections are included for comparison. The results are shown in Table 7.1. In addition, some tracking results are presented in Figure 7.11.

7.5.2 Bomni-DB Database

Bomni-DB [Demiroz et al., 2012] database consists of 23 videos recorded in a room with two cameras. The first camera is mounted on the ceiling of the room and the latter is fixed on a side wall. Since the proposed algorithm was developed for top down views only the recordings of the first camera were used. The videos were

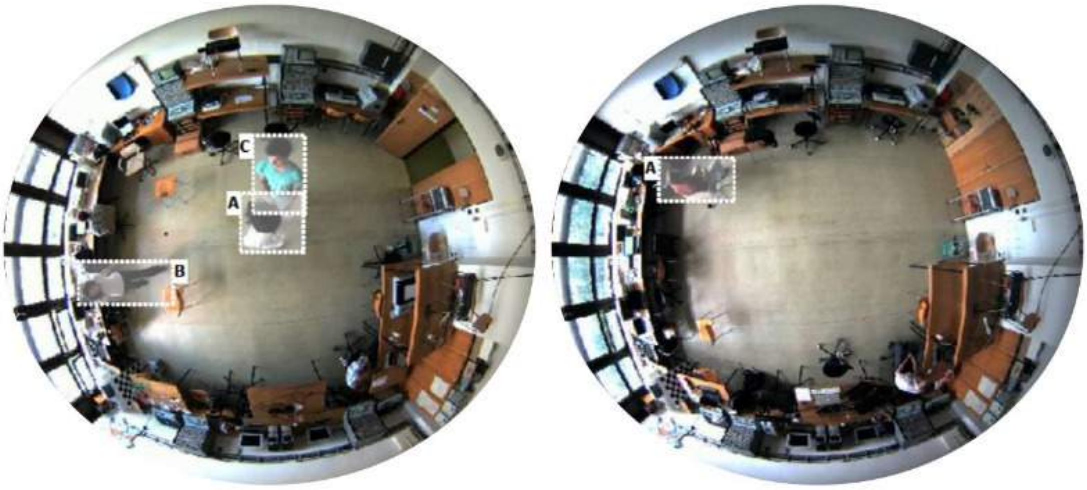


FIGURE 7.12: Sample images from the Bomni-DB (taken from [Demiroz et al., 2012]).

captured using a Oncam IPC fisheye camera with a resolution of 640 × 480 pixels and a frame rate of 8 fps. The dataset provides annotated full body locations Figure 7.12 shows some samples.

The proposed algorithm has as output head locations while the dataset used body locations. Due to the large view changes between different locations in an images it is not possible to estimate the body location as was done in Chapter 4. In order to still be able to evaluate the proposed algorithm on the dataset, the body location for each head detection is estimated using background subtraction.

Therefore, the foreground is computed using the OpenCV background subtractor `cv::BackgroundSubtractorMOG`, then for each head detection the possible area in which the body could be is estimated. In this area are then the foreground contours computed using the OpenCV function `cv::findContours`. Finally the enclosing rectangle is computed for this contours using the OpenCV function `cv::boundingRect`. Figure 7.13 illustrates this computation.

Table 7.2 summarizes the results on the Bomni-DB. The proposed system outperform the tracking algorithm of [Demiroz et al., 2012] by 5.03% (MOTA). The missed detections are reduces by 6.05% whereas the false positives increase slightly by 0.26%.

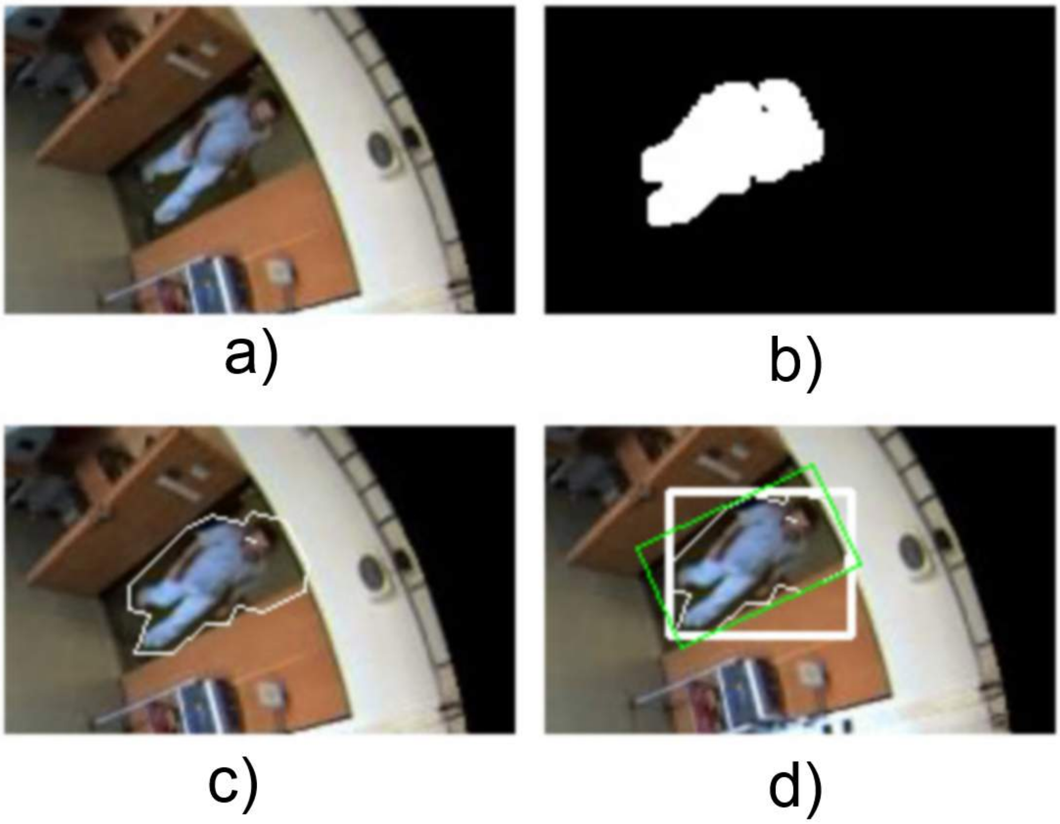


Figure 7.13: Body locations compuation: a) input image, b) foreground, c) contours, d) rotated rectangle (green) and minimum bounding rectangle (white).

Table 7.2: Bomni-DB results.

Method	MOTA	MOTP	False Pos.	Missed
Top view [Demiroz et al., 2012]	73.52%	72.00%	7.78%	18.78%
Proposed schema	78.55%	76.74%	8.04%	12.73%

7.6 Conclusion

This chapter described and demonstrated a real-time system for multi-target tracking based on fisheye cameras. It has shown that the use of standard detection algorithms is possible if the fisheye image is projected on a set of perspective images.

In order to train a detector based on annotated heads in fisheye images, a novel algorithm was proposed that automatically generates sets of aligned heads in perspective images that are then used to train a combined multi-view detector.

The proposed detector was then integrated in a real-time system for multi-target tracking. In order to evaluate the algorithm, a new dataset for indoor multi-target on fisheye cameras was created, on which a MOTA of 69.53% was achieved. In addition, evolution was performed on the Bomni-DB, where a MOTA of 78.55% was achieved, which is a 5.03% higher MOTA compared to [Demiroz et al., 2012].

Chapter 8

Conclusions and Future work

8.1 Introduction

This thesis has described a complete framework for detecting, tracking and identifying people in smart homes. This chapter summarises the main findings and results. In Section 8.2 it is investigated if the objectives specified in Chapter 1 have been achieved. Section 8.3 discusses a number of restrictions and limitations that have been identified during this research, while suggestions for future investigation are presented in Section 8.4.

8.2 Objectives Achieved

The objectives of this thesis are listed in Chapter 1. In this section, each of these objectives is revisited in order to determine the level of achievement.

- **To design new frameworks for more accurate identification of individuals in video face recognition.**

  This thesis has described a complete framework for face tracking and recognition in Chapter 3. A novel algorithm for building face tracks in real-life scenarios has been proposed that combines face detection and optical flow tracking to a face tracking algorithm. In order to recognise the identity of each track, the LBPFHist algorithm has been proposed, which recognises each face image in a track with Local Binary Patterns (LBP) face recognition and then calculates the maximum occurring identity over the track in order to improve the recognition rate by exploiting temporal information.

The evaluation has been performed using the Honda/UCSD Video Database, where 100% recognition rate has been achieved and a new Smart Home dataset where a recognition rate of 91% was achieved. The LBPFHist algorithm has outperformed a similar algorithm called LBPF, which does not use temporal information by 23% on the Honda/UCSD Video Database and by 25% on the Smart Home dataset. Results have shown that the proposed approach operates in real-time and is able to handle illumination changes, occlusions and out-of-plane rotations.

In addition, in Chapter 6 a second framework for video face recognition were proposed, which includes face detection, face tracking, face alignment and video face recognition. This second framework is based on multi-target tracking, where a novel face detector is combined with a head detector in order to create detections that are then used to build tracks. Two novel set-to-set similarity measures have been proposed that determine whether faces appearing in the two tracks are of the same subject.

The algorithms have been evaluated on the YouTube Face Database (83.44% accuracy), YouTube Celebrities Dataset (84.07% accuracy) and the ChokePoint Dataset (100% accuracy). The proposed algorithms have achieved the best result on the YouTube Celebrities Dataset and the ChokePoint Dataset and the best result on the YouTube Face Database compared to all algorithms that have used the provided aligned images. Results showed that the proposed algorithm can be used for identification and tracking of people in smart homes.

- **To investigate more reliable detection of individual people from overhead surveillance cameras under different viewpoints.**

Two approaches have been presented for detecting people, focusing on head detection. The first approach was presented in Chapter 4 and addresses the problem of detecting heads in crowded real-world scenes by combining a human head detector, an upper-body detector and a body detector to create a robust head detector. The idea has been relied not only on a single detector for decision making but to combine individual opinions of multiple detectors to derive a consensus decision.

In order to validate the findings, the performance of the detection system has been tested on the Town Centre Benchmark. The experiments performed have shown that the proposed combined head detector reduces the MR by 18% compared to the single head detector.

The second approach was presented in Chapter 7 and extends the first approach for fisheye cameras. The main novelty of that approach is that it uses sub-categories detectors in order to be robust to a wide range of views in fisheye images. A novel unsupervised clustering algorithm has been introduced that clusters head images in sub-categorises while simultaneously aligning them. These sub-categorises have been then used to train classifiers for different views. The detector has achieved a MR of 68.99 % on the Living Place Fisheye Dataset.

- **To devise more robust real-time systems for tracking multiple individuals through camera views, where inter-person and object occlusion may be present.**

In Chapter 5 a real-time multi-target tracking system for a single view was presented. To achieve real-time performance, a multi-threaded approach has been used in which one thread produces asynchronous head detections, while a second thread applies a tracking algorithm to estimate the location of pedestrians and a third thread performs data association. In order to reduce the number of false positives and simultaneously reduce the number of missed detections, a novel motion model has been presented that includes the identification of false positives in the data association and treats false positives on foreground and background objects separately.

The results have shown that this approach is superior to typical existing approaches. For the Town Centre Benchmark, the system outperforms the head tracking algorithm of [Benfold and Reid, 2011] by 36% (MOTA), while the number of false positives decreases from 18374 to 6500 and the number of missed detections decreases from 20427 to 6127.

A second approach was proposed in Chapter 7, which extends the first approach for fisheye cameras. Firstly a camera model has been described that allows fast projection between a fisheye image and a corresponding set of perspective images. Then an algorithm has been proposed that automatically generates from annotated heads in fisheye images, sets of aligned heads in perspective images that are then used to train a combined multi-view detector. These detections are then used to form a set of target hypotheses, which are then used to infer the target trajectories using the real-time multi-target tracking system from Chapter 5.

The experimental results have shown that the proposed framework is able to track humans in the context of a fisheye camera. The evaluation on the Bomni-DB has shown the proposed system outperform the tracking algorithm of [Demiroz et al., 2012] by 5.03% (MOTA).

- **To implement and empirically evaluate the performance of the proposed approaches using different datasets.**

The performance of the proposed approaches has been evaluated on several datasets. Face recognition and face tracking have been tested on Honda/UCSD Video Database (100% accuracy), Smart Home Dataset (83.44% accuracy), YouTube Face Database (83.44% accuracy), YouTube Celebrities Dataset (84.07% accuracy) and the ChokePoint Dataset (100% accuracy). The proposed algorithms have achieved the best results on all datasets, except for the YouTube Face Database. The experiments performed in Chapter 3 have shown that temporal information provided by face tracks have improved the recognition rate by 23% on the Honda/UCSD dataset and by 25% on the Smart Home dataset compared to an algorithm that does not use temporal information (LBPF).

Face detection has been evaluated on the FDDB dataset and the AFW dataset (AP 95.32%) where state-of-art results have been achieved. In addition, the computing speed has been evaluated. The proposed face detector that combines a general classifier and view-specific classifier has been on average twice as fast as a comparable detector that only uses view-specific classifiers.

Head detection has been evaluated on the Town Centre Benchmark where the combined detector of a head, an upper-body and a body detector has reached an MR of 58%. This is a decrease of the MR of 18% compared to the head detector only.

Finally, multi-target tracking has been evaluated on the Town Centre Benchmark (MOTA 81.55%), Parking Lot Benchmark (MOTA 79.71%), Bomni-DB (MOTA 78.55%) and LivingPlace Fisheye Benchmark (MOTA 69.53 %). Results have shown that the proposed approach improves the tracking performance compared to similar systems, while simultaneously running in real-time.

8.3 Limitations

The proposed framework has a number of restrictions and limitations that have been identified during this research. Some of these are due to the shortcomings of the used hardware; some are related to extensions that are worth investigating in the near future whilst the others are open issues for longer term research. This section presents the existing limitations and the next section discusses open issues and offers suggestions for future work that can be built upon the ideas and concepts presented in this thesis.

8.3.1 Setting

The LPH is a loft style apartment, which consists of one large room with different sections for dining, living, cooking, sleeping and working and a separated bathroom. This setting is not representative for most households, where several challenges maybe occur, such as tracking across multiple rooms or floors, which have not been addressed in this work. In addition, the proposed approach assumes entry and exit points which may not be present in every scenario.

8.3.2 Lighting Conditions

The research carried out in this thesis has largely assumed good lighting conditions. However, for a system to be deployed in a real-world scenario, enhancements for low-light conditions may be required. This would particularly be the case if the system has to operate in the dark or in an outdoor environment. In such scenarios night vision hardware is required.

8.4 Future Directions

Despite the advances introduced in this work, there remain some limitations and topics unaddressed in this thesis. This section provides a discussion on future research directions.

8.4.1 Generalized Similarity Metric Learning

In [Xu et al., 2012] metric learning for SVMs was investigated. An empirical study on Mahalanobis metric learning showed that metric learning explicitly for the SVM decision rule outperforms metric learning as a pre-processing step as done in Chapter 6. In the future it would be interesting to investigate if the same is true for Generalized Similarity Metric Learning and how the SVM classification can be combined with Generalized Similarity Metric Learning.

8.4.2 Position of a Person in the Ground Plane

The tracking algorithms presented in this thesis have focused on tracking on the image plane. This approach allows the focus of experiments to remain on tracking performance and simplifies the validation of techniques. However, for real-world scenarios it is more likely that the position on the ground plane is needed. In the future it would be interesting to examine capabilities to find the position of a person in the ground plane using a single camera. If a person is fully visible, the contact zone between a person and the ground plane can be used to compute a plane-to-plane homography between the image plane and the ground plane.

However, in most scenarios the contact zone is not always visible. An interesting approach to solve this problem was proposed by Rougier and Meunier [Rougier and Meunier, 2010]. Their method extracts the 3D head track of a person in a room using only a single calibrated camera by representing the head as a projection of a 3D ellipsoid model. However, their approach relies on background subtraction. It would be interesting to examine how this approach can be extended to tracking-by-detection approaches.

8.4.3 Tracking in More Complicated Scenarios

Multi-target tracking remains a significant challenge, especially in fisheye images. The multi-target tracking algorithms introduced in this thesis have been limited to humans. For more general situations, this could be extended to represent a number of different object types such as animals, cars or bicycles. For instance, in many households in addition to humans, pets could be tracked independently.

Face recognition and tracking have been treated as two separate tasks. However in practice the recognition task depends on the tracked person. In the future, the identification of humans could be combined with the data association.

Another interesting topic, which is related to identification, is to re-identify humans that leave from an area covered by one camera and enter to a different area, or re-enter the same place after a period of time. This problem is often difficult since an object could have a number of potential matches and it may not always be possible to disambiguate all the matches [Gandhi and Trivedi, 2007]. To solve this problem is of particular interest for smart environments in which the same person is tracked frequently.

Another topic for future work is to investigate human behaviour in order to improve tracking. For example Yang and Nevatia [Yang and Nevatia, 2012] established a non-linear motion map from previous human paths in order to estimate which path other humans are likely to take in the future. Other authors [Qin and Shelton, 2012] investigated how humans move in small groups. Such behaviour models can then be used to improve tracking in difficult situations.

8.4.4 Sensors-Assisted Identification

Several sensors beyond cameras are used in LPH in order to analyse the relations between inhabitants and smart home. For example textile sensors have been integrated into the sofa in order to detect where a person sits, while other sensors have been integrated into the floor for localizing humans. It would be interesting to explore the fusion of localization and identification results from two or more sensors and to investigate if this produces more reliable results.

8.4.5 Performance

In the further it would be interesting to explore performance improvements if the proposed systems would be implemented on GPUs or FPGAs.

8.4.6 Real-Life Applications

The research carried out in this thesis has been evaluated on several test databases that emulate real-life applications in smart environments. It is noted that the use of test databases is a common simplification in order to create reproducible results and be able to compare different approaches. However, it would be desirable if the performance of the proposed framework could be more thoroughly and extensively investigated in future in real-life environments and over longer periods of time. In addition, the interaction between the proposed frameworks and other research projects in the LPH such as facial expression recognition, emotion recognition and inhabitant and smart home interaction should be investigated.

Bibliography

[Adam et al., 2006] Adam, A., Rivlin, E., and Shimshoni, I. (2006). Robust Fragments-based Tracking using the Integral Histogram. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 798–805. IEEE Computer Society.

[Ahonen et al., 2004] Ahonen, T., Hadid, A., and Pietikainen, M. (2004). Face Recognition with Local Binary Patterns. *European Conference on Computer Vision (ECCV)*, pages 469–481.

[Asthana et al., 2014a] Asthana, A., Zafeiriou, S., Cheng, S., and Pantic, M. (2014a). Incremental Face Alignment in the Wild. In *International Conference on Computer Vision & Pattern Recognition (CVPR 2014)*.

[Asthana et al., 2014b] Asthana, A., Zafeiriou, S., Cheng, S., and Pantic, M. (2014b). Incremental face alignment in the wild. https://sites.google.com/site/chehrahome/home. Accessed: 21.02.2015.

[Augusto, 2009] Augusto, J. C. (2009). Past, Present and Future of Ambient Intelligence and Smart Environments. In *International Conference on Agents and Artificial Intelligence (ICAART)*, pages 11–18. INSTICC Press.

[Avidan, 2004] Avidan, S. (2004). Support Vector Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072.

[Avidan, 2007] Avidan, S. (2007). Ensemble Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271.

[AVSS, 2007] AVSS (2007). i-Lids dataset for AVSS 2007. http://www.eecs.qmul.ac.uk/ andrea/avss2007_d.html. Accessed: 21.02.2015.

[Aztiria et al., 2012] Aztiria, A., Augusto, J. C., Basagoiti, R., Izaguirre, A., and Cook, D. J. (2012). Discovering frequent user-environment interactions in intelligent environments. *Personal and Ubiquitous Computing*, 16(1):91–103.

[Bai and Tang, 2012] Bai, Y. and Tang, M. (2012). Robust tracking via weakly supervised ranking SVM. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1854–1861. IEEE.

[Ban et al., 2014] Ban, Y., Kim, S., Kim, S., Toh, K.-A., and Lee, S. (2014). Face detection based on skin color likelihood. *Pattern Recognition*, 47(4):1573–1585.

[Belhumeur et al., 2011] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 545–552. IEEE.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

[Ben-Ari and Ben-Shahar, 2013] Ben-Ari, R. and Ben-Shahar, O. (2013). A computationally efficient tracker with direct appearance-kinematic measure and adaptive Kalman filter. *Journal of Real-Time Image Processing*.

[Benenson et al., 2012] Benenson, R., Mathias, M., Timofte, R., and Gool, L. J. V. (2012). Pedestrian detection at 100 frames per second. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2903–2910. IEEE.

[Benenson et al., 2011] Benenson, R., Timofte, R., and Gool, L. J. V. (2011). Stixels estimation without depth map computation. In *International Conference on Computer Vision (ICCV)*, pages 2010–2017. IEEE.

[Benfold and Reid, 2011] Benfold, B. and Reid, I. (2011). Stable Multi-Target Tracking in Real-Time Surveillance Video. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3457–3464.

[Berclaz et al., 2011] Berclaz, J., Fleuret, F., Tueretken, E., and Fua, P. (2011). Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819.

[Bernardin and Stiefelhagen, 2008] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *Journal of Image Video Process.*, 2008:1:1–1:10.

[Black et al., 2002] Black, J., Ellis, T., and Rosin, P. (2002). Multi View Image Surveillance and Tracking. In *Motion and Video Computing Workshop*.

[Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A Morphable Model for the Synthesis of 3D Faces. In *Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, pages 187–194.

[Bolme et al., 2010] Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2550. IEEE.

[Bradski, 1998] Bradski, G. R. (1998). Real Time Face and Object Tracking As a Component of a Perceptual User Interface. In *IEEE Workshop on Applications of Computer Vision (WACV)*, Washington, DC, USA. IEEE Computer Society.

[Breitenstein et al., 2009] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Gool, L. J. V. (2009). Robust tracking-by-detection using a detector confidence particle filter. In *International Conference on Computer Vision (ICCV)*, pages 1515–1522. IEEE.

[Breitenstein et al., 2011a] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Gool, L. J. V. (2011a). Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833.

[Breitenstein et al., 2011b] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. (2011b). Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833.

[Brendel et al., 2011] Brendel, W., Amer, M. R., and Todorovic, S. (2011). Multiobject tracking as maximum weight independent set. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1273–1280. IEEE.

[Burges, 1998] Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[Cao et al., 2013] Cao, Q., Ying, Y., and Li, P. (2013). Similarity Metric Learning for Face Recognition. In *International Conference on Computer Vision (ICCV)*.

[Cao et al., 2012] Cao, X., Wei, Y., Wen, F., and Sun, J. (2012). Face alignment by Explicit Shape Regression. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2887–2894. IEEE.

[Cevikalp and Triggs, 2010] Cevikalp, H. and Triggs, B. (2010). Face recognition based on image sets. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2567–2573. IEEE.

[Cevikalp et al., 2013] Cevikalp, H., Triggs, B., and Franc, V. (2013). Face and landmark detection by using cascade of classifiers. In *International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7. IEEE.

[Chan et al., 2013] Chan, K.-C., Koh, C.-K., and Lee, C. (2013). Collaborative object tracking with motion similarity measure. In *International Conference on Robotics and Biomimetics (ROBIO)*, pages 964–969.

[Collins, 2003] Collins, R. T. (2003). Mean-shift Blob Tracking through Scale Space. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 234–240. IEEE Computer Society.

[Comaniciu et al., 2003] Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575.

[Cootes et al., 2001] Cootes, T., Edwards, G., and Taylor, C. (2001). Active appearance models. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 23(6):681–685.

[Cox et al., 2008] Cox, M., Sridharan, S., Lucey, S., and Cohn, J. F. (2008). Least squares congealing for unsupervised alignment of images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society.

[Cristinacce and Cootes, 2008] Cristinacce, D. and Cootes, T. (2008). Automatic feature localisation with constrained local models . *Pattern Recognition*, 41(10):3054 – 3067.

[Cui et al., 2013] Cui, Z., Li, W., Xu, D., Shan, S., and Chen, X. (2013). Fusing Robust Face Region Descriptors via Multiple Metric Learning for Face Recognition in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3554–3561. IEEE.

[Cui et al., 2012] Cui, Z., Shan, S., Zhang, H., Lao, S., and Chen, X. (2012). Image sets alignment for Video-Based Face Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2633. IEEE.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, Washington, DC, USA. IEEE Computer Society.

[Dalal et al., 2006] Dalal, N., Triggs, B., and Schmid, C. (2006). Human Detection Using Oriented Histograms of Flow and Appearance. In *European conference on Computer Vision (ECCV)*, volume 3952 of *Lecture Notes in Computer Science*, pages 428–441. Springer.

[Davis et al., 2007] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *International Conference on Machine Learning (ICML)*, volume 227 of *ACM International Conference Proceeding Series*, pages 209–216. ACM.

[Demiroz et al., 2012] Demiroz, B., Ari, I., Eroglu, O., Salah, A., and Akarun, L. (2012). Feature-based tracking on a multi-omnidirectional camera dataset. In *International Symposium on Communications Control and Signal Processing (ISCCSP)*, pages 1–5.

[Deng et al., 2005] Deng, W., Hu, J., and Guo, J. (2005). Gabor-Eigen-Whiten-Cosine: a robust scheme for face recognition. In *International Conference on Analysis and Modelling of Faces and Gestures.*

[Dinh et al., 2011] Dinh, T. B., Vo, N., and Medioni, G. (2011). Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1177–1184, Washington, DC, USA. IEEE Computer Society.

[Dollár et al., 2014] Dollár, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).*

[Dollár et al., 2010] Dollár, P., Belongie, S., and Perona, P. (2010). The Fastest Pedestrian Detector in the West. In *British Machine Vision Conference (BMVC).* British Machine Vision Association.

[Dollár et al., 2009] Dollár, P., Tu, Z., Perona, P., and Belongie, S. (2009). Integral Channel Features. In *British Machine Vision Conference (BMVC).* British Machine Vision Association.

[Dollár et al., 2012] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34.

[Dowson and Bowden, 2005] Dowson, N. D. H. and Bowden, R. (2005). Simultaneous Modeling and Tracking (SMAT) of Feature Sets. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–105. IEEE Computer Society.

[Everingham et al., 2006] Everingham, M., Sivic, J., and Zisserman, A. (2006). Hello! My name is... Buffy“ Automatic naming of characters in TV video. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338.

[Felzenszwalb et al., 2010a] Felzenszwalb, P. F., Girshick, R. B., and McAllester, D. A. (2010a). Cascade object detection with deformable part models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2241–2248. IEEE.

[Felzenszwalb et al., 2010b] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., and Ramanan, D. (2010b). Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645.

[Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial Structures for Object Recognition. *International Journal Computer Vision*, 61(1):55–79.

[Fortmann et al., 1983] Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *Journal of Oceanic Engineering*, 8(3):173–184.

[Frey and Jojic, 1999] Frey, B. J. and Jojic, N. (1999). Transformed Component Analysis: Joint Estimation of Spatial Transformations and Image Components. In *International Conference on Computer Vision (ICCV)*, pages 1190–1196.

[Fusco et al., 2013] Fusco, G., Zini, L., Noceti, N., and Odone, F. (2013). Structured Multi-class Feature Selection for Effective Face Recognition. In *International Conference on Image Analysis and Processing (ICIAP)*, volume 8156 of *Lecture Notes in Computer Science*, pages 410–419. Springer.

[Gallery, 2013] Gallery, P. K. (2013). James Nares: "STREET". https://vimeo.com/47457051. Accessed: 21.02.2015.

[Gandhi and Trivedi, 2007] Gandhi, T. and Trivedi, M. M. (2007). Person tracking and reidentification: Introducing Panoramic Appearance Map (PAM) for feature representation. *Mach. Vis. Appl.*, 18(3-4):207–220.

[Ge et al., 2011] Ge, K.-B., Wen, J., and Fang, B. (2011). Adaboost algorithm based on MB-LBP features with skin color segmentation for face detection. In *International Conference onWavelet Analysis and Pattern Recognition (ICWAPR)*, pages 40–43.

[Ge and Collins, 2008] Ge, W. and Collins, R. (2008). Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. *British Machine Vision Conference (BMVC).*

[Giebel et al., 2004] Giebel, J., Gavrila, D., and Schnoerr, C. (2004). A Bayesian Framework for Multi-cue 3D Object Tracking. In *European conference on Computer Vision (ECCV)*, volume 3024 of *Lecture Notes in Computer Science*, pages 241–252. Springer.

[Goldberger et al., 2004] Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. (2004). Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press.

[Grabner et al., 2006] Grabner, H., Grabner, M., and Bischof, H. (2006). Real-Time Tracking via On-line Boosting. In *British Machine Vision Conference (BMVC)*, pages 47–56. British Machine Vision Association.

[Grabner et al., 2010] Grabner, H., Matas, J., Gool, L. J. V., and Cattin, P. C. (2010). Tracking the invisible: Learning where the object might be. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1285–1292. IEEE.

[Greenspan et al., 2001] Greenspan, H., Goldberger, J., and Eshet, I. (2001). Mixture model for face-color modeling and segmentation. *Pattern Recognition Letters*, 22(14):1525–1536.

[Guillaumin et al., 2009] Guillaumin, M., Verbeek, J., and Schmid, C. (2009). Is that you? Metric learning approaches for face identification. In *International Conference on Computer Vision (ICCV)*, Kyoto, Japan.

[Hadid and Pietikaeinen, 2009] Hadid, A. and Pietikaeinen, M. (2009). Combining appearance and motion for face and gender recognition from videos. *Pattern Recognition*, 42(11):2818–2827.

[Hare et al., 2011] Hare, S., Saffari, A., and Torr, P. H. S. (2011). Struck: Structured output tracking with kernels. In *International Conference on Computer Vision (ICCV)*, pages 263–270. IEEE.

[He et al., 2009] He, W., Yamashita, T., Lu, H., and Lao, S. (2009). SURF Tracking. In *International Conference on Computer Vision (ICCV)*, pages 1586–1592. IEEE.

[Heisele et al., 2001] Heisele, B., Serre, T., Pontil, M., and Poggio, T. (2001). Component-based Face Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 657–662. IEEE Computer Society.

[Henriques et al., 2012] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2012). Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *European conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 702–715. Springer.

[Hoai and Zisserman, 2013] Hoai, M. and Zisserman, A. (2013). Discriminative Sub-categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1666–1673. IEEE.

[Hu et al., 2014] Hu, J., Lu, J., and Tan, Y.-P. (2014). Discriminative Deep Metric Learning for Face Verification in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Hu et al., 2011] Hu, Y., Mian, A. S., and Owens, R. A. (2011). Sparse approximated nearest points for image set classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 121–128. IEEE.

[Huang et al., 2008] Huang, C., Wu, B., and Nevatia, R. (2008). Robust Object Tracking by Hierarchical Association of Detection Responses. In *European conference on Computer Vision (ECCV)*, volume 5303 of *Lecture Notes in Computer Science*, pages 788–801. Springer.

[Huang et al., 2007] Huang, G. B., Jain, V., and Learned-Miller, E. G. (2007). Unsupervised Joint Alignment of Complex Images. In *International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE.

[Huang et al., 2012a] Huang, G. B., Lee, H., and Learned-Miller, E. G. (2012a). Learning hierarchical representations for face verification with convolutional deep belief networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2518–2525. IEEE.

[Huang et al., 2012b] Huang, G. B., Mattar, M. A., Lee, H., and Learned-Miller, E. G. (2012b). Learning to Align from Scratch. In *Neural Information Processing Systems Foundation (NIPS)*, pages 773–781.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28.

[Jain and Learned-Miller, 2010] Jain, V. and Learned-Miller, E. (2010). Face Detection Data Set and Benchmark. http://vis-www.cs.umass.edu/fddb/results.html. Accessed: 21.02.2015.

[Jepson et al., 2003] Jepson, A. D., Fleet, D. J., and El-Maraghi, T. F. (2003). Robust Online Appearance Models for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311.

[Jin and Ruan, 2009] Jin, Y. and Ruan, Q. (2009). Face Recognition Using Gabor-based Improved Supervised Locality Preserving Projections. *Computing and Informatics*, 28(1):81–95.

[Juan Carlos Augusto, 2006] Juan Carlos Augusto, C. D. N., editor (2006). *Designing Smart Homes, The Role of Artificial Intelligence.*, volume 4008 of *Lecture Notes in Computer Science*. Springer.

[Kalal et al., 2011] Kalal, Z., Matas, J., and Mikolajczyk, K. (2011). Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Kalal et al., 2010] Kalal, Z., Mikolajczyk, K., and Matas, J. (2010). Forward-Backward Error: Automatic Detection of Tracking Failures. In *International Conference on Computer Vision (ICCV)*, pages 2756–2759. IEEE.

[Kaucic et al., 2005] Kaucic, R., Perera, A. G. A., Brooksby, G., Kaufhold, J. P., and Hoogs, A. (2005). A Unified Framework for Tracking through Occlusions and across Sensor Gaps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 990–997. IEEE Computer Society.

[Kim et al., 2008] Kim, M., Kumar, S., Pavlovic, V., and Rowley, H. A. (2008). Face tracking and recognition with visual constraints in real-world videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society.

[Kim and Cipolla, 2008] Kim, T.-K. and Cipolla, R. (2008). MCBoost: Multiple Classifier Boosting for Perceptual Co-clustering of Images and Visual Features.

In *Neural Information Processing Systems Foundation (NIPS)*, pages 841–856. Curran Associates, Inc.

[Koestinger et al., 2012] Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2012). Large scale metric learning from equivalence constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2295. IEEE.

[Koestinger et al., 2011] Koestinger, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2011). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies.*

[Kubo et al., 2007] Kubo, Y., Kitaguchi, T., and Yamaguchi, J. (2007). Human tracking using fisheye images. *SICE Annual Conference 2007*, pages 2013–2017.

[Kuhn, 1955] Kuhn, H. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

[Kwon and Lee, 2011] Kwon, J. and Lee, K. M. (2011). Tracking by Sampling Trackers. In *International Conference on Computer Vision (ICCV)*, pages 1195–1202. IEEE.

[la Torre and Black, 2003] la Torre, F. D. and Black, M. J. (2003). Robust parameterized component analysis: theory and applications to 2D facial appearance models. *Computer Vision and Image Understanding*, 91(1-2):53–71.

[Learned-Miller, 2006] Learned-Miller, E. G. (2006). Data Driven Image Models through Continuous Joint Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250.

[Lee et al., 2003] Lee, K., Ho, J., Yang, M., and Kriegman, D. (2003). Video-Based Face Recognition Using Probabilistic Appearance Manifolds. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:313–320.

[Lee et al., 2005] Lee, K., Ho, J., Yang, M., and Kriegman, D. (2005). Visual Tracking and Recognition Using Probabilistic Appearance Manifolds. *Computer Vision and Image Understanding*.

[Leibe et al., 2005] Leibe, B., Seemann, E., and Schiele, B. (2005). Pedestrian Detection in Crowded Scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 878–885. IEEE Computer Society.

[Lewis, 1995] Lewis, J. P. (1995). Fast Normalized Cross-Correlation. *Vision Interface*, pages 120–123.

[Li et al., 2013a] Li, H., Hua, G., Lin, Z., Brandt, J., Yang, J., and Jose, S. (2013a). Probabilistic Elastic Matching for Pose Variant Face Verification. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Li et al., 2013b] Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., and Hengel, A. V. D. (2013b). A Survey of Appearance Models in Visual Object Tracking. *ACM Transactions on Intelligent Systems and Technology*, 4(4):58:1–58:48.

[Lienhart and Maydt, 2002] Lienhart, R. and Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *International Conference on Image Processing (ICIP)*, pages 900–903.

[Liu et al., 2007a] Liu, J., Tong, X., Li, W., Wang, T., 0001, Y. Z., Wang, H., 0008, B. Y., Sun, L., and Yang, S. (2007a). Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video. In *British Machine Vision Conference (BMVC)*, pages 1–10. British Machine Vision Association.

[Liu et al., 2007b] Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H., Yang, B., Sun, L., and Yang, S. (2007b). Automatic player detection, labeling and tracking in broadcast soccer video . *British Machine Vision Conference (BMVC)*.

[Liu et al., 2010] Liu, Y., Shan, S., Chen, X., Heikkila, J., Gao, W., and Pietikainen, M. (2010). Spatial-temporal Granularity-tunable Gradients Partition

(STGGP) Descriptors for Human Detection. In *European Conference on Computer Vision (ECCV)*, pages 327–340, Berlin, Heidelberg. Springer-Verlag.

[Liu et al., 2009] Liu, Y., Shan, S., Zhang, W., Chen, X., and Gao, W. (2009). Granularity-tunable gradients partition (GGP) descriptors for human detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1255–1262. IEEE.

[Lowe, 2003] Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20.

[Lowe, 1999] Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision (ICCV)*.

[Lucas and Kanade, 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679.

[Marin-Jimenez et al., 2014] Marin-Jimenez, M. J., Zisserman, A., Eichner, M., and Ferrari, V. (2014). Detecting People Looking at Each Other in Videos. *International Journal of Computer Vision*, 106(3):282–296.

[Mathias et al., 2014] Mathias, M., Benenson, R., Pedersoli, M., and Van Gool, L. (2014). Face detection without bells and whistles. In *European conference on Computer Vision (ECCV)*.

[Matthews et al., 2004] Matthews, I., Ishikawa, T., and Baker, S. (2004). The Template Update Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810–815.

[Mendez-Vazquez et al., 2013] Mendez-Vazquez, H., Martinez-Diaz, Y., and Chai, Z. (2013). Volume Structured Ordinal Features with Background Similarity Measure for Video Face Recognition. *International Conference on Biometrics (ICB)*.

[Merad et al., 2010] Merad, D., Aziz, K.-E., and Thome, N. (2010). Fast people counting using head detection from skeleton graph. In *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 233–240, Washington, DC, USA. IEEE Computer Society.

[Mian, 2008] Mian, A. (2008). Unsupervised Learning from Local Features for Video-based Face Recognition. *International Conference on Automatic Face and Gesture Recognition.*

[Mikolajczyk et al., 2004] Mikolajczyk, K., Schmid, C., and Zisserman, A. (2004). Human Detection Based on a Probabilistic Assembly of Robust Part Detectors. In *European conference on Computer Vision (ECCV)*, volume 3021 of *Lecture Notes in Computer Science*, pages 69–82. Springer.

[Milan et al., 2013] Milan, A., Schindler, K., and Roth, S. (2013). Detection- and Trajectory-Level Exclusion in Multiple Object Tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3682–3689. IEEE.

[Miller et al., 1997] Miller, M. L., Stone, H. S., , Cox, I. J., and Cox, I. J. (1997). Optimizing Murty's Ranked Assignment Method. *IEEE Transactions on Aerospace and Electronic Systems*, 33:851–862.

[Munkres, 1957] Munkres, J. R. (1957). Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

[Myung, 2003] Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *J. Math. Psychol.*, 47:90–100.

[Nguyen and Bai, 2010] Nguyen, H. V. and Bai, L. (2010). Cosine Similarity Metric Learning for Face Verification. In *Asian Conference on Computer Vision (ACCV)*, volume 6493 of *Lecture Notes in Computer Science*, pages 709–720.

[Nguyen et al., 2009] Nguyen, H. V., Bai, L., and Shen, L. (2009). Local Gabor Binary Pattern Whitened PCA: A Novel Approach for Face Recognition from

Single Image Per Person. In *International Conference on Biometrics (ICB)*, volume 5558 of *Lecture Notes in Computer Science*, pages 269–278. Springer.

[Ni and Caplier, 2011] Ni, W. and Caplier, A. (2011). Newton optimization based Congealing for facial image alignment. In *International Conference on Image Processing (ICIP)*, pages 577–580. IEEE.

[Ni et al., 2012] Ni, W., Vu, N.-S., and Caplier, A. (2012). Lucas-Kanade based entropy congealing for joint face alignment. *Image Vision Computer*, 30(12):954–965.

[Oh et al., 2004] Oh, S., Russell, S., and Sastry, S. (2004). Markov Chain Monte Carlo Data Association for General Multiple-Target Tracking Problems. In *Conference on Decision and Control (CDC)*.

[Ojala, 1996] Ojala, T. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59.

[Ojala et al., 2002] Ojala, T., Pietikaeinen, M., and Maeenpaeae, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

[Okuma et al., 2004] Okuma, K., Taleghani, A., de Freitas, N., Little, J. J., and Lowe, D. G. (2004). A Boosted Particle Filter: Multitarget Detection and Tracking. In *European conference on Computer Vision (ECCV)*, volume 3021 of *Lecture Notes in Computer Science*, pages 28–39. Springer.

[Ortiz et al., 2013] Ortiz, E. G., Wright, A., and Shah, M. (2013). Face Recognition in Movie Trailers via Mean Sequence Sparse Representation-Based Classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3531–3538. IEEE.

[Ozkan and Duygulu, 2006] Ozkan, D. and Duygulu, P. (2006). A Graph Based Approach for Naming Faces in News Photos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1477–1482. IEEE Computer Society.

[Papageorgiou and Poggio, 2000] Papageorgiou, C. and Poggio, T. (2000). A Trainable System for Object Detection. *International Journal of Computer Vision*, 38(1):15–33.

[Pasula et al., 1999] Pasula, H., Russell, S., Ostland, M., and Ritov, Y. (1999). Tracking Many Objects with Many Sensors. *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1160–1167.

[Peng et al., 2010] Peng, Y., Ganesh, A., Wright, J., Xu, W., and Ma, Y. (2010). RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 763–770. IEEE.

[Perez et al., 2002] Perez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002). Color-Based Probabilistic Tracking. In *European conference on Computer Vision (ECCV)*, volume 2350 of *Lecture Notes in Computer Science*, pages 661–675. Springer.

[Pietikaeinen et al., 2011] Pietikaeinen, M., Hadid, A., Zhao, G., and Ahonen, T. (2011). Local Binary Patterns for Still Images. In *Computer Vision Using Local Binary Patterns*, volume 40 of *Computational Imaging and Vision*, pages 13–47. Springer London.

[Place, 2012] Place, L. (2012). A place for concepts of IT based modern living.

[Platt, 1999] Platt, J. C. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

[Prisacariu and Reid, 2009] Prisacariu, V. and Reid, I. (2009). fastHOG - a realtime GPU implementation of HOG. Technical Report 2310/09, Department of Engineering Science, Oxford University.

[Qin and Shelton, 2012] Qin, Z. and Shelton, C. R. (2012). Improving multi-target tracking via social grouping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1972–1978. IEEE.

[Rahimi et al., 2008] Rahimi, A., Morency, L.-P., and Darrell, T. (2008). Reducing drift in differential tracking. *Computer Vision and Image Understanding*, 109(2):97–111.

[Reid, 1979] Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854.

[Rodriguez et al., 2011] Rodriguez, M., Sivic, J., Laptev, I., and Audibert, J.-Y. (2011). Density-aware person detection and tracking in crowds. *International Conference on Computer Vision (ICCV).*

[Ross et al., 2008] Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental Learning for Robust Visual Tracking. *International Journal Computer Vision*, 77(1-3):125–141.

[Rougier and Meunier, 2010] Rougier, C. and Meunier, J. (2010). 3D Head Trajectory Using a Single Camera. In *International Conference on Image and Signal Processing (ICISP)*, volume 6134 of *Lecture Notes in Computer Science*, pages 505–512. Springer.

[Sabzmeydani and Mori, 2007] Sabzmeydani, P. and Mori, G. (2007). Detecting Pedestrians by Learning Shapelet Features. In *Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society.

[Saito et al., 2010] Saito, M., Kitaguchi, K., Kimura, G., and Hashimoto, M. (2010). Human detection from fish-eye image by Bayesian combination of probabilistic appearance models. *International Conference on Systems, Man and Cybernetics*, pages 243–248.

[Santner et al., 2010] Santner, J., Leistner, C., Saffari, A., Pock, T., and Bischof, H. (2010). PROST: Parallel robust online simple tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 723–730. IEEE.

[Saragih et al., 2009] Saragih, J. M., Lucey, S., and Cohn, J. (2009). Face Alignment through Subspace Constrained Mean-Shifts. In *International Conference of Computer Vision (ICCV).*

[Scaramuzza et al., 2006] Scaramuzza, D., Martinelli, A., and Siegwart, R. (2006). A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In *International Conference on Computer Vision Systems*, Washington, DC, USA.

[Schwalbe, 2005] Schwalbe, E. (2005). Geometric modelling and calibration of fisheye lens camera systems. In *Proceedings 2nd Panoramic Photogrammetry Workshop, Int. Archives of Photogrammetry and Remote Sensing*, pages 5–8.

[Shu et al., 2012] Shu, G., Dehghan, A., Oreifej, O., Hand, E., and Shah, M. (2012). Part-based multiple-person tracking with partial occlusion handling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1815–1821. IEEE.

[Silveira and Malis, 2007] Silveira, G. and Malis, E. (2007). Real-time Visual Tracking under Arbitrary Illumination Changes. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 0:1–6.

[Singh et al., 2008] Singh, V. K., Wu, B., and Nevatia, R. (2008). Pedestrian Tracking by Associating Tracklets Using Detection Residuals. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 1–8, Washington, DC, USA. IEEE Computer Society.

[Smith et al., 2014] Smith, B. M., Brandt, J., Lin, Z., and Zhang, L. (2014). Nonparametric Context Modeling of Local Appearance for Pose- and Expression-Robust Facial Landmark Localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Song et al., 2010] Song, B., Jeng, T.-Y., Staudt, E., and Chowdhury, A. K. R. (2010). A Stochastic Graph Evolution Framework for Robust Multi-target Tracking. In *European conference on Computer Vision (ECCV)*, volume 6311 of *Lecture Notes in Computer Science*, pages 605–619. Springer.

[Stalder et al., 2010] Stalder, S., Grabner, H., and Gool, L. J. V. (2010). Cascaded Confidence Filtering for Improved Tracking-by-Detection. In *European conference on Computer Vision (ECCV)*, pages 369–382.

[Stallkamp et al., 2007] Stallkamp, J., Ekenel, H. K., and Stiefelhagen, R. (2007). Video-based Face Recognition on Real-World Data. *International Conference on Computer Vision (ICCV)*, pages 1–8.

[Stauffer, 2003] Stauffer, C. (2003). Estimating tracking sources and sinks. In *Processing of the IEEE Workshop on Event Mining*, pages 259–266, Madison, WI.

[Sudowe and Leibe, 2011] Sudowe, P. and Leibe, B. (2011). Efficient Use of Geometric Constraints for Sliding-window Object Detection in Video. In *Proceedings of the International Conference on Computer Vision Systems*, pages 11–20, Berlin, Heidelberg. Springer-Verlag.

[Sun et al., 2013] Sun, Y., Wang, X., and Tang, X. (2013). Hybrid Deep Learning for Face Verification. In *International Conference on Computer Vision (ICCV).*

[Taigman et al., 2014] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

[Tan and Triggs, 2007] Tan, X. and Triggs, B. (2007). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. In *International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, volume 4778 of *Lecture Notes in Computer Science*, pages 168–182. Springer.

[Tang et al., 2007] Tang, F., Brennan, S., Zhao, Q., and Tao, H. (2007). Co-Tracking Using Semi-Supervised Support Vector Machines. In *International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE.

[Toscani, 2015] Toscani, P. (2015). Pierre Toscani Photographe en montagne. http://www.pierretoscani.com/images/echo_fisheyes/Figure-05.jpg. Accessed: 21.02.2015.

[Turk and Pentland, 1991] Turk, M. and Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

[Tuzel et al., 2006] Tuzel, O., Porikli, F., and Meer, P. (2006). Region Covariance: A Fast Descriptor for Detection and Classification. In *European conference on Computer Vision (ECCV)*, pages 589–600, Berlin, Heidelberg. Springer-Verlag.

[ul Hussain et al., 2012] ul Hussain, S., Napoléon, T., and Jurie, F. (2012). Face Recognition using Local Quantized Patterns. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association.

[ul Hussain and Triggs, 2012] ul Hussain, S. and Triggs, B. (2012). Visual Recognition using Local Quantized Patterns. In *European conference on Computer Vision (ECCV)*, Berlin, Heidelberg.

[Vandewiele et al., 2012] Vandewiele, F., Motamed, C., and Yahiaoui, T. (2012). Visibility management for object tracking in the context of a fisheye camera network. In *International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6.

[Venkatesh et al., 2012] Venkatesh, B. S., Descamps, A., and Carincotte, C. (2012). Counting People in the Crowd Using a Generic Head Detector. In *nternational Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 470–475. IEEE Computer Society.

[Viola and Jones, 2002] Viola, P. and Jones, M. (2002). Robust Real-time Object Detection. *International Journal of Computer Vision.*

[Viola et al., 2005] Viola, P. A., Jones, M. J., and Snow, D. (2005). Detecting Pedestrians Using Patterns of Motion and Appearance. *International Journal of Computer Vision*, 63(2):153–161.

[Walk et al., 2010] Walk, S., Majer, N., Schindler, K., and Schiele, B. (2010). New features and insights for pedestrian detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1030–1037. IEEE.

[Wang et al., 2009] Wang, H., Wang, Y., and Cao, Y. (2009). Video-based Face Recognition : A Survey. *World Academy of Science Engineering and Technology*, 60:293–302.

[Wang, 2006] Wang, M.-l. (2006). An Intelligent Surveillance System Based on an Omnidirectional Vision Sensor. *IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6.

[Wang and Chen, 2009] Wang, R. and Chen, X. (2009). Manifold Discriminant Analysis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 429–436. IEEE.

[Wang et al., 2012] Wang, R., Guo, H., Davis, L. S., and Dai, Q. (2012). Covariance discriminative learning: A natural and efficient approach to image set classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2496–2503. IEEE.

[Wang, 2013] Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3 – 19. Extracting Semantics from Multi-Spectrum Video.

[Weinberger et al., 2006] Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. In *Neural Information Processing Systems Foundation (NIPS)*. MIT Press.

[Weinberger and Saul, 2008] Weinberger, K. Q. and Saul, L. K. (2008). Fast solvers and efficient implementations for distance metric learning. In *International Conference on Machine Learning (ICML)*, volume 307 of *ACM International Conference Proceeding Series*, pages 1160–1167. ACM.

[Weiser et al., 1999] Weiser, M., Gold, R., and Brown, J. S. (1999). The Origins of Ubiquitous Computing Research at PARC in the Late 1980s. *IBM Systems Journal*, 38(4):693–696.

[Wojek et al., 2008] Wojek, C., Dorkó, G., Schulz, A., and Schiele, B. (2008). Sliding-Windows for Rapid Object Class Localization: A Parallel Technique. In *Pattern Recognition (DAGM)*.

[Wojek and Schiele, 2008] Wojek, C. and Schiele, B. (2008). A Performance Evaluation of Single and Multi-feature People Detection. In *DAGM-Symposium*, volume 5096 of *Lecture Notes in Computer Science*, pages 82–91. Springer.

[Wolf et al., 2011] Wolf, L., Hassner, T., and Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Wolf et al., 2010] Wolf, L., Hassner, T., and Taigman, Y. (2010). Similarity Scores Based on Background Samples. In *Asian Conference on Computer Vision (ACCV)*, pages 88–97, Berlin, Heidelberg. Springer-Verlag.

[Wolf and Levy, 2013] Wolf, L. and Levy, N. (2013). The SVM-Minus Similarity Score for Video Face Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3523–3530. IEEE.

[Wong et al., 2011] Wong, Y., Chen, S., Mau, S., Sanderson, C., and Lovell, B. (2011). Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. In *Biometrics Workshop, Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 81 – 88, Colorado Springs, USA. IEEE.

[Wu and Nevatia, 2007] Wu, B. and Nevatia, R. (2007). Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision*, 75(2):247–266.

[Wu and Ai, 2008] Wu, Y. and Ai, X. (2008). Face Detection in Color Images Using AdaBoost Algorithm Based on Skin Color Information. In *International Workshop on Knowledge Discovery and Data Mining (WKDD)*, pages 339–342. IEEE Computer Society.

[Wu et al., 2012a] Wu, Y., Cheng, J., Wang, J., Lu, H., Wang, J., Ling, H., Blasch, E., and Bai, L. (2012a). Real-Time Probabilistic Covariance Tracking With Efficient Model Update. *IEEE Transactions on Image Processing*, 21(5):2824–2837.

[Wu et al., 2013] Wu, Y., Lim, J., and Yang, M.-H. (2013). Online Object Tracking: A Benchmark. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 0:2411–2418.

[Wu et al., 2012b] Wu, Y., Minoh, M., Mukunoki, M., and Lao, S. (2012b). Set Based Discriminative Ranking for Recognition. In *European conference on Computer Vision (ECCV)*, volume 7574 of *Lecture Notes in Computer Science*, pages 497–510. Springer.

[Xie et al., 2012] Xie, D., Dang, L., and Tong, R. (2012). Video based head detection and tracking surveillance system. In *International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 2832–2836. IEEE.

[Xing et al., 2002] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. J. (2002). Distance Metric Learning with Application to Clustering with Side-Information. In *Neural Information Processing Systems Foundation (NIPS)*, pages 505–512.

[Xu et al., 2012] Xu, Z. E., Weinberger, K. Q., and Chapelle, O. (2012). Distance Metric Learning for Kernel Machines. *CoRR*.

[Yan et al., 2013] Yan, J., Zhang, X., Lei, Z., and Li, S. Z. (2013). Face detection by structural models. *Image and Vision Computing*.

[Yang and Nevatia, 2012] Yang, B. and Nevatia, R. (2012). Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1918–1925. IEEE.

[Yang and Nevatia, 2014] Yang, B. and Nevatia, R. (2014). Multi-Target Tracking by Online Learning a CRF Model of Appearance and Motion Patterns. *International Journal of Computer Vision*, 107(2):203–217.

[Yang et al., 2011] Yang, H., Shao, L., Zheng, F., Wang, L., and Song, Z. (2011). Recent Advances and Trends in Visual Tracking: A Review. *Neurocomputing*, 74(18):3823–3831.

[Yang et al., 2009] Yang, M., Wu, Y., and Hua, G. (2009). Context-aware visual tracking. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 31(7):1195–1209.

[Yilmaz et al., 2006] Yilmaz, A., Javed, O., and Shah, M. (2006). Object Tracking: A Survey. *ACM Computing Surveys*, 38(4).

[Yoon et al., 2012] Yoon, J. H., Kim, D. Y., and Yoon, K.-J. (2012). Visual Tracking via Adaptive Tracker Selection with Multiple Features. In *European conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 28–41. Springer.

[Yu et al., 2007] Yu, Q., Medioni, G. G., and Cohen, I. (2007). Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. *Conference on Computer Vision and Pattern Recognition (CVPR).*

[Yu et al., 2013] Yu, X., Huang, J., Zhang, S., Yan, W., and Metaxas, D. (2013). Pose-free Facial Landmark Fitting via Optimized Part Mixtures and Cascaded Deformable Shape Model. *International Conference on Computer Vision (ICCV).*

[Yuan et al., 2011] Yuan, X., Song, Y., and Wei, X. (2011). Automatic surveillance system using fish-eye lens camera. *Chinese Optics Letters*, 9(2):021101.

[Zhan et al., 2007] Zhan, C., Li, W., Ogunbona, P., and Safaei, F. (2007). Real-time Facial Feature Point Extraction. In *Proceedings of the Multimedia 8th Pacific Rim Conference on Advances in Multimedia Information Processing*, PCM'07, pages 88–97, Berlin, Heidelberg. Springer-Verlag.

[Zhang and Zhang, 2010] Zhang, C. and Zhang, Z. (2010). A Survey of Recent Advances in Face Detection. Technical Report MSR-TR-2010-66.

[Zhang et al., 2004] Zhang, G., Huang, X., Li, S. Z., Wang, Y., and Wu, X. (2004). Boosting Local Binary Pattern (LBP)-Based Face Recognition. In *SINOBIOMETRICS*, volume 3338 of *Lecture Notes in Computer Science*, pages 179–186. Springer.

[Zhang et al., 2007] Zhang, W., Zelinsky, G. J., and Samaras, D. (2007). Real-time Accurate Object Detection using Multiple Resolutions. In *International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE.

[Zhang and Gao, 2009] Zhang, X. and Gao, Y. (2009). Face recognition across pose: A review. *Pattern Recognition*, 42(11):2876–2896.

[Zhang et al., 2011] Zhang, Z., Wang, C., and Wang, Y. (2011). Video-based Face Recognition: State of the Art. *Chinese Journal of Computers*, pages 1–9.

[Zhao and Pietikainen, 2007] Zhao, G. and Pietikainen, M. (2007). Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928.

[Zhong et al., 2012] Zhong, W., Lu, H., and Yang, M.-H. (2012). Robust object tracking via sparsity-based collaborative model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1838–1845. IEEE.

[Zhou et al., 2009] Zhou, H., Yuan, Y., and Shi, C. (2009). Object tracking using {SIFT} features and mean shift. *Computer Vision and Image Understanding*, 113(3):345 – 352. Special Issue on Video Analysis.

[Zhu et al., 2009] Zhu, J., Gool, L. J. V., and Hoi, S. C. H. (2009). Unsupervised face alignment by robust nonrigid mapping. In *International Conference on Computer Vision (ICCV)*, pages 1265–1272. IEEE.

[Zhu et al., 2013] Zhu, P., Zhang, L., Zuo, W., and Zhang, D. (2013). From Point to Set: Extend the Learning of Distance Metrics. In *International Conference on Computer Vision (ICCV)*.

[Zhu et al., 2006] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1491–1498. IEEE Computer Society.

[Zhu and Ramanan, 2012] Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886. IEEE.

[Zini et al., 2014] Zini, L., Noceti, N., Fusco, G., and Odone, F. (2014). Structured multi-class feature selection with an application to face recognition. *Pattern Recognition Letters*.