Weakly Supervised and On-line Machine Learning for Object Tracking and Recognition in Images and Videos

Stefan Duffner

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

HABILITATION À DIRIGER DES RECHERCHES

présentée devant

l'Institut National des Sciences Appliquées de Lyon
et l'Université Claude Bernard LYON I

Spécialité : Informatique

Weakly Supervised and On-line Machine Learning for Object Tracking and Recognition in Images and Videos

par

Stefan Duffner

Soutenue le 5 avril 2019 devant la commission d'examen

Composition du jury

Rapporteurs :	Pr. Michel Paindavoine	Université de Bourgogne
	DR François Bremond	INRIA Sophia Antipolis
	Pr. Nicolas Thome	CNAM
Examinateurs :	MER Jean-Marc Odobez	Idiap Research Institute / EPFL
	DR Carole Lartizien	CNRS
	Pr. Christophe Garcia	INSA de Lyon
	Pr. Mohand-Saïd Hacid	Université Claude Bernard Lyon 1

Acknowledgements

First of all, I would like to express my profound gratitude to my parents who, not only gave me all the opportunities and support to accomplish this work throughout all these years, but also endowed me with some extremely precious traits and abilities that helped me to develop myself, professionally as well as personally. I also want to thank my wife who supported me all the time and always respected me and my personal choices regarding my career.

I would like to especially thank Christophe Garcia who accompanied me for a large part of my professional career. He was, and he still is, a true mentor for me. He guided me all these years and set a real example to me, both from a professional and from a human point of view. I am deeply thankful, not only for all his advice, support, and ideas on a scientific level, but also for the principles, virtues and values that he gave me and still drive my work today.

I also want to thank Jean-Marc Odobez with whom I worked for 4 years in Switzerland, from 2008 to 2012. I have very positive memories of this time and of this collaboration, and I left it behind with great regret. In fact, much of my current knowledge and scientific and professional abilities I owe to Jean-Marc.

Another fruitful collaboration has been the one with Atilla Baskurt with whom I co-supervised two PhD theses at LIRIS. This habilitation work would not have been possible without him and the professional skills he conveyed me.

I would like to thank all the persons I met during my career and with whom I had the chance to work with or the exchange ideas and thoughts. For example, Professor Oliver Bittel who supervised my first research project at the University of Applied Sciences in Konstanz, Germany, which was my first contact with machine learning with convolutional neural networks in 2002! Also Stéphanie Jehan-Besson and Marinette Revenu at the GREYC laboratory in Caen helped me with my first steps in research for image and video processing algorithms. I thank also Professor Hans Burkhardt at the University of Freiburg, Germany, for supervising my PhD work and the interesting scientific discussions we had. Also Franck Mamalet, Sébastien Roux, Grégoire Lefebvre, Muriel Visani, Zora Saidane at Orange Labs, Rennes. And at Idiap, in Martigny: Hervé Bourlard, Sébastien Marcel, Daniel Gatica-Perez, Petr Motlicek, Philip N. Garner, John Dines, Danil Korchagin, Carl Scheffler, Jagannadan Varadarajan, Elisa Ricci, Silèye Ba, Rémi Emonet, Radu-Andrei Negoescu, Xavier Naturel. At LIRIS, I would like to thank my colleagues, Mathieu Lefort, Frédéric Armetta, Véronique Eglin, Khalid Idrissi, Serge Miguet, Michaela Scuturici, Liming Chen, Céline Robardet, Marc Plantevit, Jean-François Boulicaut, just to name a few. Special thanks also to Thierry Château and to Renato Cintra, and to all those I may have forgotten – I appologise.

Finally, I want to sincerely thank the members of my habilitation jury and manuscript reviewers for having accepted and performed this task: thank you Michel Paindavoine, Nicolas Thome, François Bremond, Carole Lartizien, Mohand-Säid Hacid, and again, Jean-Marc Odobez and Christophe Garcia.

Last but not least, I would not be able to defend my habilitation without the work of my PhD students: Samuel Berlemont (now at Orange Labs, Meylan), Salma Moujahid (now at Valeo, Paris), Lilei Zheng (now at Northwestern Polytechnical University, Xi'an, China) and Yiqiang Chen (now at Huawei, Shanghai, China). Thank you so much for your work!

Abstract

This manuscript summarises the work that I have been involved in for my post-doctoral research and in the context of my PhD supervision activities during the past 11 years. I have conducted this work partly as a post-doctoral researcher at the Idiap Research Institute in Switzerland, and partly as an associate professor at the LIRIS laboratory and INSA Lyon in France.

The technical section of the manuscript comprises two main parts: the first part being on on-line learning approaches for visual object tracking in dynamic environments, and the second part on similarity metric learning algorithms and Siamese Neural Networks (SNN).

I first present our work on on-line multiple face tracking in a dynamic indoor environment, where we focused on the aspects of track creation and removal for long-term tracking. The automatic detection of the faces to track is challenging in this setting because they may not be detected for long periods of time, and false detections may occur frequently. Our proposed algorithm consisted in a recursive Bayesian framework with a separate track creation and removal step based on Hidden Markov Models including observation likelihood functions that are learnt off-line on a set of static and dynamic features related to the tracking behaviour and the objects' appearance. This approach is very efficient and showed superior performance to the state of the art in on-line multiple object tracking. In the same context, we further developed a new on-line algorithm to estimate the Visual Focus of Attention from videos of persons sitting in a room. This unsupervised on-line learning approach is based on an incremental k-means algorithm and is able to automatically extract, from a video stream, the targets that the persons are looking at in a room.

I further present our research on on-line learnt robust appearance models for single-object tracking. In particular we focused on the problem of model-free, on-line tracking of arbitrary objects, where the state and model of the object to track is initialised in the first frame and updated throughout the rest of the video. Our first approach, called PixelTrack, consists in a combined detection and segmentation framework that robustly learns the appearance of the object to track and avoids drift by an effective on-line co-training algorithm. This method showed excellent tracking performance on public benchmarks, both in terms of robustness and speed, and is particularly suitable for tracking deformable objects. The second tracking approach, called MCT, employs an on-line learnt discriminative classifier that stochastically samples the training instances from a dynamic probability density function that is computed from moving and possibly distracting image background regions. The use of this motion context showed to be very effective and lead to a significant gain in the overall tracking robustness and performance. We extended this idea by designing a set of features that concisely describe the visual context of the overall scene shown in a video at a given point in time. Then, we applied several complementary tracking algorithms on a set of training videos and computed the corresponding context features for each frame. Finally, we trained a discriminative classifier off-line that estimates the most suitable tracker for a given context, and applied it on-line in an effective tracker-selection framework. Evaluated on several different "pools" of individual trackers, the combined model lead to an increased performance in terms of accuracy and robustness on challenging public benchmarks.

In the second part of the manuscript, I present several contributions related to SNNs for simi-

larity metric learning. First, we proposed a new objective function and training algorithm called Triangular Similarity Metric Learning that enhances the convergence behaviour and achieved state-of-the-art results on pairwise verification tasks, like face, speaker or kinship verification. Then, I present our work on SNNs for gesture classification from inertial sensor data, where we proposed a new class-balanced learning strategy operating on tuples of training samples and an objective function based on a polar sine formulation. Finally, I present several contributions on SNN with deeper and more complex Convolutional Neural Network models applied to the problem of person re-identification in images. In this context, we proposed different neural architectures and triplet learning methods that include semantic prior knowledge, *e.g.* on pedestrian attributes, body orientation and surrounding group context, using a combination of supervised and weakly supervised algorithms. Also, a new learning-to-rank algorithm for SNN, called Rank-Triplet, has been introduced and successfully applied to person re-identification. These recent works achieved state-of-the-art re-identification results on challenging pedestrian image datasets and opened new perspectives for future similarity metric approaches.

Contents

List of Figures

Introduction

1 Overview

In this chapter, I will first present the general context of our research on computer vision and machine learning – two areas that have been tightly linked and that I have been working on since roughly 15 years. After my Curriculum Vitae, I will give an overview of my post-doctoral research work during the last 10 years as well as my PhD supervision activities.

In this period, I have been working in two different laboratories and countries: first at the Idiap Research Institute (Martigny, Switzerland) in the team of Jean-Marc Odobez, and then, at LIRIS (Lyon, France) in the Imagine team with Christophe Garcia. In both teams, my research globally concerned the areas of computer vision and machine learning, but the context and applications have been slightly different. At the Idiap Research Institute, I have been mostly involved in a large European FP7 project and working on real-time on-line multi-object tracking methods using probabilistic approaches. At LIRIS, I continued the research on on-line methods for visual object tracking but then focused more on machine learning and neural network-based models that I have been already working on during my PhD thesis.

These two different contexts, both from a methodological and application point of view, lead to the two parts forming the technical portion of this manuscript: the first part focusing on visual object tracking and on-line learning appearance models, and the second part on similarity metric learning approaches with Siamese Neural Networks.

In the last chapter, I will draw general conclusions and outline some of the perspectives of our research.

2 Context

2.1 Computer Vision: from the laboratory "into the wild"

The research on Computer Vision has come a long way starting from image and signal processing techniques in the 1970s-1980s, *e.g.* filtering, de-noising, contour extraction, basic shape analysis, thresholding, geometrical model fitting *etc.*, that have been applied in relatively controlled conditions, for example, in camera-based production quality control systems, or for relatively elementary segmentation, detection, and recognition problems in constrained laboratory environments. Relatively simple parametric models, *e.g.* based on rules or, later, fuzzy logic or statistics, were employed to perform classification or regression and thus accomplish some basic perception tasks in a given environment. For example, the first visual object tracking approaches were based on relatively simple methods like cross-correlation or other template matching techniques [95, 123, 201] applied frame-by-frame on the raw pixel intensities, edges or other low-level visual features computed on image regions or key points (see Fig. 2.1). Motion and optical flow estimation [263, 339] as well as background subtraction techniques [141, 350, 403] have been used to include the temporal aspect in video analysis and in particular object tracking. Also various shape tracking approaches relying on parametric models (*e.g.* ellipses, splines) [179] or non-parametric models (*e.g.* level sets) [91, 271] have been proposed during that time. However, these methods showed clear limitations with cluttered background and when the tracked objects underwent more severe image deformations, like lighting changes or rotations.

In the 1990s and 2000s, the advent of effective machine learning algorithms being able to learn from examples and operate on high-dimensional vector spaces, and their combination with classical image processing techniques led to powerful methods for a variety of different automatic perception tasks, *e.g.* image segmentation or enhancement, object detection and recognition, image classification, tracking, motion estimation or 3D reconstruction just to name a few. These approaches were generally based on a two-stage procedure: first, the extraction of (local) visual features that have been designed "manually" in order to be robust and invariant to different types of noise (histograms of colour, texture *etc.*) and, second, the classifier that has been (automatically) trained beforehand on these features computed on a training data set. In online visual object tracking, mostly probabilistic Bayesian methods (Kalman Filters, Particle Filters) have been proposed [139, 198, 214, 304], which were able to perform the inference in real-time, frame-by-frame, and using robust appearance likelihood models (colour histograms, or simple shape models). This not only allowed to track one or several objects efficiently in realistic environments, but also to include the *uncertainty* in the estimation and cope with several hypotheses in parallel. Although these predominantly probabilistic algorithms required some manual parametrisation, they have been extended with observation likelihood models resulting from statistical machine learning. That means, either a classical discriminative classifier (SVM, neural network) is trained on an annotated data set for detecting objects of a specific category (faces, pedestrians, cars *etc.*) [143, 291], or an on-line learning classifier (on-line SVM, on-line

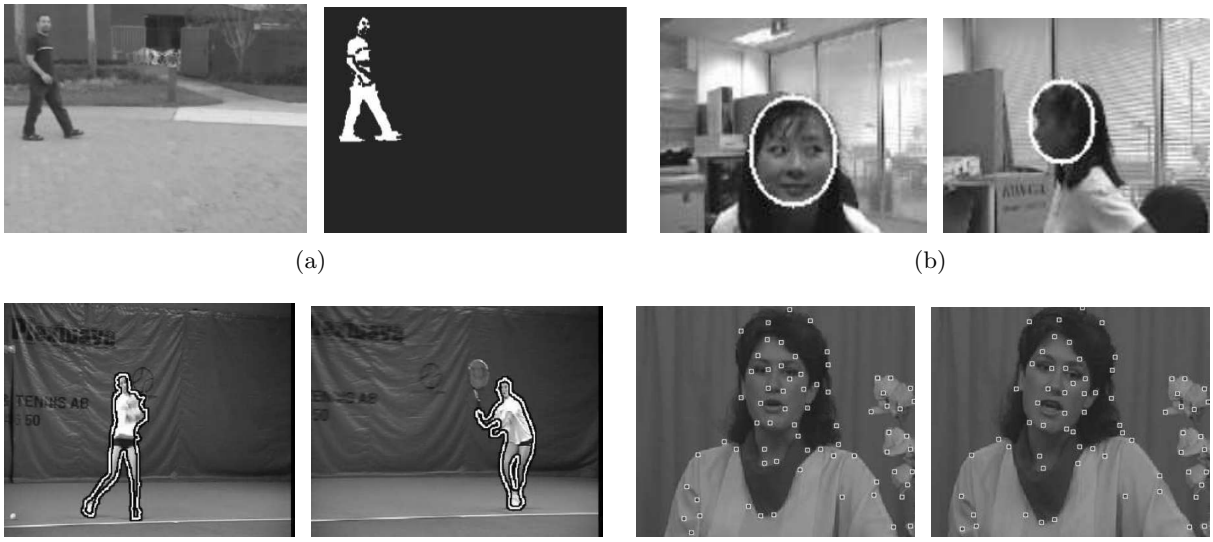


Figure 2.1: Classical object detection and tracking approaches: (a) background subtraction, (b) template and geometrical shape matching, (c) level sets, (d) key point and motion tracking.

Adaboost *etc.*) is trained "on-the-fly" on the particular object to track in the given video [68, 74, 165, 175, 206], (see Fig. 2.2). These powerful on-line classifiers have been extensively applied in so-called tracking-by-detection methods and mostly for *single* object tracking, where the object's bounding box is detected at each frame (within a search window), and thus a tedious parametrisation of likelihood or motion models was not necessary, or to a lesser extent. The use of machine learning techniques for building more discriminative appearance models also led to considerable progress in on-line multi-object tracking, notably improving the performance of data association between consecutive frames. However, they are computationally quite expensive, and when the models are adapted on-line for each tracked object the complexity, in general, increases linearly with the number of objects, which is an issue in real-time applications. In the first part of this manuscript, we will present some of our previous research that addressed these robust visual on-line learning problems in the context of on-line single and multiple object tracking.

From 2012 on, the Computer Vision field has changed very rapidly with the broad adoption of Convolutional Neural Networks (introduced in the 1990s) that learn the parameters of feature extraction and classification jointly from annotated example images, and employ a layer-wise feed-forward architecture which can be effectively trained by the Gradient Backpropagation algorithm. More and more complex models have been proposed, extracting deep and semantic feature hierarchies and showing excellent performance on realistic data under challenging conditions. However, to be effective, they rely on large annotated image datasets (like ImageNet) and considerable computational resources (mostly GPUs). Also, several deep learning object tracking approaches have been proposed during the last years [242, 289, 391]. They generally follow the tracking-by-detection approach and focus on the construction of robust appearance models to track any object under challenging conditions. In this context, these learnt feature hierarchies showed excellent performance, and the availability of large volumes of annotated video data enabled further enabled to inclusion of learnt motion patterns. Nevertheless, with respect to the tracking problem, some fundamental issues remain. That is, the effective on-line learning to quickly adapt to new conditions and to be able to operate in dynamic environments

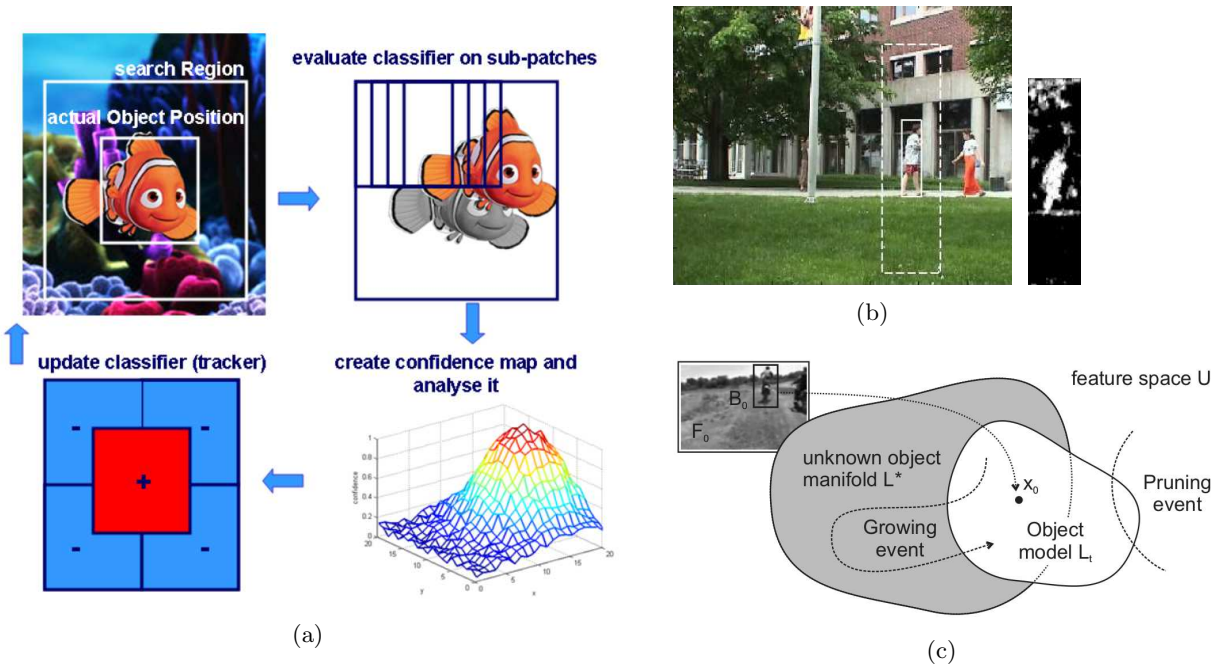


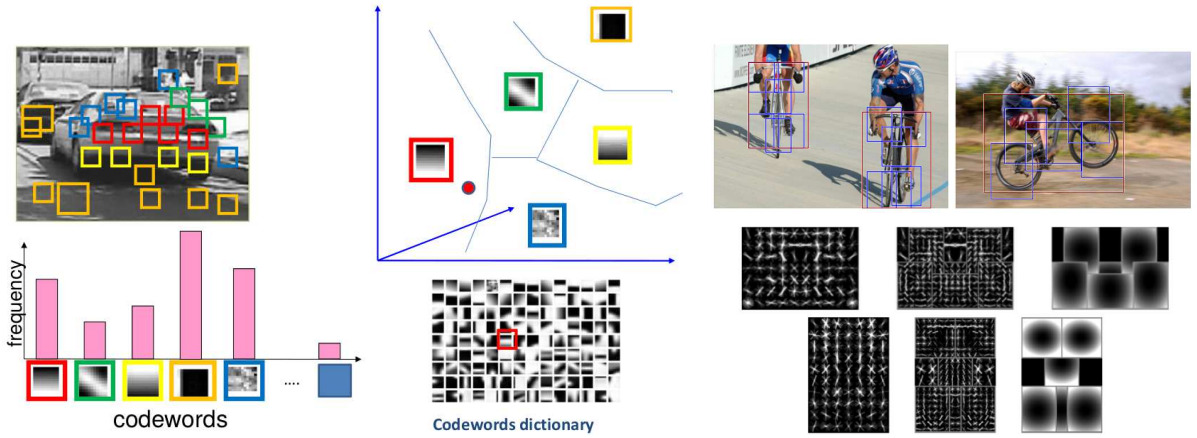
Figure 2.2: Tracking-by-detection approach. A discriminative classifier is trained on-line with foreground and background images patches (a) or pixels (b). Classical motion-based trackers have been combined with on-line learnt detectors to select the features to learn (c).

without forgetting previously acquired knowledge in the long term; and the design of models and inference algorithms of *low complexity* and *high generalisation capacity* (without the need of large annotated datasets).

Our work described in this manuscript especially focused on these aspects.

2.2 Learning representations and making decisions

As mentioned above, in Computer Vision, classical machine learning algorithms operated on "hand-crafted", generic features such as colour or texture histograms, Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT) features or dictionaries (bags of visual words), and the trained classifiers evolved from relatively simple decision trees or probabilistic methods such as Bayesian Networks to more "advanced" models based on Support Vector Machines and kernel-based methods that are able to cope with relatively high-dimensional input features vectors and that can be optimised efficiently (see Fig. 2.3). On the one hand, much research has been performed in improving the performance of descriptors and visual features, *i.e.* increasing their robustness and invariance to typical image noise and making them more discriminative with respect to a given classification task. On the other hand, numerous works concentrated on improving the classification algorithms. For example, by designed specific kernel functions for kernel-based classifiers like SVMs, or by structuring the models and inference in a way that better represents the data, like in probabilistic graphical models or deformable part models (DPM) [147] (see Fig. 2.3). Also, the combination of these classifiers using, for example, bagging or boosting techniques increased the overall classification performance [235, 340, 379]. In these approaches, the design of robust features is crucial and much work has been performed in



(a) (b)

Figure 2.3: Extracting "hand-crafted" low-level features and (a) forming codeword dictionaries (bags of visual words) or (b) learning Deformable Part Models (DPM).

the 1990s and 2000s to "manually" create and refine such effective, invariant representations of the image data.

Convolutional Neural Networks (CNN), based on the fundamental work of K. Fukushima [152] and Y. LeCun [228], allow to overcome this manual feature design and learn discriminative features automatically from annotated data. Using neural architectures with several layers of subsequent convolution and pooling operations (and non-linear activation functions) together with the Gradient Backpropagation algorithm thus results in a hierarchy of learnt feature extractors from the lowest (pixel) to a higher (semantic) level (see Fig. 2.4). With the advent of cheap GPU hardware and optimised software libraries in the 2010s, more complex models with many layers have been used, and these deep neural networks achieved excellent performance on image classification [161, 220] and have then been widely adopted for the majority of Computer Vision problems. Neural networks have a long history in Computer Vision, and their advantage is their flexibility in the model complexity that comes with different architectures, *e.g.* the number of layers, feature maps and neurons, optimised in a uniform way through Gradient Backpropagation, and especially their robustness to noise in the input data. Also, different training strategies can be adopted depending on the amount of available annotated data and the nature of the given task to perform. Classically, neural networks has been trained in a supervised way for classification problems. But, semi-supervised and weakly supervised algorithms exist as well, *e.g.* with Siamese Neural Networks [99, 121]. With few labelled data, a common technique consists in retraining or fine-tuning the last layer(s) of an existing deep neural network model, trained for example on an image classification problem with the ImageNet dataset [325] (transfer learning). When no labelled training data is available, one can use an unsupervised learning approach with specific neural network architectures such as auto-encoders [187] and Generative Adversarial Networks (GAN) [163] to automatically extract high-level information for further analysis. Recently, many variants of such deep neural network architectures, training algorithms and loss functions have been proposed in the literature.

However, with deeper and more complex models come several difficulties and limitations. One problem is over-fitting, that has either been addressed by specific regularisation techniques such as DropOut or DropConnect or by increasing the training dataset and semi-supervised learning methods. The lack of annotated training data is a frequent issue for many application when

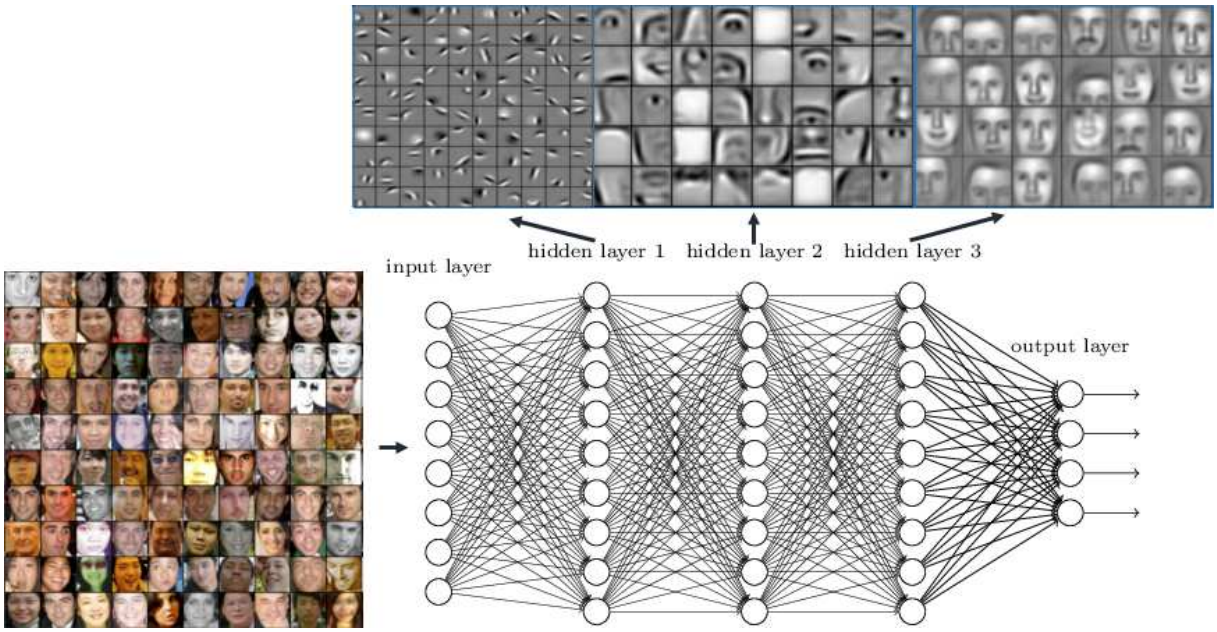


Figure 2.4: Learning deep semantic feature hierarchies.

using deep neural networks, and, as mentioned above, a common strategy is to perform transfer learning using an existing trained deep neural network model. However, in a dynamic evolving environment, for instance, with a mobile autonomous robot, one open question is how such deep and complex models can be continuously adapted and "fine-tuned" to perform optimally in new and unknown situations and conditions. How can we effectively integrate non-annotated or partially annotated data in a principal way to learn an model that generalises well? And how to learn with very few data?

With our work on semi-supervised similarity metric learning with Siamese Neural Networks presented in the second part, we tried to tackle some of these problems starting with more fundamental studies on small datasets and shallow neural network models and then extending some of the ideas to deeper CNN structures.

3 Curriculum vitae

Stefan DUFFNER, born on the 15/02/1978 in Schorndorf (Germany).

Appointment and career:

Institution: INSA Lyon	**Grade:** Maître de Conférences CN
Since: 1st September 2012	**CNU Section:** 27ème section
Departement: 50% Premier Cycle (PC), 50% Informatique	**Laboratory:** LIRIS

Professional address:

✉: INSA Lyon
Bâtiment Blaise Pascal
7 avenue Jean Capelle
69621 Villeurbanne Cedex
☎: +33 (0)4 72 43 63 65
@: stefan.duffner@insa-lyon.fr
: http://www.duffner-net.de

Education

2008 PhD in Computer Science from Freiburg University, Germany

Defended:	28/03/2008 Freiburg University, Germany
Title:	**"Face Image Analysis with Convolutional Neural Networks"**
Funding:	Orange Labs
Supervision:	Prof. Dr. Christophe Garcia (Orange Labs), Prof. Dr. Hans Burkhardt (Freiburg University, Germany)

2004 Master's degree in Computer Science (MSc), Freiburg University, Germany

1st year:	Freiburg University, Germany
2nd year:	École Nationale Supérieure d'Ingénieurs de Caen (ENSICAEN), GREYC, Image Team, Caen, Supervision Dr. Stéphanie Jehan-Besson (GREYC) Dr. Gerd Brunner (Freiburg University)

2002 Bachelor's degree University of Applied Sciences, Constance, Germany

Professional Experience

since 2012 LIRIS/CNRS, Imagine team, INSA de Lyon

Associate Professor

Research topics: image classification, object detection and recognition, video analysis, visual object tracking, machine learning, neural networks

2008-2012 Idiap Research Institute, Martigny, Switzerland

Post-doctoral researcher in computer vision and object tracking in the context of the European project TA2 (Together Anywhere, Together Anytime)
Team Dr. Jean-Marc Odobez

2004-2007 Orange Labs, Rennes, France

PhD in the field of object detection in images and videos using machine learning.

PhD thesis: *"Face Image Analysis with Convolutional Neural Networks"*

Supervision Prof. Dr. Christophe Garcia (Orange Labs),
Prof. Dr. Hans Burkhardt (Freiburg University, Germany)

Teaching

Recent teaching activities at the departments "Premier Cycle", "Informatique" as well as "Telecommunications" at INSA Lyon :

1A PCC CM/TD/TP "Algorithms and Programming"

2A PCC/SCAN CM/TD/TP "Algorithms and Programming", "Introduction to Databases"

3IF CM/TD/TP "Software engineering" (responsible of module)

3IF TD/TP "Probability Theory"

3IF/4IF TP "Operating Systems"

5IF CM/TP "Big Data Analytics" / "Machine Learning"

4TC CM/TP "Image and Video Processing"

5TC (SJTU) CM/TP "Computer Vision and Machine Learning"

5TC CM/TP "Scientific Computing and Data Analytics"

Publications

See publication list on page 135.

Other responsibilities and activities

Teaching:

- Responsible of the *Computer Science* module of second year SCAN, INSA Lyon (English teaching)
- Responsible of the *Software Engineering* module at the third year IF (computer science) department of INSA Lyon
- Responsible of personalised curricula (parcours aménagé) at IF department, INSA Lyon
- Member of the ATER candidate selection committee of INSA Lyon regarding the 27th CNU section ("Informatique")

Research:

- Council member of the Lyon Computer Science Federation ("Fédération Informatique de Lyon", FIL)
- Responsible of the topic "Image and Graphics" of the FIL
- Expertises for ANR (and Swiss FNS) research project proposals and ANRT CIFRE applications
- Numerous reviews for renowned journals and conferences (IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on Image Processing (IP), IEEE Transactions on Neural Networks and Learning Systems (NNLS), IEEE Transactions on Cybernetics (CYB), Pattern Recognition (PR) *etc.*)
- Co-organisation of a workshop on Deep Learning at the French conference RFIA 2017
- Co-organisation of a workshop on Neural Networks and Deep Learning (RNN) in Lyon 2018

4 Overview of research work and supervision

During the last 16 years, I conducted my research activities in five different academic and industrial research institutions in France, Germany and Switzerland. They are mainly related to the fields of computer vision, image and video analysis and machine learning, *i.e.* the automatic extraction, analysis and interpretation of semantic information from digital images and videos.

From 2002, I prepared my Master's degree at the University of Freiburg in Germany, focusing on topics concerning image processing, artificial intelligence, robotics, machine learning and applied mathematics. I performed part of my studies at the ENSICAEN in Caen, France, and my master thesis at the GREYC laboratory, on spatio-temporal object segmentation in videos. In 2004, I started my PhD research on Convolutional Neural Networks applied to face image analysis, at France Telecom R&D, Rennes, (now Orange Labs), and I defended it in the beginning of 2008 at the Freiburg University. Between 2008 and 2012, I worked as a post-doctoral researcher at the Idiap Research Institute in Martigny, Switzerland, on visual object tracking and probabilistic models for long-term multi-object face tracking. Since, 2012, I am a associate professor at INSA Lyon and the LIRIS laboratory, working on machine learning and computer vision for various applications, such as object detection, tracking and recognition, and face and gesture analysis.

This rich experience from several scientific, social and cultural contexts gave me a broad range of technical capacities and knowledge under diverse points of view and approaches, from a more specific to a more general level, professionally as well as personally.

In the following, I will briefly describe our research and my supervision work and, in the succeeding chapters, go into more technical detail on the research we have conducted after my PhD thesis.

4.1 Convolutional Neural Networks for face image analysis

During my PhD [56], supervised by Christophe Garcia at France Telecom R&D, Rennes, (now professor at LIRIS) and Prof. Dr. Hans Burkhardt at the University of Freiburg, Germany (now emeritus professor), we worked on new Convolutional Neural Network (CNN) models for the automatic analysis of faces in images. Our starting point was the well-known Convolutional Face Finder (CFF) from Garcia and Delakis [156], and we proposed new neural architectures and learning algorithms for different computer vision and face analysis problems. Our first contribution [45, 47, 50] concerned the automatic detection of facial feature points (facial landmarks) in face images in order to align the face to a canonical position for further processing (*e.g.* for recognition) (see Fig. 4.1). Then, we proposed a novel CNN-based regression approach for automatic face alignment that did not require an explicit detection of feature points [41]. We also

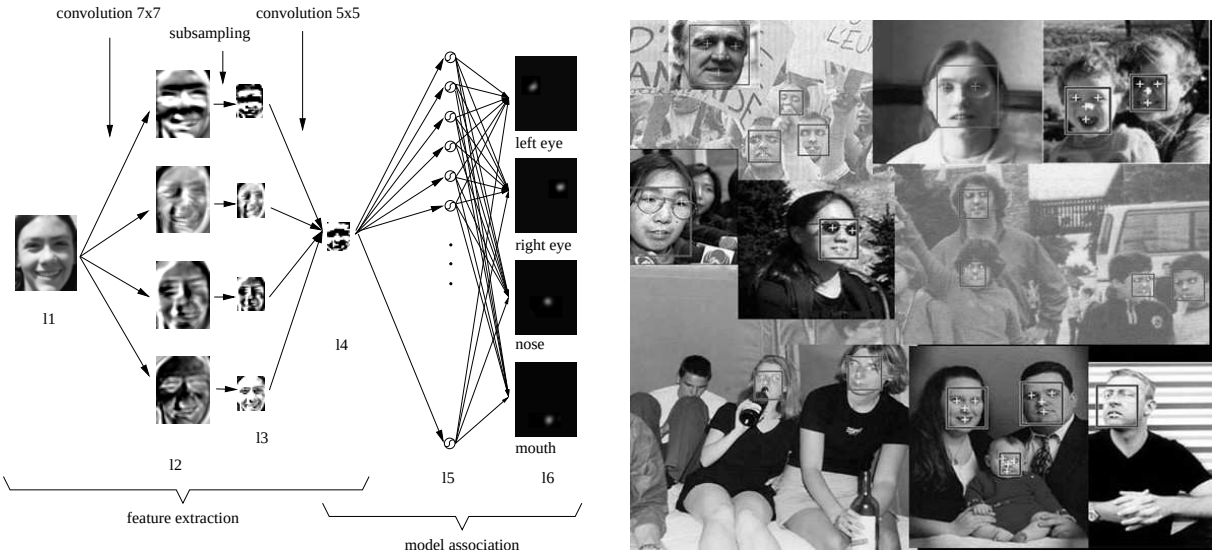


Figure 4.1: *Left:* proposed Convolutional Neural Network-based approach for facial landmark detection [45, 47]. *Right:* some example images showing the detection results "in the wild".

introduced an original approach for face recognition using a learnt CNN model for non-linear face image reconstruction [42]. Finally, we adapted the CFF model to detect other types of objects in images, such as football players [49], cars and motorbikes [52] (participating in the international challenge PASCAL VOCC 2005) and even virtual objects such as transparent logos in TV broadcasts.

The developed techniques and models show a strong robustness against noise and excellent performance under real-world conditions ("in the wild", see Fig. 4.1), and they have then been widely used for industrial applications within France Telecom and optimised for embedded devices and mobile phones [44], which lead to three patents [53–55].

4.2 Visual object tracking

I conducted several years of post-doctoral research at Idiap Research Institute in Martigny, Switzerland, where I worked in the team of Jean-Marc Odobez, on visual object tracking in the context of a large EU FP7 project called TA2, "Together Anywhere, Together Anytime". The project aimed at improving the feeling of "togetherness" of distant family members through technology, and our part consisted in developing a dynamic face and person detection and tracking algorithm operating in real-time on a video (and audio) stream in order to built a type of enhanced video-conferencing system that allows for a more engaging user experience. The main challenges in this setting were frequent occlusions, a varying number of persons to track and re-identify and arbitrary person and a room configurations as well as different lighting conditions. To tackle this multi-object tracking and identification problem, we proposed a novel multi-face tracking algorithm based on a probabilistic model and Bayesian inference [12, 38] specifically handling the problem of long-term tracking and robust track creation and deletion, and we integrated this method in a more complex multi-modal speaker detection, and visual focus of attention recognition system [11, 36, 37] (see Fig. 4.2). As faces are often difficult to detect in such dynamic environments, we also developed an enhanced upper-body detection algorithm based on the semantic segmentation of different body parts [10]. Finally, we proposed a new probabilistic

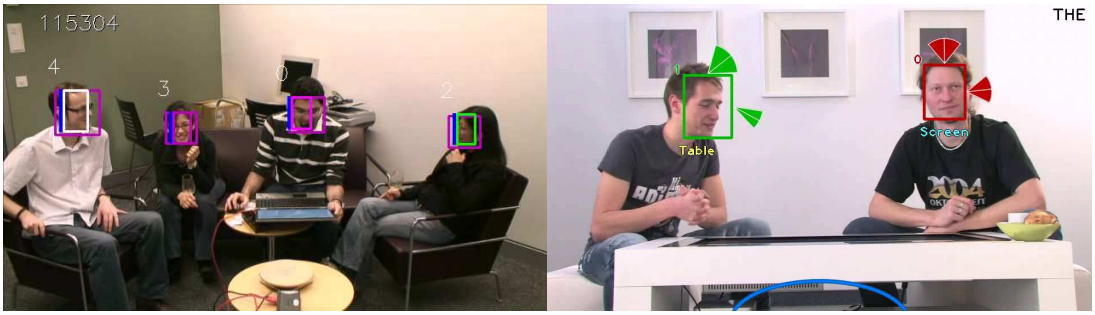


Figure 4.2: Real-time on-line multi-face tracking in dynamic environments with an RGB camera. People may move freely in the room and leave and enter the scene. *Left:* The proposed algorithm tracks and re-identifies a varying number of persons over long time periods despite frequent false detections and missing detections. *Right:* integration in a multi-modal person and speaker tracking and identification system.

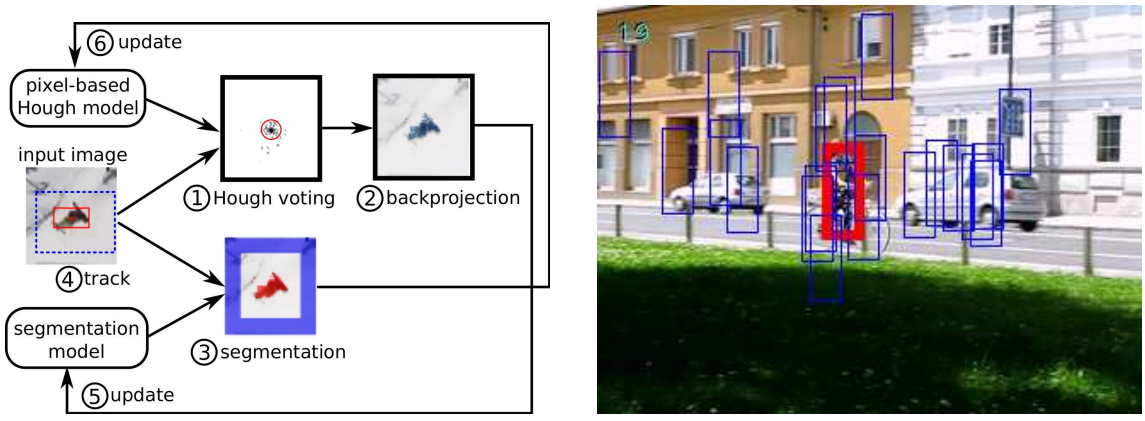


Figure 4.3: *Left:* procedure of PixelTrack. Pixel-wise detection and segmentation is performed in parallel, and the two models are jointly learnt on-line. *Right:* particle filter-based MCT approach, where a discriminative model is learnt on-line using negative examples from likely distracting regions in the scene (blue rectangles).

face tracking method [40] that takes into account different visual cues (colour, texture, shape) and dynamically adapts the inference according to an integrated reliability measure.

Later, with Prof. Garcia at LIRIS, we continued our research on visual object tracking, but on more generic methods that track arbitrary single objects under challenging conditions, *i.e.* moving camera, difficult changing lighting conditions, deforming and turning objects, with other distracting objects etc. In this context, we developed several original methods. The first one, called "PixelTrack" [5, 34], is based on a pixel-based Hough voting and a colour segmentation model and is particularly well suited for fast tracking of arbitrary deformable objects (see Fig. 4.3). The second one, called "Motion Context Tracker" (MCT) [6, 31], is based on a particle filter framework and uses a discriminative on-line learnt model to dynamically adapt to the scene context by taking into account other distracting objects in the background (see Fig. 4.3).

These different works on visual object tracking will be described in more detail in Part I of the manuscript.

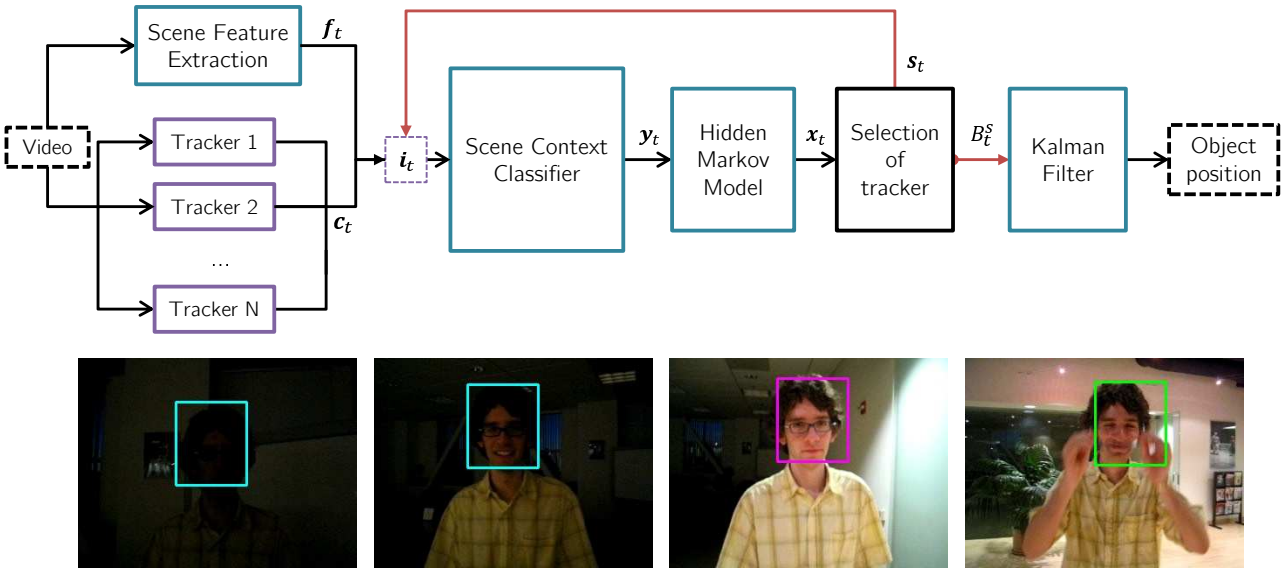


Figure 4.4: *Top:* dynamic tracker selection algorithm based on a scene context classifier. *Bottom:* example video with tracking result of the proposed method. Different colours represent different trackers.

4.3 PhD thesis supervision

My supervision activities of PhD students began in 2012 when I joined the IMAGINE team of the LIRIS laboratory and INSA Lyon as associate professor.

From 2013 to 2016, I co-supervised the thesis of **Salma Moujtahid** [283] on visual object tracking with Prof. Atilla Baskurt, and it was defended on 03/11/2016. We started from the observation and hypothesis that some tracking algorithms perform better in a given environment and others perform better in other settings. Thus, we developed a method [29] that, given a set of tracking algorithms running in parallel on a video, evaluates the confidence of each tracker and chooses the best one at each instant of the video stream according to an additional spatio-temporal coherence criterion. Further, we conceived a set of visual features that were able to quantify the characteristics of the global scene context in a video at a given time. Using these scene context features, we trained a classifier that estimates with high precision the best tracker for a given visual context over time [28] (see Fig. 4.4). Using this tracker selection algorithm, we combined several state-of-the-art trackers operating on different (complementary) visual features, and achieved an improved performance. With this approach, we also participated in the international Visual Object Tracking challenge, VOT 2015, where we obtained a good ranking among many powerful state-of-the-art methods [23].

From 2012, I also co-supervised the thesis of **Samuel Berlemont** [90] with Prof. Christophe Garcia and Dr. Grégoire Lefebvre from Orange Labs, Grenoble. This industrial thesis was defended in February 2016 and concerned the development of new algorithms to automatically recognise symbolic 3D gestures performed with a mobile device in the hand using inertial sensor data (see Fig. 4.5). We proposed a new model and a weakly supervised machine learning technique based on a Siamese neural network to learn similarities between gesture signals in a low-dimensional sub-space, achieving state-of-the-art recognition rates [2, 22, 24] compared to previously proposed approaches, including our own method based on a CNN [33]. We not

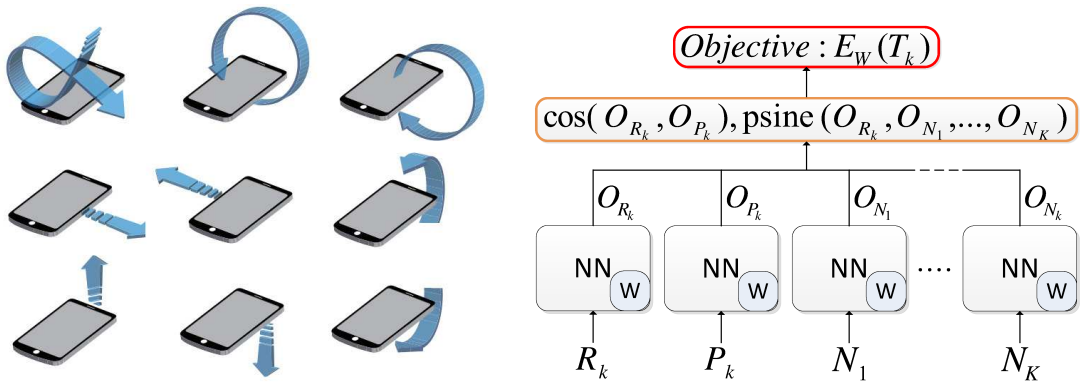


Figure 4.5: *Left:* symbolic 3D gesture recognition with inertial sensor data from mobile devices. *Right:* proposed Siamese neural network architecture learning similarities and dissimilarities from tuples of input samples and using a polar sine-based objective function.

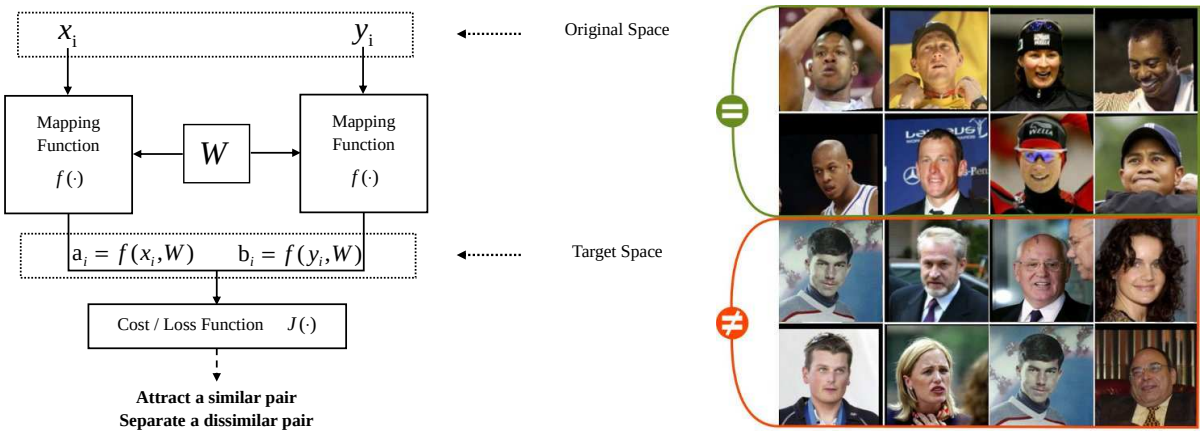


Figure 4.6: *Left:* Siamese neural network architecture learning a similarity metric in the feature space. *Right:* similar and dissimilar (face) pairs used for training (LFW dataset).

only showed an increased rejection performance for unknown gestures, but also and improved classification rate with a specific neural network structure learning from *tuples* of data samples (instead of pairs or triplets as in previous approaches) and a novel polar sine-based objective function (see Fig.4.5), which leads to a better training stability and a better modelling of the relation between similar and dissimilar training examples.

I further co-supervised the thesis of **Lilei Zheng** [441], defended on 10/05/2016, together with Atilla Baskurt, Khalid Idrissi and Christophe Garcia. In this work, we introduced several new similarity metric learning methods [4, 8, 26, 27], based on linear and non-linear projections through Siamese Multi-Layer Perceptrons, and applied them to the problem of face verification, *i.e.* given two (unknown) face images, deciding if they belong to the same (unknown) person or not (see Fig. 4.6). The first method that we proposed for training these Siamese neural network models is called Triangular Similarity Metric Learning (TSML) [4, 26] and is based on an objective function using the cosine distance and conditions the norm of the projected feature vectors, thus improving the convergence and overall performance of this learnt metric. The second method is called Discriminative Distance Metric Learning (DDML) [4] and uses the Euclidean distance and a margin to define the objective function for training the model, giving comparable performance to TSML. We further evaluated the proposed methods on the problem

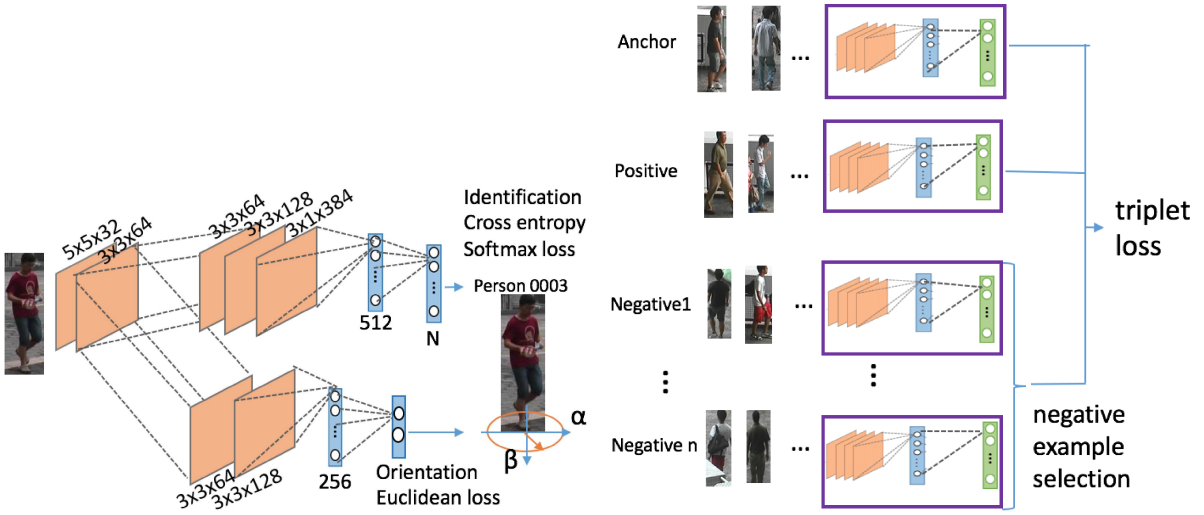


Figure 4.7: Proposed orientation-specific re-identification deep neural network. *Left:* in the first step, the two neural network branches are pre-trained separately on identity labels and body orientations (supervised learning). *Right:* in the second step, the whole neural network is fine-tuned for person re-identification (triplet learning with hard negative selection).

of speaker verification (*i.e.* audio signals) and also achieved state-of-the-art results. Finally, we applied our TSML algorithm on kinship verification [25], *i.e.* verifying parent-child relationships in images, and participated in an international competition organised at the FG conference in 2015, and our approach was ranked first in one of the sub-competitions and second in the other one.

From 2015 to 2018, I co-supervised the thesis of **Yiqiang Chen** [116] together with Prof. Atilla Baskurt and Jean-Yves Dufour from Thales ThereSIS Lab, Paris, and it was defended on the 12/10/2018. The topic of the thesis was "person re-identification in images with Deep Learning", which is a similar problem to the above-mentioned face verification, *i.e.* given two images of (unknown) persons (pedestrians), *e.g.* coming from different cameras, we want to know if they belong to the same person or not. Because the identities of the persons are generally not known before building the model and because we want the method to be as generic as possible, we tackled this problem with a similarity metric learning approach, as is commonly done in the literature. To this end, we proposed several deep learning-based approaches, where the similarity learning is generally performed with a variant of the Siamese neural network using *triplets* of examples instead of pairs, *i.e.* one reference example (or anchor), one similar example and one dissimilar example.

First, we developed a CNN-based discriminative algorithm to automatically recognise pedestrian attributes, *i.e.* semantic mid-level descriptions of persons, such as gender, accessories, clothing *etc.* [15]. These attributes are helpful to describe characteristics that are invariant to pose and viewpoint variations. Then we combined this model with another CNN trained for person identification, and the two branches are fine-tuned for the final person re-identification task [20]. Secondly, among the challenges, one of the most difficult is the variation under different viewpoints. To deal with this issue, we propose an orientation-specific deep neural network [17] (see Fig. 4.7), where one branch performs body orientation regression and steers another branch that learns separate orientation-specific layers. The combined orientation-specific CNN feature representations are then used for the final person re-identification task. Further, developed an-

other approach to include visual context into the re-identification process [16], *i.e.* for a given person query image, we additionally used the image region of the surrounding persons to compute a pairwise combined distance measure based on an location-invariant group feature representation. Finally, as a fourth contribution in this thesis, we proposed a novel listwise loss function taking into account the order in the ranking of gallery images with respect to different probe images [18]. Further, an evaluation gain-based weighting is introduced in the loss function to directly optimise the evaluation measures of person re-identification.

Currently, there are four ongoing PhD theses at LIRIS that I actively co-supervise :

- Paul Compagnon, "Prédiction de routines situationnelles dans l'activité des personnes fragiles", co-supervision with Christophe Garcia (LIRIS) and Grégoire Lefebvre (Orange Labs Grenoble),
- Thomas Petit, "Reconnaissance faciale à large échelle dans des collections d'émissions de télévision", co-supervision with Christophe Garcia (LIRIS) and Pierre Letessier (Institut National de l'Audiovisuel),
- Guillaume Anoufa, "Reconnaissance d'objets intrus en phase de vol hélicoptère par Apprentissage Automatique", co-supervision with Christophe Garcia (LIRIS) and Nicolas Bélanger (Airbus Helicopters)
- Ruiqi Dai, "Apprentissage autonome pour la vision intelligente", co-supervision with Véronique Eglin (LIRIS) and Mathieu Guillermin (Institut Catholique de Lyon)

PART I

On-line learning and applications to visual object tracking

5 Multiple object tracking in unconstrained environments

5.1 Introduction

Visual object tracking consists in following a given object in an image sequence or video over time. The object to follow is usually given, the first time it is visible in the video, by a human or an automatic detection algorithm. If a detection algorithm is used, applying it to every frame of the video is generally not an acceptable or sufficient solution as the object may not always be detected, false detections may occur, previous detections are not taken into account such that a certain continuity is lacking and, finally, running the detector may be computationally too expensive. The output of tracking algorithms is a description of the state of the object at each point in time in the video. This can be its position and scale in the image (usually described as a bounding box), or a "real-world" position (2D on the ground plane, or 3D), but also its orientation, speed or a finer description such as a parametric shape or part-based model.

Tracking can be performed *off-line* or *on-line*. In *off-line* tracking, the entire video is available at once for analysis and inference, whereas in *on-line* tracking, a video stream is analysed sequentially from the beginning, *i.e.* at each point in time only the past and present information can be used to estimate the object state but not the future. We will only consider on-line tracking in this work.

Finally, an important aspect of tracking algorithms concerns the number of objects to track. If we know that there is only one object to track, and it is visible throughout the whole video, *Single Object Tracking* (SOT) approaches are used. In that case, the algorithm is given the state (*e.g.* the bounding box) of the object in the beginning, and it is supposed to track it until the end. We will come back to SOT in chapter 6. In Multiple Object Tracking (MOT), several objects (usually of the same category) are to be followed in an image sequence, and state-of-the art tracking algorithms mostly rely on a separate object detector trained (off-line) for the given category of objects to track (*e.g.* a person, face or car detector).

This presents a certain number of inherent challenges:

- in most applications, the number of visible objects is not known *a priori*, and the algorithm needs to handle newly arriving and disappearing objects,
- objects may occlude each other partially or completely,
- the object detector may miss objects, and false detections may occur,
- the algorithm may confuse two objects, *e.g.* when they come close to each other (track ID switch).

Figure 5.1: Example video frames from the considered application; dataset 1 and 2 (top), and 3 (bottom). Faces may be difficult to detect, and occlusions can occur requiring an effective mechanism to remove and reinitialise tracks.

The linking of new detections to existing tracked objects is called "data association". The above-mentioned difficulties become more or less problematic depending on the given application context.

In our previous research, we worked on both SOT and MOT and made contributions for different scientific challenges of each of them. In the following, I will first present our work at Idiap Research Institute with Jean-Marc Odobez on multiple face tracking, where we introduced an original algorithm for long-term tracking by defining a framework for robust track creation and removal in MOT, as well as work performed with Christophe Garcia on an unsupervised incremental learning algorithm for estimating the visual focus of attention of a person in a video. Our contributions to SOT with different on-line learning approaches will be presented in chapter 6.

5.2 On-line long-term multiple face tracking

5.2.1 Introduction

The work described in the following has been performed at Idiap with Jean-Marc Odobez in the context of a large European FP7 project called TA2, where we tried to improve group-to-group communication using different technological approaches. The given context corresponds to a type of video-conferencing application where people interact with each other using a touch-table (*e.g.* playing a game) (see Fig. 5.1). In this setting, multiple faces need to be tracked robustly, in real-time and over long periods of time. This is a rather unconstrained and dynamic environment as people may enter and leave at any moment, and the room and person configuration is not fixed, *i.e.* people might be relatively far from the camera, and frequent occlusions may occur. An additional challenge for face tracking here is that the participants do not always look into the camera, and their attention might be on the touch-table or on another person in the room.

The most straightforward approach for solving the face tracking problem is to employ a face detector (*e.g.* [379]). However, despite much progress in recent years on multi-view face

detection, these methods are mostly employed in scenarios where people predominantly look towards the camera. As we demonstrate in our results, this is not sufficient for more complex scenarios, where faces are missed around $30 - 40\%$ of the time due to less common head poses. Unfortunately, the difficult head postures can last for relatively long periods of time (up to one minute in some of our videos). This means that face detection algorithms have to be complemented by robust tracking approaches; not only to interpolate detection results or filter out spurious detection, as is often assumed, but also to allow head localisation over extended periods of time.

Numerous multiple faces tracking methods have been proposed in the literature (*e.g.* [93, 142, 269, 304, 405]), mainly focusing on new features, new multi-cue fusion mechanisms, better dynamics or adaptive models for instance [58, 74, 164, 326], and results are demonstrated mostly on *short* sequences [58, 74, 164, 326].

However, very few of them address track initialisation and termination, especially in terms of performance evaluation. A face detector is often used to initialise new tracks, but how to cope with its uncertain output? A *high* confidence threshold may lead to missing an early track initialisation. Conversely, with a *low* threshold false tracks are likely to occur. Track *termination* can be even more difficult. How do we know at each point in time if a tracker is operating correctly? This is an important issue in practice, especially since an *incorrect* failure detection can lead to losing a person track for a long time until the detector finds the face again.

5.2.2 State of the art

Many existing MOT approaches operate *off-line*, *i.e.* the information from the entire video is available for the inference, or sometimes these methods are applied on sliding time windows in order to allow for a "pseudo"-on-line operation. In most of these works, data association is formulated as a global optimisation problem on a graph-based representation [88, 102, 181, 281, 309, 433]. However, these off-line algorithms are not suited for the real-time on-line setting that we investigated here.

In on-line MOT, existing approaches either employ *deterministic* methods for data association [75, 215] based on the Hungarian algorithm [286] or on a greedy approach [98] or *probabilistic* methods, like the traditional Multiple Hypothesis Tracker (MHT) [317], the Joint Probabilistic Data Association Filter (JPDAF) [149], both based on a Kalman Filter framework, and Particle Filter-based techniques [185, 202, 214, 295, 297, 412].

Most of these methods do not explicitly incorporate mechanisms for track creation and deletion, especially with longer periods of missed detections and frequent false detections, as is the case in the application scenario that is considered here. Principled methods exist to integrate track creation and termination within the tracking framework, for example Reversible-Jump Markov Chain Monte Carlo (RJ-MCMC) [214, 417]. But to be effective, they require appropriate global scene likelihood models involving a fixed number of observations (independent from the number of objects), and these are difficult to build in multi-face tracking applications.

Kalal *et al.* [205] present an interesting approach for failure detection in visual object tracking that is based on the idea that a correctly tracked target can be tracked *backwards* in time. Unfortunately, the backward tracking greatly increases the overall computational complexity (by a factor that is linear in the backward depth). In a particle filter tracking framework, another solution is to directly model a failure state as a random variable within the probabilistic model [310]. However, this increases the complexity of the model and thus the inference, and it is difficult, in practice, to model the distribution of a failure state or failure parameters. Closer to our work, Dockstader *et al.* [137] proposed to detect failure states in articulated human body

tracking using a Hidden Markov Model (HMM). However, their method differs significantly from ours: they only use one type of observation (the state covariance estimate) which in our case proves to be insufficient for assessing tracking failure; their observation are quantised to use a standard discrete multinomial likelihood model, whereas our method learns these likelihoods in a discriminative fashion; and their HMM structure (number of states, connections) is specifically designed for their articulated body tracking application.

In applications that are similar to ours, the problem of deciding when to stop tracking a face is usually solved in a recursive manner. This means, assessing tracking failure is often left to the (sudden) drop of objective or likelihood measures which are not easy to control in practice [279, 280].

In many scenarios of interest, the camera is fixed, and due to the application and the room configuration, people in front of the camera tend to behave similarly over long periods of time. However, most of the existing tracking methods ignore this long-term information, as they concentrate on video clips that are often not longer than a minute. Or if they use long-term information, it is mainly for constructing stable appearance models of tracked objects [200, 428], *e.g.* by working at different temporal scales [364]. Similarly, some methods [74, 203] train an (object-specific) detector online, during tracking, to make it more robust to short-term and long-term appearance changes. However this increases the computational complexity, because a separate model has to be built for each person, and each such detector has to be applied on the input frames. Mikami *et al.* [279] introduced the Memory-based Particle Filter where a history of past states (and appearances [280]) is maintained and used to sample new particles. However, they only addressed single, near-frontal face tracking, in high resolution videos and only evaluated the method on 30 to 60-second video clips. Other works (*e.g.* [221, 249, 347]) tackle the problem of long-term *person* tracking by analysing the statistics of features from shorter tracks (tracklets), and by proposing methods to effectively associate them. These algorithms are different from ours as they process the data *off-line*, *i.e.* the observations at each point in time are known in advance, and they mainly deal with tracking the position of the *full human body* as opposed to just faces. Another approach for multiple pedestrian tracking [86] associates smaller tracklets *on-line* and in a statistical sampling framework but no principled mechanism for starting and ending tracks is proposed. Recently, and after our work, an approach similar to ours from Xiang *et al.* [407] has been proposed, using Markov Decision Processes and re-inforcement learning for data association, and to decide on track deletion.

In the following, we will first introduce the principal framework and equations for MOT with particle filters and Markov Chain Monte Carlo (MCMC) sampling, and then describe our contributions related to the probabilistic framework of track creation and removal, long-term static and dynamic observation models as well as experimental results [12, 38].

5.2.3 Proposed Bayesian multiple face tracking approach

We tackle the problem of multi-face tracking in a recursive Bayesian framework. Assuming we have the observations $\mathbf{Y}_{1:t}$ from time 1 to t, we want to estimate the posterior probability distribution over the state $\tilde{\mathbf{X}}_t$ at time t:

$$p(\tilde{\mathbf{X}}_t|\mathbf{Y}_{1:t}) = \frac{1}{C}\, p(\mathbf{Y}_t|\tilde{\mathbf{X}}_t) \times \int_{\tilde{\mathbf{X}}_{t-1}} p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1}) p(\tilde{\mathbf{X}}_{t-1}|\mathbf{Y}_{1:t-1})\, \mathrm{d}\tilde{\mathbf{X}}_{t-1}\ , \tag{5.1}$$

where C is a normalisation constant. As closed-form solutions are usually not available in practice, this estimation is implemented using a particle filter with a Markov Chain Monte Carlo (MCMC) sampling scheme [214]. The main elements of the model are described below.

5.2.3.1 State space

We use a multi-object state space formulation, with our global state defined as $\tilde{\mathbf{X}}_t = (\mathbf{X}_t, \mathbf{k}_t)$, where $\mathbf{X}_t = \{\mathbf{X}_{i,t}\}_{i=1..M}$ and $\mathbf{k}_t = \{k_{i,t}\}_{i=1..M}$. The variable $\mathbf{X}_{i,t}$ denotes the state of face i, which comprises the position, speed, scale and eccentricity (*i.e.* the ratio between height and width) of the face bounding box. Each $k_{i,t}$ denotes the status of face i at time t, *i.e.* $k_{i,t} = 1$ if the face is visible at time t, and $k_{i,t} = 0$ otherwise. Finally, M denotes the maximum number of faces visible at a current time step.

5.2.3.2 State Dynamics

The overall state dynamics, used to predict to current state from the previous one, is defined as:

$$p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1}) \propto p_0(\mathbf{X}_t|\mathbf{k}_t) \prod_{i\in\{1..M\}|k_{i,t}=1} p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1}), \tag{5.2}$$

that is the product of an interaction prior p_0 and of the dynamics of each individual face that is visible at iteration t like in tracking methods for a fixed number of targets [214]. Note that this is actually feasible since the creation and deletion of targets are defined outside the filtering step (see next section). The position and speed components of the visible faces are described by a mixture of a first-order auto-regressive model p_a and a uniform distribution p_u, i.e., if x denotes a position and speed component vector, we have: $p(x_{i,t}|x_{i,t-1}) = \alpha p_a(x_{i,t}|x_{i,t-1}) + (1 - \alpha)p_u(x_{i,t}|x_{i,t-1})$, with $p_a(x_{i,t}|x_{i,t-1}) = \mathcal{N}(Ax_{t-1}; 0, \mathbf{\Sigma})$, and $p_u(x_{i,t}|x_{i,t-1})) = c$ with c being a constant allowing for small "jumps" coming from face detection proposals (see Eq. 5.8). A first order model with steady-state is used for the scale and eccentricity parameters. If x denotes one such component: $(x_t - SS) = \mathcal{N}(a(x_{t-1} - SS); 0, \sigma_{SS})$, where SS denotes the steady-state value. The steady-state values for scale and eccentricity are updated only when a detected face is associated with the face track and at a much slower pace compared to the frame-to-frame dynamics.

The interaction prior p_0 is defined as

$$p_0(\mathbf{X}_t|\mathbf{k}_t) = \prod_{\{i,j\}\in\mathcal{P}} \phi(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) \propto \exp\Big\{-\lambda_g \sum_{\{i,j\}\in\mathcal{P}} g(\mathbf{X}_{i,t}, \mathbf{X}_{j,t})\Big\}, \tag{5.3}$$

preventing targets to become too close to each other. The set $\mathcal{P}$ consists of all possible pairs of objects that are visible. The penalty function $g(\mathbf{X}_{i,t}, \mathbf{X}_{j,t}) = \frac{2a(B_i \cap B_j)}{a(B_i)+a(B_j)}$ is the intersection area as a fraction of the average area of the two bounding boxes B_i and B_j defined by $\mathbf{X}_{i,t}$ and $\mathbf{X}_{j,t}$, where $a(.)$ denotes the area operator. The factor λ_g controls the strength of the interaction prior (set to 5 in our experiments).

5.2.3.3 Observation Likelihood

As a trade-off between robustness and computational complexity, we employ a relatively simple but effective observation likelihood for tracking. Another model could be used as well.

Given our scenario, we assume that the face observations $\mathbf{Y}_{i,t}$ are conditionally independent given the state, leading to an observation likelihood defined as the product of the visible individual faces likelihoods:

$$p(\mathbf{Y}_t|\tilde{\mathbf{X}}_t) = \prod_{i|k_{i,t}=1} p(\mathbf{Y}_{i,t}|\mathbf{X}_{i,t}). \tag{5.4}$$

Note that we did not include a partial (or full) overlap model in the likelihood component, nor any other contextual tracking techniques [414]. Strong overlaps are prevented explicitly by the interaction term (Eq. 5.3) in the dynamics. This approach is appropriate for our scenarios (teleconference, HCI/HRI), where continuous partial face occlusions happen only rarely. More often, faces are occluded by other body parts that are not followed by the tracker, like a person's hand, or another person's body crossing in front. Even a joint likelihood model would not handle these cases. Thus, for longer full occlusions, our strategy is to have the algorithm remove the track of the occluded face, and restart it afterwards as soon as possible.

The observation model for a face i is based on $R = 6$ HSV colour histograms $\mathbf{Y}_{i,t} = [h(r, \mathbf{X}_{i,t})]$ $(r = 1..R)$, that are computed on the face region described by the state $\mathbf{X}_{i,t}$. They are compared to histogram models $h^*_{i,t}(r)$, to define the observation likelihood for a tracked face as follows:

$$p(\mathbf{Y}_{i,t}|\mathbf{X}_{i,t}) \propto \exp(-\lambda_D \sum_{r=1}^{6} \left(D^2[h^*_{i,t}(r), h(r, \mathbf{X}_{i,t})]\right) - D_0)\,, \tag{5.5}$$

where D denotes the Euclidean distance , $\lambda_D = 20$, and D_0 is a constant offset defining the distance at which the likelihood in Eq. (5.5) gives 1.0. More precisely, we divided the face into three horizontal bands and in each band computed two normalised histograms with two different levels of quantisation. Specifically, we used the scheme proposed in [304] which decouples coloured pixels (put into $N_b \times N_b$ HS bins) from grey-scale pixels (N_b separate bins) and applied it with two different quantisation levels, $N_b = 8$ and $N_b = 4$ bins per channel. This choice of semi-global multi-level histograms results from a compromise between speed, robustness to appearance variations across people as well as head pose variations for individuals, and a well conditioned likelihood, *i.e.* peaky enough to accept a well identified optimum, but with a smooth basin of attraction towards this optimum, adapted to low sampling strategies.

The histogram models of one face are initialised when a new target is added to the tracker. Furthermore, to improve the tracker's robustness to improper initialisation and changing lighting conditions, they are updated whenever a detected face is associated with the given face track (see below):

$$h^*_{i,t}(r) = (1 - \epsilon)h^*_{i,t-1}(r) + \epsilon h^d_{i,t}(r) \quad \forall r\,, \tag{5.6}$$

where $h^d_{i,t}$ denotes the histograms from the detected face region, and ϵ is the update factor (set to 0.2 in our experiments).

5.2.3.4 Tracking algorithm

At each time instant, the tracking algorithm proceeds in two main stages: first, recursively estimate the states of the currently visible faces relying on the model described above and solved using an MCMC sampling scheme. Second, make a decision on adding a new face or on deleting currently tracked faces. This second stage is described in Section 5.2.4. The MCMC sampling scheme allows for efficient sampling in this high-dimensional state space of interacting targets, and follows the method described in [214].

Let N be the total number of particles and N_{bi} the number of "burn-in" particles. At each tracking iteration, we do:

1. initialise the MCMC sampler at time t with the sample $\tilde{\mathbf{X}}^{(0)}_t$ obtained by randomly selecting a particle from the set $\{\tilde{\mathbf{X}}^{(s)}_{t-1}, s = (N_{bi} + 1) \ldots N\}$ at time $t - 1$ and sample the state of every visible target i using the dynamics $p(\mathbf{X}_{i,t}|\mathbf{X}_{i,t-1})$ (deleted targets are ignored);

2. sample iteratively N particles from the posterior distribution of (5.1) using the Metropolis-Hastings algorithm:

   (a) sample a new particle $\tilde{\mathbf{X}}_t{}'$ from a proposal distribution $q(\tilde{\mathbf{X}}_t{}'|\tilde{\mathbf{X}}_t{}^{(s)})$ (described below);

   (b) compute the acceptance ratio:

$$a = \min\left(1, \frac{p(\tilde{\mathbf{X}}_t{}'|\mathbf{Y}_{1:t})\, q(\tilde{\mathbf{X}}_t{}^{(s)}|\tilde{\mathbf{X}}_t{}')}{p(\tilde{\mathbf{X}}_t{}^{(s)}|\mathbf{Y}_{1:t})\, q(\tilde{\mathbf{X}}_t{}'|\tilde{\mathbf{X}}_t{}^{(s)})}\right) \tag{5.7}$$

   (c) accept the particle (*i.e.* define $\tilde{\mathbf{X}}_t{}^{(s+1)} = \tilde{\mathbf{X}}_t{}'$) with probability a. Otherwise, add the old particle (*i.e.* set $\tilde{\mathbf{X}}_t{}^{(s+1)} = \tilde{\mathbf{X}}_t{}^{(s)}$)

After time step t, the particle set $\{\tilde{\mathbf{X}}_t^{(s)}\}_{s=N_{bi}+1}^{N}$ represents an estimation of the posterior $p(\tilde{\mathbf{X}}_t|\mathbf{Y}_{1:t})$.

The proposal function $q(\cdot)$ allows for selecting good candidates for the particle set. Efficiency in MCMC sampling is obtained by modifying object states one at a time. More precisely, a new sample is selected by letting $\tilde{\mathbf{X}}_t{}' = \tilde{\mathbf{X}}_t{}^{(s)}$, randomly select a face i amongst the visible ones, and then sample the proposed state $\mathbf{X}'_{i,t}$ of face i from:

$$q(\mathbf{X}'_{i,t}|\tilde{\mathbf{X}}_t) = \left[(1-\alpha)\frac{1}{N-N_{bi}}\sum_r p(\mathbf{X}'_{i,t}|\mathbf{X}^{(s)}_{i,t-1}) + \alpha p(\mathbf{X}'_{i,t}|\mathbf{X}^d_t)\right] \tag{5.8}$$

that is a mixture of the state dynamics (ensuring temporal smoothness) and the output of a face detector (avoiding tracker drift) controlled by the factor α, where $\mathbf{X}^d_t$ denotes the state of the closest detection coming from a face detector [379] and associated with face i. Again, targets removed at the previous step are ignored, while recently added targets are simply sampled around their initial position.

5.2.4 Target creation and removal

The way objects are added and removed from the tracker is a key feature of the algorithm that we proposed. In our application scenario, the goal is to avoid false alarms as much as possible. This means, the tracker should be able to detect as quickly as possible if there is a tracking failure. On the other hand, it should not stop tracking when there is no failure since it may take a long time until the object is detected again.

We proposed to use two different Hidden Markov Models (HMM) for that purpose, as described in the following sections. One is used for object creation and the other for object removal. Each of them receives different types of observations.

A face detector (for both frontal and profile views) is called every 10 frames (*i.e.* around twice per second, as our algorithm is able to process around 20-23 frames/s in real-time). The HMMs are updated only at these instants, but rely on observations computed on all frames since the last update. According to our experiments, applying the detector to every video frame did not significantly improve the tracking performance and considerably slowed down the algorithm.

Before the creation and removal step, each detection is associated to a track provided the following conditions hold:

1. the detection is not associated with any other target,

2. it has the smallest distance to the tracked target,

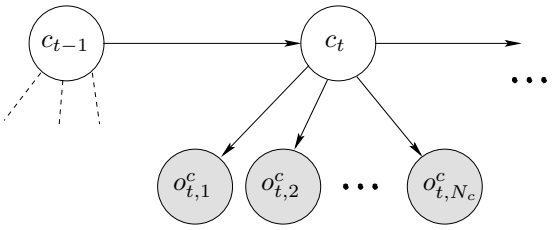


Figure 5.2: The HMM used at each image position for tracker target creation. The variable c_t indicates a face centred at a particular image position. The probability of c_t is estimated recursively using the observations $o^c_{t,1}..o^c_{t,N_c}$.

3. the distance between detection and target is smaller than two times the average width of their bounding boxes,

4. the two bounding boxes overlap.

Although a more generic way would be to use training data to learn the association rules and parameters as done in [322], for instance, the above conditions work well for our data in the large majority of cases.

In the following, we describe the HMMs for target creation and removal. Note that naturally, only un-associated detections are considered for the initialisation of a new target.

5.2.4.1 Creation

When initialising a new target we have two objectives: first, minimise erroneous initialisations due to false detections, and second, initialise correct targets as early as possible.

For deciding when to add new targets to the face tracker, we propose a simple HMM that estimates the probability of a hidden, discrete variable $c_t(i,j)$ indicating at each image position (i,j) if there is a face or not at this position. Figure 5.2 illustrates the model. In the following, we drop the indices (i,j) for clarity. Let us denote by $\mathbf{O}^c_t = [o^c_{t,1}, \ldots, o^c_{t,N_c}]$ the set of N_c observations at each time step t, and by $\mathbf{O}^c_{1:t} = [\mathbf{O}^c_1, \ldots, \mathbf{O}^c_t]$ the sequence of observations from time 1 to time t. Assuming the transition matrix is defined as: $p(c_t|c_{t-1}) = 1$ iff $c_t = c_{t-1}$ and 0 otherwise, the posterior probability of the state c_t can be recursively estimated as:

$$p(c_t = s|\mathbf{O}^c_{1:t}) = \frac{p(\mathbf{O}^c_t|c_t = s)\, p(c_{t-1} = s|\mathbf{O}^c_{1:t-1})}{\sum_{s'} p(\mathbf{O}^c_t|c_t = s')\, p(c_{t-1} = s'|\mathbf{O}^c_{1:t-1})}\,, \tag{5.9}$$

where

$$p(\mathbf{O}^c_t|c_t) = \prod_{i=1}^{N_c} p(o^c_{t,i}|c_t)\,. \tag{5.10}$$

Track creation: for each detected face that is not associated with any current face target, we decide whether a track is created or not. To this end, if (i,j) denotes the centre position of the face detection, the ratio:

$$r^c_t(i,j) = \frac{p\left(c_t(i,j) = 1|\mathbf{O}^c_{1:t}(i,j)\right)}{p\left(c_t(i,j) = 0|\mathbf{O}^c_{1:t}(i,j)\right)} \tag{5.11}$$

is computed. If $r^c_t(i,j) > 1$, then a new track is initialised at (i,j). Otherwise, no track is created from the given detection.

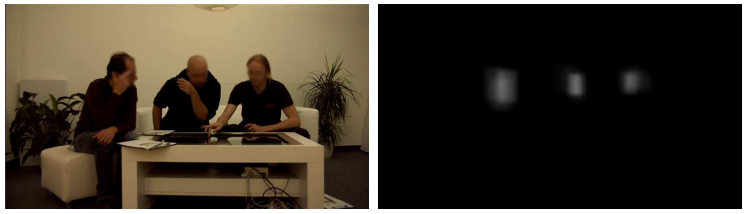


Figure 5.3: Example image (left) with an illustration of the corresponding tracking memory (right) during tracking. Qualitatively speaking, track creation will be faster (almost immediate) when a new face detection is observed in the "white" regions whereas repetitive detection will be needed to initiate a track in a "black" region. Similarly, when an object track moves to black regions, its failure probability will become higher. See text for details.

Observations and likelihood models: we propose to use two different types of observations: $o^c_{\cdot,1}$, the output of the face detector and $o^c_{\cdot,2}$, a long-term "memory" of the states (*i.e.* positions) of tracked faces $\mathbf{X}_t$.

The first observation is defined as follows. At time t and image position (i, j) we set:

$$o^c_{t,1} = \begin{cases} 1 & \text{if } (i,j) \text{ is covered by one of the bounding boxes of the detected faces,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.12}$$

The likelihood of the first observation is then defined as

$$\begin{aligned} p(o^c_{t,1} = 0|c_t = 0) &= 1 - fa, & p(o^c_{t,1} = 1|c_t = 0) &= fa, \\ p(o^c_{t,1} = 0|c_t = 1) &= md, & p(o^c_{t,1} = 1|c_t = 1) &= 1 - md\ , \end{aligned} \tag{5.13}$$

where fa is the empirical false alarm rate and md the missed detection rate of the detector. According to our detection results from several datasets, we set $fa = 0.0001$ and $md = 0.4$.

The second observation $o^c_{t,2}$ is based on the history of past image positions of tracked faces, which we will call "tracking memory" in the following. At each iteration of the tracker, the tracking memory is updated slowly according to the mean of the current state distribution $\bar{\mathbf{X}}_t$:

$$o^c_{t,2} = (1 - \beta)o^c_{t-1,2} + \beta I_t\ , \tag{5.14}$$

where $\beta = 0.001$ and

$$I_t(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ is covered by one of the bounding boxes described by } \bar{\mathbf{X}}_t, \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

Figure 5.3 shows an example of the tracking memory during a run of the face tracker. Intuitively, we would like to initialise targets more quickly in regions where a person has been "seen" previously. Thus, we model $p(o^c_{t,2}|c_t)$ with a pair of sigmoid functions:

$$p(o^c_{t,2}|c_t = 1, \Theta) = \frac{1}{\pi}\arctan(\delta_l(o^c_{t,2} - \mu_l)) + \frac{1}{2} \tag{5.16}$$

$$p(o^c_{t,2}|c_t = 0, \Theta) = 1 - p(o^c_{t,2}|c_t = 1)\ , \tag{5.17}$$

where the parameters $\Theta_l = (\mu_l, \delta_l)$, denote the offset and the slope of the sigmoid (see Fig. 5.4). Intuitively, the offset μ_l denotes the threshold value beyond which an observation $o^c_{t,2}$ is more likely to occur within the bounding box of a detected face than in a non-face area, whereas the slope controls how fast the likelihood change is around this threshold.

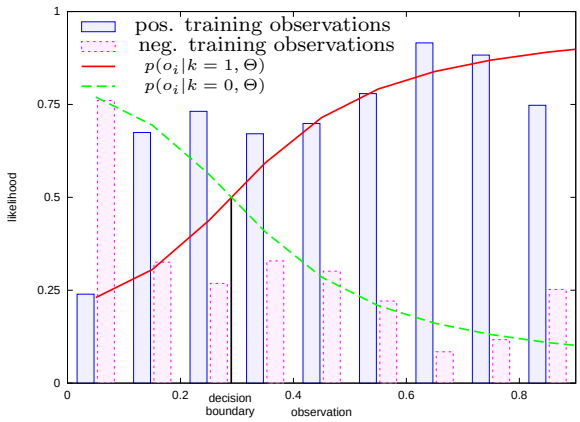


Figure 5.4: Example of an observation likelihood model (here o_3^r) described by a pair of sigmoid functions with learnt parameters $\Theta = \{\delta, \mu\}$. The parameters of the positive sigmoid function (solid red curve) have been optimised to model best the positive (blue solid boxes) and negative (purple dotted boxes) training observations (here illustrated by histograms) according to Eq. 5.18. The offset μ (here at $x = 0.29$), where the two sigmoid paired curves cross, defines a soft decision boundary.

Parameter learning: the parameters $\Theta_l = (\delta_l, \mu_l)$ of the sigmoid functions in equations 5.16 and 5.17 have been trained offline with a set of $N^{\pm}$ observations o_i. These observations are tracking memory values that have been collected from real tracking sequences and are composed of N^+ positive instances measured at image positions of correct face detections, and N^- negative instances measured at image positions of false detections. To train the model, we maximise the posterior probability of the labels for the given observations o:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \prod_{i=1}^{N^{\pm}} p(c = C_i|o_i, \Theta), \tag{5.18}$$

where $C_i \in \{0, 1\}$ denotes the class label of o_i, and $p(c = c_i|o_i, \Theta) \propto p(o_i|c = c_i, \Theta)$ is given by Eq. 5.16 and 5.17, and we assumed an equal prior on both classes. In practise, we find a good approximation of Θ^* by doing a grid search in a reasonable range over the parameter space Θ. Figure 5.4 shows an example of a pair of learnt sigmoid functions and the respective decision boundary (in this case for target removal).

If observations are greater than μ, the ratio $\frac{p(o_i|c=1)}{p(o_i|c=0)} > 1$, that means a face is more likely to be present. Otherwise, it is more likely that no face is present.

5.2.4.2 Removal

During tracking, we want to assess at each point in time if the algorithm is still correctly following a face or if it has lost track. The algorithm can lose track, for example, when it gets distracted by a similar background region or when a person leaves the scene. More concretely, the objective is to interrupt the tracking as soon as possible if a failure occurs, and to continue tracking otherwise, even when a face has not been detected and associated with the track for a long time.

In a way similar to target initialisation, we propose to use for each tracked face i an HMM estimating at each time step t the hidden status variable $k_{i,t}$ indicating correct tracking ($k_{i,t} = 1$)

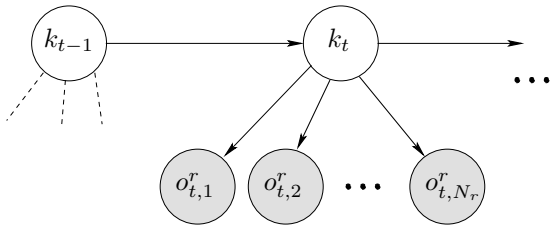


Figure 5.5: The HMM for target removal, used for each tracked face. The variable k_t indicates if a given face is still tracked correctly or if a failure occurred. The probability of k_t is estimated recursively using the observations $o^r_{t,1}..o^r_{t,N_r}$.

or tracking failure ($k_{i,t} = 0$). We will drop the face index i in the following. Figure 5.5 illustrates the proposed model.

Let us denote by $\mathbf{O}^r_t = [o^r_{t,1}, \ldots, o^r_{t,N2}]$ the set of N_r observations at each time step t, and by $\mathbf{O}^r_{1:t} = [\mathbf{O}^r_1, \ldots, \mathbf{O}^r_t]$ the sequence of observations from time 0 to time t. The posterior probability of k_t can be recursively estimated as:

$$p(k_t|\mathbf{O}^r_{1:t}) = \frac{\sum_{k'_{t-1}} p(\mathbf{O}^r_t|k_t)\, p(k_t|k'_{t-1})\, p(k'_{t-1}|\mathbf{O}^r_{1:t-1})}{\sum_{k'_t,k'_{t-1}} p(\mathbf{O}^r_t|k'_t)\, p(k'_t|k'_{t-1})\, p(k'_{t-1}|\mathbf{O}^r_{1:t-1})}\,, \tag{5.19}$$

where

$$p(\mathbf{O}^r_t|k_t) = \prod_{i=1}^{N_r} p(o^r_{t,i}|k_t)\ . \tag{5.20}$$

The state transition probability $p(k_t|k_{t-1})$ is set to 0.999 for staying in the same state and 0.001 for changing state, assuming a frame rate of approximately 20 frames per second as in our experiments.

Track ending: for each tracked face and at each time step, we compute the ratio:

$$r^k_t = \frac{p(k_t = 1|\mathbf{O}^r_{1:t})}{p(k_t = 0|\mathbf{O}^r_{1:t})}\ . \tag{5.21}$$

If $r^k_t < 1$ for a given face, then the tracking is considered to have failed and the target is removed.

Observations and likelihood models: we propose to use $N_r = 7$ different types of observations $\mathbf{O}^r_t = [o^r_{t,1}, \ldots, o^r_{t,7}]$ extracted from the image as well as the state of the tracker itself. We can divide them into two categories:

- four *static* observations $(o^r_{t,1}, \ldots, o^r_{t,4})$ that provide indications on the *state of the tracker*, and
- three *dynamic* observations $(o^r_{t,5}, \ldots, o^r_{t,7})$ that provide indications on the *temporal evolution* and variability of certain observations.

Except for one observation, all likelihoods are modelled by pairs of sigmoid functions:

$$p(o^r_{t,i}|k_t = 1, \Theta) = a_i \arctan(\delta_i(o^r_{t,i} - \mu_i)) + \frac{1}{2}, \tag{5.22}$$

$$p(o^r_{t,i}|k_t = 0, \Theta) = 1 - p(o^r_{t,i}|k_t = 1), \tag{5.23}$$

where, as for target creation observations (section 5.2.4.1), the amplitude a_i is set to $\frac{1}{\pi}$ (or $-\frac{1}{\pi}$ for some observation types), and the parameters $\Theta_i = (\delta_i, \mu_i)$, *i.e.* the slope and the offset of the

sigmoid, have been trained offline with a set of positive and negative observations as described at the end of Section 5.2.4.1. The only difference is that the training observations are collected at each time instant during tracking runs and not only when faces are detected. Below, we describe each observation we have used and comment on the learnt parameters.

Static observations: the first static observation for a given target is based on the output of the face detector:

$$o^r_{t,1} = \begin{cases} 1 & \text{if a detection is associated with the target} \\ 0 & \text{otherwise.} \end{cases} \tag{5.24}$$

The likelihood $p(o^r_{t,1}|k_t)$ is defined in the same way as for $o^c_{t,1}$ in Eq. 5.13 (that is, $p(o^r_{t,1} = u|k_t = l) = p(o^c_{t,1} = u|c_t = l)$).

The second observation $o^r_{t,2}$ is the tracking memory value at the respective target position (m, n) in the image, as defined in the previous section (Eq. 5.14 and 5.15):

$$o^r_{t,2} = o^c_{t,2}(m, n) \; . \tag{5.25}$$

This ensures that the tracking of a face is more likely to be maintained if a face stays at its previous position (with a high tracking memory value). And conversely, the target should be removed with a higher probability when it moves to image regions that were never occupied by a face before.

The third observation type is the tracker observation likelihood computed at the mean state value $\bar{\mathbf{X}}_{i,t}$ of target i:

$$o^r_{t,3} = p(\mathbf{Y}_{i,t}|\bar{\mathbf{X}}_{i,t}) \, , \tag{5.26}$$

as defined by Eq. 5.5. The likelihood $p(o^r_{t,3}|k_t)$ is again defined by a pair of sigmoids (Eq. 5.22 and 5.23).

The fourth observation relates to the variance of the target filtering distribution. More precisely, let $\sigma^2_{i,t,x}$ and $\sigma^2_{i,t,y}$ be the variances of the horizontal and vertical position of target state $\mathbf{X}_{i,t}$. Then we define

$$o^r_{t,4} = \max(\sigma^2_{i,t,x}, \sigma^2_{i,t,y}). \tag{5.27}$$

A higher variance of the state distribution means a higher uncertainty (and vice versa), and the track should be stopped more quickly.

Dynamic observations: the three remaining observations are based on the temporal variation of different features. They rely on the detection of rapid increases or decreases over time of particle variance and observation likelihood. To this end, we assume that the values of these features are normally distributed during tracking, and we use the Page-Hinckley test [82] to detect jumps or drops of these (one-dimensional, Gaussian) "signals" with respect to their means. This test works as follows: let ω_t be the signal for which we want to detect an abrupt *decrease*. Then, the following values are computed at each iteration t:

$$M_{\omega,t} = M_{\omega,t-1} + \left(\omega_t - (\bar{\omega}_t - \frac{j_\omega}{2})\right) \tag{5.28}$$

$$m_{\omega,t} = \max(m_{\omega,t-1}, M_{\omega,t}) \tag{5.29}$$

$$\hat{m}_{\omega,t} = m_{\omega,t} - M_{\omega,t} \; , \tag{5.30}$$

where $M_{\omega,0} = 0$, j_ω is a constant that determines the tolerated change of value ω, and $\bar{\omega}_t$ is the running average of ω. $M_{\omega,t}$ accumulates the values going above the expected lower bound $(\bar{\omega} - j_\omega)$. The value $m_{\omega,t}$ memorises the maximum value of this cumulative sum, and the difference between these last two values $\hat{m}_{\omega,t}$ (Eq. 5.30) is an indication of an abrupt decrease of

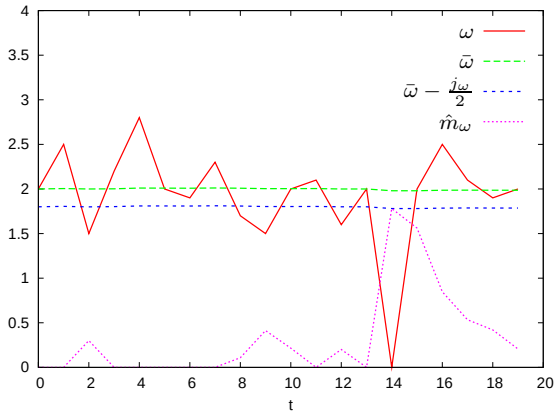


Figure 5.6: Illustration of the Page-Hinckley test to detect abrupt decreases of a signal. The solid red line shows the temporal evolution of some signal, the purple dotted line shows the computed result of the Page-Hinckley test that we use as observation to detect abrupt signal drops. At time $t = 14$ a drop of the signal occurs and is correctly detected (peak of purple dotted line).

the value ω. On the other hand, if ω_t decreases only *gradually*, then the running average $\bar{\omega}_t$ will follow this decrease. The cumulative sum $M_{\omega,t}$ will constantly increase, leading to $m_{\omega,t} = M_{\omega,t}$ and thus $\hat{m}_{\omega,t} = 0$. Figure 5.6 illustrates the Page-Hinckley test with some example data.

Similarly, for detecting an abrupt *increase* of ω we compute:

$$U_{\omega,t} = U_{\omega,t-1} + \left(\omega_t - (\bar{\omega}_t + \frac{j_\omega}{2})\right) \tag{5.31}$$

$$u_{\omega,t} = \min(u_{\omega,t-1}, U_{\omega,t}) \tag{5.32}$$

$$\hat{u}_{\omega,t} = U_{\omega,t} - u_{\omega,t} \ , \tag{5.33}$$

where $U_{\omega,0} = 0$. In its original form, the Page-Hinckley test produces a binary output. It is one if $\hat{m}_{\omega,t}$ or $\hat{u}_{\omega,t}$ is above a predefined threshold and zero otherwise. Here, we propose to directly use the values $\hat{m}_{\omega,t}$ or $\hat{u}_{\omega,t}$ as observations.

Thus, using equations 5.28-5.33, we define:

$$o^r_{t,5} = \hat{m}_{o^r_t,3} \qquad o^r_{t,6} = \hat{m}_{o^r_t,4} \qquad o^r_{t,7} = \hat{u}_{o^r_t,4} \ . \tag{5.34}$$

Observations $o^r_{t,5}$ indicate drops of the likelihood $p(\mathbf{Y}_t|\bar{\mathbf{X}}_t)$ of a given face (see 5.26). And $o^r_{t,6}$, $o^r_{t,7}$ indicate abrupt decreases and increases of the variance of the state distribution defined in 5.27. The likelihood functions $p(o^r_{t,5}|k_t), p(o^r_{t,6}|k_t)$, and $p(o^r_{t,7}|k_t)$ are defined by pairs of sigmoids (Eq. 5.22, 5.23) with parameters trained offline.

5.2.5 Re-identification

Our algorithm further tries to keep track of the identities of different persons and associates each track with a person, *i.e.* for each new target track it decides if it belongs to a previously seen person or if it is an new person. In this work, we built person models which are longer-term descriptions of person appearance acquired from observations during the tracking process. Here, a simple colour-based model, similar to [364] has been used. More specifically, the model $P_{j,t}$ of a person j is composed of two colour histograms: one describing the face region, $h^f_{j,t}$, and one

for the shirt, $h^s_{j,t}$. The structure of the histograms is similar to the one used for the observation likelihood in the tracking algorithm (5.2.3.3), *i.e.* two different quantisation levels and decoupled colour and grey-scale bins.

If a target is added to the tracker and there is no stored person model that is un-associated, a new model is initialised immediately and associated to the target. Otherwise, the face and shirt histograms ($h^f_{i,t}$, $h^s_{i,t}$) of the new target i are computed recursively over r successive frames and stored in $P^*_{i,t}$. After this period, we calculate the likelihood of each stored model $P_{j,t}$ given an unidentified candidate $P^*_{i,t}$:

$$p(P_{j,t}|P^*_{i,t}) = \exp\left(-\lambda(w_f D^2[h^f_{j,t}, h^f_{i,t}] + w_s D^2[h^s_{j,t}, h^s_{i,t}])\right), \tag{5.35}$$

where D is the Euclidean distance, and the weights are $w_f = 1, w_s = 2$. A given person i is then identified by simply determining the model $P_{m,t}$ with the maximum likelihood:

$$m = \underset{j}{\operatorname{argmax}}\, p(P_{j,t}|P^*_{i,t}) \,, \tag{5.36}$$

provided that $p(P_{m,t}|P^*_{i,t})$ is above a threshold θ (we chose 0.1 here). If not, a new person model is created and added to the stored list. All associated person models are updated at each iteration with a small factor $\alpha^p = 0.01$. The candidate models are updated with factor $\alpha^* = 0.1$.

5.2.6 Experiments

Experiments have been conducted on more than 9 hours of video data that have been annotated extensively. We used three sets of videos recorded in different environments (see Fig. 5.1). According to our scenario, recorded people have been sitting at a table and filmed by a central camera (roughly 2-3 metres away). They have been playing online games with people in a remote location using a laptop or touch-screen. As a result, they are often looking downwards and their faces are often not detected by a standard detector [379].

The principal performance measures are precision and recall (over time) of the face tracking result, as we want to track faces as long as possible (to obtain a high recall) and stop tracking as soon as a failure occurs (to increase the precision). The recall and false positive rate for an entire video are defined as:

$$R = \frac{\sum_{i=2}^{G} \delta_i d_i}{\sum_{i=2}^{G} \delta_i} \,, \qquad\qquad FP = \frac{\sum_{i=2}^{G} \delta_i f_i}{\sum_{i=2}^{G} \delta_i} \,, \tag{5.37}$$

where G is the number of annotated frames, d_i the proportion of correctly tracked/detected faces in frame i (*i.e.* those for which $F > 0.1$), f_i is the number of false positive outputs divided by the number of ground truth objects in frame i, and δ_i is the time difference between frame i and $i-1$.

We also measure the total number of interruptions for a given dataset. An interruption is defined as the event when a track is falsely ended, *i.e.* the face (ground truth) is still present but the respective target is removed from the tracker.

Finally, to measure the accuracy of identification as described section 5.2.5, we computed the *object purity* [345] for each ground truth object:

$$OP = \frac{\sum_{i=2}^{G} \delta_i q_i}{\sum_{i=2}^{G} \delta_i} \,, \tag{5.38}$$

where G is again the number of annotated frames, and q_i is the proportion of correctly identified faces in frame i, as explained in the following. The identity assigned to a ground truth object at time i is given by the algorithm described in 5.2.5 and more specifically Eq. 5.36. Once the tracking has been run on a complete video, we can compute the above rate by associating to each object the face track that has the longest overlap with the object (according to the F-measure).

We compared our results against a standard face detector [379] including models for frontal and profile views, with two competitive baselines:

- *RJ-MCMC*: a tracker based on the Reversible-Jump Monte Carlo Markov Chain algorithm [214, 417]. In addition to the *Update* move, which follows the MCMC description in section 5.2.3.4, four other moves have been implemented to handle the creation and removal of targets: *Add, Delete, Stay*, and *Leave*. For more details, we refer the reader to [214, 417].

- *MCMC baseline:* an MCMC-based tracker, *i.e.* the algorithm described in section 5.2.3. For target creation and removal, the following strategy has been used: every (un-associated) face detection is initialised as a new target. We also tried to initialise a target only after *several* successive detections but this didn't have a large impact on the precision. A tracked target was removed if it had no associated detections for 100 frames (8 seconds) or if the likelihood dropped below 10% of its running average.

Table 5.1 compares the algorithms for a given face detector threshold. The proposed method outperforms the others for all three datasets. Since the tracking algorithm for the MCMC baseline is the same as for the proposed method, the performance improvement is clearly due to our proposed target creation and removal mechanisms. The precision of RJ-MCMC is rather low because the creation and removal of targets is only based on the observation likelihood, as in [214]. Note that, unlike our approach, RJ-MCMC adds and removes targets at the particle level. Although this is a principled statistical framework that models at each point in time the current belief on the number of visible targets, it is more difficult to capture longer-term dynamics and features from the state distribution itself. The MCMC baseline, on the other hand, adds and removes targets based on more efficient, longer-term observations, namely the likelihood with respect to its mean and the face detector output. Thus, its performance is better than the one of RJ-MCMC. Also, the total number of tracker interruptions is decreased. This means that the proposed method maintains face tracks longer, even when the face detector provides no output for extended periods of time or when the likelihood is temporarily decreasing.

Figure 5.7 shows some tracking results of a video from dataset 3 containing 3-4 persons. The people change their seats from time to time, occlusions occur, and head poses can be challenging, as illustrated in the example.

More explanation and results can be found in [12, 38].

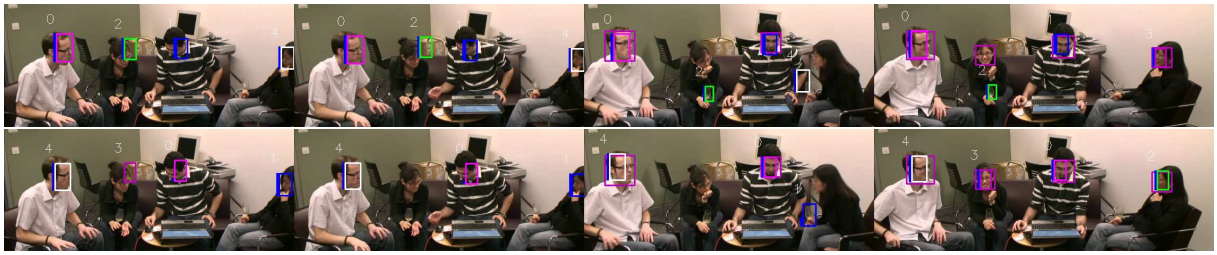
5.2.7 Conclusion

We have proposed a Bayesian framework for long-term, on-line MOT including a learnt track creation and removal approach to explicitly handle inherent weaknesses of object detection algorithms. Our experiments showed that the precision and recall with our tracking algorithm is considerably increased. We want to note that we also successfully applied our algorithm to long-term SOT, as well as to other scenarios related to Human-Robot Interaction, involving several people interacting with a robot [12]. Further, other existing algorithms could largely benefit from our track creation and removal framework, when applied on a longer time scale. Also, in the future, other types of long-term cues could be included in the proposed framework,

data set		face detection	RJ-MCMC	MCMC baseline	ours
1	recall	55.0%	89.5%	85.2%	**93.9**%
	FP rate	2.00%	20.75%	4.29%	**1.45**%
	# interrupt.	–	861	395	**112**
	average OP	–	41.35%	68.69%	**68.98**%
2	recall	39.9%	75.7%	69.9%	**76.0**%
	FP rate	0.41%	3.27%	1.21%	**0.77**%
	# interrupt.	–	2062	1004	**567**
	average OP	–	49.09%	66.61%	**69.60**%
3	recall	48.3%	77.2%	75.1%	**93.7**%
	FP rate	0.33%	18.2%	**1.06**%	1.19%
	# interrupt.	–	1299	455	**166**
	average OP	–	27.96%	34.23%	**57.46**%

Table 5.1: Performance comparison with the proposed multiple face tracking framework on the three datasets.

(a) frame 20000 (b) frame 20038 (c) frame 52408 (d) frame 114870



(e) frame 114948 (f) frame 114962 (g) frame 115148 (h) frame 115302

Figure 5.7: Snapshots of tracking results on dataset 3. Different coloured rectangles represent different identities (purple: face detector). *Top:* MCMC baseline, *bottom:* proposed approach. With the baseline method, some target are initialised from false detections 5.7(b), 5.7(g), and tracks are not maintained when detections are missing 5.7(c). Our approach avoids false initialisations and maintains good tracks longer. In 5.7(f) failures are detected earlier, and in 5.7(h), the lost target is re-initialised earlier (second person from the left).

Figure 5.8: Graphical illustration of VFOA estimation in one of the investigated settings (a meeting room). Targets 1 to 4 are persons, 5 corresponds to the table.

There are still some limitations of the proposed framework. Even if we are able to accurately and quickly detect failures, the result is still dependent on the time it takes to obtain a new face detection to reinitialise the tracking. Thus, having less tracking failures and better multi-view detectors is obviously a way to increase the performance. In our given face-tracking scenario, we could also apply different detectors based on the head, upper-body or full body, and combine the results for a more robust result. In this regard, we have conducted some work on improving the precision of upper-body detection by including semantic colour segmentation in existing texture-based models [10].

5.3 Visual focus of attention estimation

5.3.1 Introduction

The approach I presented in the previous section is able to detect, re-identify and track persons in front of a camera. It is often necessary to further analyse the behaviour of the persons in a given scene. In particular, one wants to know what they are looking at, at any given point in time, *i.e.* what they are paying attention to in terms of their eye gaze. This can be an object or another person and is called the Visual Focus Of Attention (VFOA) (see Fig. 5.8)

Similar to our previous work on face tracking, we place this problem in a dynamic context, where the actual number of present persons is not fixed, and where their positions and the configuration of the room and furniture are not constrained. In this type of environment, many existing approaches fail, as they rely on a certain proximity to the camera or more specific hardware (depth and/or infra-red sensors). We have proposed an original algorithm that *automatically and incrementally learns* models for VFOA recognition from video data, and thus is independent from a given room or person configuration.

5.3.2 State of the art

In principle, the VFOA of a person can be determined by the person's eye gaze direction. Many studies about automatic gaze estimation from video exist [153, 268, 282, 387, 388, 396], but their use is mostly limited to close-up and near-frontal views of a person's face, for example in Human-Computer Interaction applications. Other works [138, 380, 381] rely on the fusion of information from several cameras. But often the spatial camera configuration is very constrained

or a preceding calibration step is required, which can be difficult or even impossible depending on the application and environment. Also depth sensors, like Kinect, have been used for head pose and eye gaze estimation [153]. Although, their precision depends highly on the distance of the person from the sensor, this is an interesting direction for future research beyond the scope of this work. In our work, we have focused on (non-intrusive) scenarios where a single camera is fixed at a few meters from the filmed persons and where the persons stay roughly at the same places most of the time, like in formal meetings or video-conferencing applications (as illustrated in Fig. 5.8).

Previous work on VFOA analysis in such open spaces has mostly been based on the estimation of head pose as a surrogate for gaze [67, 69, 71, 83, 87, 110, 225, 299, 308, 344, 352, 354, 355, 381, 429]. This is done either globally, *e.g.* by learning to classify image patches of the head at different angles based on low-level visual features or locally, *i.e.* by localising certain facial features [318, 411] and by geometrically and statistically inferring the global orientation, or a combination of the two [67] (see [287] for a literature survey). However, these algorithms mostly require the person(s) to face the camera more or less and be rather close to it in order to have a relatively high image resolution of the face. Using video, head pose estimation can be included in a joint head and pose *tracking* algorithm [69, 226, 262, 321]. Early works of Stiefelhagen and Zhu [353], for example, used a Gaussian Mixture Model (GMM) on head pose angles to estimate VFOA. The model is initialised with k-means and further updated with an Expectation-Maximisation algorithm. They also showed that using the other participant's speaking status increases the VFOA performance. Note that, in our work, we have concentrated on methods that are relying on *visual* information, although there are previous works that use audio, actions or or types of cues to infer the VFOA [73, 300, 344, 353]. Otsuka and Yamato [299] proposed a method based on a Dynamic Bayesian Network (DBN) that also analyses the group behaviour and detects certain conversational patterns. A GMM and Hidden Markov Model (HMM) approach for modelling and recognising VFOA was proposed by Smith *et al.* [346] for people walking by an outdoor advertisement and by Ba and Odobez [71] for analysing meeting videos. In the latter work, the authors also presented a MAP adaptation method to automatically adapt the VFOA model to the individual persons as well as a geometrical model (based on findings from [151, 177]) combining head orientation and eye gaze direction. Voit and Stiefelhagen [381, 382] built on this geometrical model and presented VFOA recognition results on a dynamic dataset with multiple cameras. Later, Ba and Odobez [72] extended their approach on VFOA estimation for meetings with a DBN that incorporates contextual information, like speaking status, slide change, and modelling conversation behaviour. Dong *et al.* [138] proposed an approach also based on a DBN which is similar to ours in the fact that they recognise VFOA by comparing tracked face image patches with a set of clusters modelling the face appearance for each attention target. However, the difference to our approach is that the clusters in their algorithm are trained before the tracking and in a supervised way. Thus, the number of targets and the targets itself are known in advance.

The work of Benfold *et al.* [87] is similar to ours in that they also perform *unsupervised* training on head images in order to determine where people look at in a given video. However, their approach is not incremental (although they claim that it could be extended) and needs an initial training of prior models using hand-labelled ground plane velocities and gaze directions of persons in a given video. They do not extract VFOA but head orientation (using a given number of classes), and they apply their approach to video surveillance data where they take advantage of people moving, which is different from our indoor scenario. On the one hand, the advantage of their probabilistic model – a conditional random field (CRF) – is that a more powerful discriminative head pose classifier can be learnt taking into account several hidden

variables (walking speed, angle velocities *etc.*). On the other hand, the complexity of learning and inference is increased, and the model is also independent from the head tracking as opposed to our approach that allows for a purely sequential and joint inference.

As experimental results of these previous works show, head pose can be used effectively to estimate the VFOA of a group of people, *e.g.* in a meeting room, to a certain extend. However, there are certain drawbacks of this approach: for example, in uncontrolled environments it is difficult to estimate head pose reliably because it often requires a large amount of annotated training data of head appearances or shapes beforehand in order to model all the possible variations of a head and face among different people as well as for a given individual. These data are often not available, or too time-consuming to produce. Further, for accurate head pose estimation results, a relatively precise localisation of the head, the face, or facial features – commonly called face alignment – is crucial but challenging in unconstrained application scenarios.

Another difficulty in automatic VFOA estimation is to determine the number of semantic visual targets for a given person in a video and to map them to given head pose or eye gaze angles. A preceding supervised training step is commonly performed on separate video data, and in some approaches the model (*e.g.* a GMM) is adapted on-line to a given video. However, it is desirable to avoid this scene-dependant training step or in some applications it might even be impossible. Further, the subsequent model adaptation can in many cases not cope with a different number of focusing targets or when the persons' locations differ too much from those in the training data.

We proposed a novel approach that alleviates these problems. Our algorithm, given a video stream from a single camera and the rough 2D position estimation of a person's head, incrementally learns to automatically extract the VFOA of the person *without explicitly estimating head pose or gaze and without any prior model of the head, face, the room configuration, or other external conditions*. The proposed method learns *on-line* the different classes of targets in an unsupervised way directly from the low-level visual features. This means also that, as opposed to supervised algorithms, it will not assign labels to the different targets (*e.g.*'table', 'screen', 'person 1'). However, we will experimentally show that the proposed unsupervised approach is able to identify and estimate the (unlabelled) targets with higher accuracy than a classical supervised approach. The fact that no pre-trained model is needed makes this approach especially interesting for applications where the specific environment, as well as the configuration of the room and the filmed persons is not known *a priori*, and where an explicit training phase is not possible.

5.3.3 Face and VFOA tracking

The principal procedure of our approach is illustrated in Fig. 5.9. First, a basic tracking algorithm is initialised and tracks a rectangular face region throughout the video stream. The image patch inside the tracked face region is extracted and visual features are computed to initialise the VFOA model at the first video frame and to update it at each subsequent frame during the training phase (see section 5.3.4). An incremental clustering algorithm on these low-level features is used to learn face appearances corresponding to attention targets of the person. At the same time a matrix modelling the transition probabilities between the different targets is learnt, and together with the clusters forms a continuous HMM. Note that the learning is performed *on-line* and does not require any prior knowledge on head pose or room configuration.

After a given number of iterations (a couple of minutes from the beginning of a video) the training phase stops and the Particle Filter continues to jointly track face position and VFOA

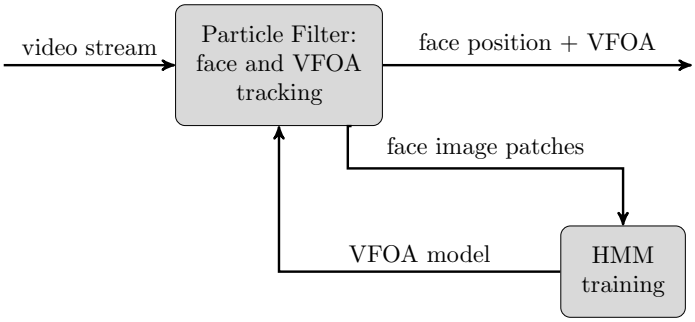


Figure 5.9: Principal procedure of the VFOA learning and tracking approach.

of a person using the learnt HMM model, *i.e.* the transition probabilities and the face clusters.

In order to facilitate understanding, before describing the main contribution of this work, *i.e.* the unsupervised VFOA learning, we will first explain the underlying tracking framework in the following section.

For tracking the face position and VFOA of a person, we used the Sequential Monte Carlo algorithm, commonly known as Particle Filter (*c.f.* [12, 304, 321]). It provides a solution for the classical recursive Bayesian model, where, assuming we have the observations $\mathbf{Y}_{1:t}$ from time 1 to t, we estimate the posterior probability distribution over the state $\mathbf{X}_t$ at time t:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = \frac{1}{C}\, p(\mathbf{Y}_t|\mathbf{X}_t) \times \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})\, \mathrm{d}\mathbf{X}_{t-1}\ , \tag{5.39}$$

where C is a normalisation constant. For simplicity and more principled evaluation, we only consider the tracking of a *single* face in a video, although our approach can easily be extended to multiple faces using an MCMC sampling approach (*c.f.* section 5.2.3) and thus be integrated in our MOT framework described in the previous section.

In our experiments, the state $\mathbf{X}_t = (\hat{\mathbf{X}}_t, v)$ is composed of the state of the face $\hat{\mathbf{X}}_t = (x, y, s)$ with x, y being its position and s being its bounding box scale factor, as well as the current VFOA target index $v \in 1..V$.

The dynamics of the face state $p(\hat{\mathbf{X}}_t|\hat{\mathbf{X}}_{t-1})$ are defined by a first-order auto-regressive model with Gaussian noise:

$$p(\hat{\mathbf{X}}_t|\hat{\mathbf{X}}_{t-1}) = \mathcal{N}(\hat{\mathbf{X}}_{t-1}; 0, \Sigma_p)\ . \tag{5.40}$$

The dynamics of the discrete VFOA target index v are defined by transition probability matrix

$$\begin{aligned} \boldsymbol{A} &:= [a_{ij}], \quad i, j = 1..V \quad \text{with} \\ a_{ij} &:= p(v_t = j | v_t = i) \end{aligned} \tag{5.41}$$

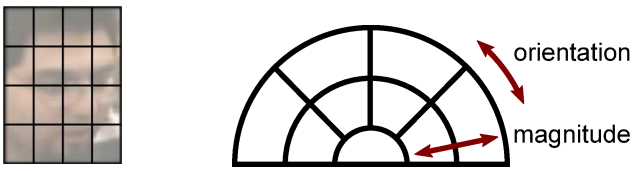
being the transition probability from VFOA target i to j. The co-variance matrix $\Sigma_p = \mathrm{diag}(\sigma_{px}, \sigma_{py}, \sigma_{ps})$ of the auto-regressive model is fixed, whereas the matrix $\boldsymbol{A}$ is learnt online during the tracking of a person in a given video stream (see section 5.3.4.2).

The observations likelihood is defined as the product of a colour likelihood and texture likelihood:

$$p(\mathbf{Y}_t|\mathbf{X}_t) = p(\mathbf{Y}_t^C|\mathbf{X}_t) p(\mathbf{Y}_t^T|\mathbf{X}_t)\ , \tag{5.42}$$

where the colour likelihood is used to track the position and size (x, y, s) of the face bounding box, and the texture likelihood is mainly used to track the VFOA target v. We define:

$$p(\mathbf{Y}_t^C|\mathbf{X}_t) \propto \exp\left(-\lambda_1 \sum_{r=1}^{9} \left(D_C^2[h_r^*, h_r(\mathbf{X}_t)] \right) \right)\ , \tag{5.43}$$



(a) (b)

Figure 5.10: Visual feature extraction for the VFOA model. a) HOG features are computed on a grid of 4×4 cells placed on the tracked face. b) To compute the histograms, gradient orientation is quantised into 4 bins (respectively 8 bins) and magnitude into 2 bins.

where λ_1 is a constant, $h_r(\mathbf{X}_t)$ are HSV colour histograms extracted from a grid of $r = 9$ cells centred at $\mathbf{X}_t$, h_r^* is the reference histogram initialised from the face region in the first frame, and D_C is the Bhattacharyya distance. As in [304], the histogram bins for the H and S channels are decoupled from the V channel. Also the quantisation is applied at two different levels, *i.e.* 4 bins and 8 bins, to improve the robustness under difficult lighting conditions. This leads to an overall colour observation vector size of $9 \cdot (8 \cdot 8 + 8 + 4 \cdot 4 + 4) = 828$.

The texture likelihood is defined similarly:

$$p(\mathbf{Y}_t^T|\mathbf{X}_t) \propto \exp\left(-\lambda_2 \sum_{r=1}^{16} \left(D_T[\boldsymbol{\mu}_{r,v}, \mathbf{t}_r(\mathbf{X}_t)]\right)\right), \tag{5.44}$$

where λ_2 is a constant, $\mathbf{t}_r(\mathbf{X}_t)$ are Histograms of Oriented Gradients (HOG) (see description below) extracted (similarly to h_r) from a grid of 16 cells (indexed by r) centred at $\mathbf{X}_t$, and $\boldsymbol{\mu}_{r,v}$ are the reference histograms corresponding to the VFOA target index v in $\mathbf{X}_t$. The overall texture model is composed of a set of N-dimensional clusters with means $\boldsymbol{\mu}_{r,i}$ where each cluster $i \in 1..V$ corresponds to a VFOA target. D_T is the normalised Euclidean distance:

$$D_T(\boldsymbol{\mu}_{r,i}, \mathbf{t}_r(\mathbf{X}_t)) = \sqrt{\sum_{j=1}^{N} \frac{(t_{r,j}(\mathbf{X}_t) - \mu_{r,i,j})^2}{\sigma_j^2 + \epsilon}}, \tag{5.45}$$

with ϵ being a small constant avoiding division by zero.

The feature vectors $\mathbf{t}_r(\mathbf{X}_t)$ constitute the visual observations used for recognising the VFOA targets of a person in a video by means of $p(\mathbf{Y}_T|\mathbf{X}_t)$. They are computed on a 4 by 4 grid of non-overlapping cells on a face image patch as illustrated in Fig. 5.10(a). For each cell, two normalised two-dimensional histograms of unsigned oriented gradients and magnitudes are computed using a specific quantisation scheme illustrated in Fig. 5.10(b). The gradient orientation is quantised in 4 bins and the magnitude in 2 bins. An additional bin (with no orientation) is used for very weak gradients (in the centre of the half circle in the diagram). Also, to improve the overall robustness and discriminative power, we compute *two* histograms at different quantisation levels for orientation: 4 and 8, and normalise each of them separately. Thus, the dimension N of the feature vector is: $16 \cdot (4 \cdot 2 + 1 + 8 \cdot 2 + 1) = 416$. One advantage of these histogram features is that they are relatively robust to small spatial shifts of the overall bounding box, which frequently occur with common face tracking methods.

5.3.4 Unsupervised, incremental VFOA learning

The VFOA model can be regarded as a dynamic HMM estimating the hidden variable v, the VFOA target index, from the observed features $\mathbf{t}_r(\mathbf{X}_t)$, illustrated in Fig. 5.11. It consists of

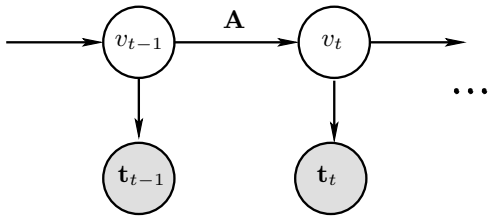


Figure 5.11: The Hidden Markov Model used to estimate the hidden discrete variable v (the VFOA target) from the observations $\mathbf{t}_t$ (feature vectors) using the learnt transition probability matrix $\boldsymbol{A}$.

two main parts. First, the data model that is used for the likelihood computation in Eq. 5.44 and that contains the k cluster means $\boldsymbol{\mu}_i$ and a global co-variance matrix $\boldsymbol{\Sigma}$, and second, the matrix $\boldsymbol{A}$ (Eq. 5.41) defining the transition probabilities from one cluster to another. All, these parameters are learnt on-line during the training phase, and used subsequently in the tracking (*c.f.* section 5.3.3). After training, the learnt parameters $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}$, and $\boldsymbol{A}$ of the HMM are used in the Particle Filter framework explained in the previous section to jointly estimate the posterior probability of the state $\mathbf{X}_t$ at each time step. In the following, the training procedures are described in more detail.

5.3.4.1 Clustering algorithm

The visual feature vectors $\mathbf{t}_r(\bar{\mathbf{X}}_t)$ computed on the image region corresponding to the mean state of the current distribution at time t are used to incrementally learn the VFOA classes. To this end, we propose a specific sequential k-means clustering algorithm with an adaptive number of clusters. The algorithm constructs a model of k clusters corresponding to the VFOA classes and described by their mean feature vectors $\boldsymbol{\mu}_{r,i}\,(i = 1..k)$ and a global diagonal co-variance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_N)$. For better readability, in the following notation, we drop the indexes for the cell r and the time step t, denoting the current cluster means as $\boldsymbol{\mu}_i$ and the current feature vector as $\mathbf{t}$. Algorithm 1 summarises the main learning procedure. At each time step the observed feature vector $\mathbf{t}$ is computed, and the closest cluster c is determined using the normalised Euclidean distance (Eq. 5.45). Also, the mean distance $\bar{D}_T$ between each of the k clusters is calculated, and a new cluster is created if the distance of the current feature vector to the closest cluster is greater than $\theta_c \bar{D}_T$, where θ_c is a parameter of our algorithm (set to 2 in our experiments). Then, the mean vector $\boldsymbol{\mu}_c$ of the closest cluster as well as the global covariance matrix $\boldsymbol{\Sigma}$ are incrementally updated using the current feature vector $\mathbf{t}$. Finally, pairs of clusters are merged together if the distance of their means are below the threshold $\theta_d \bar{D}_T$ (with $\theta_d = 0.01$ in our experiments). At each time step, the algorithm classifies the observed features $\mathbf{t}$ from the mean state of a face into one of the k clusters: c, and, as we will show with the following experimental results, the learnt classes correspond to a large degree to specific targets of VFOA.

5.3.4.2 VFOA transition model

The transition probability matrix $\boldsymbol{A}$ of equation 5.41 is learnt on-line during the training phase at the same time as the cluster centres. The main procedure is the following. The visual feature vectors $\mathbf{t}$ of the image patch corresponding to the current mean state are extracted, and the closest cluster c_t according to the normalised Euclidean distance (Eq. 5.45) is computed. Then, the transition probabilities $a_{c_{t-1},j} := p(v = j | v = c_{t-1})$ are linearly updated, using the following

Algorithm 1 Incremental VFOA clustering algorithm

$k = k_{ini}$ $\quad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{ini}$ $\quad \boldsymbol{\mu}_i = \mathbf{t}_0$ $\quad n_i = 0$ $\qquad (i = 1..k)$
for $t = 1$ to T **do**
  $c = \text{argmin}_i(D_T(\mathbf{t}, \boldsymbol{\mu}_i))$ ▷ get closest cluster
  $\bar{D}_T = \frac{2}{N(N+1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} (D_T(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j))$
  if $D_T(\mathbf{t}, \boldsymbol{\mu}_c) > \theta_c \bar{D}_T$ **then** ▷ add new cluster
    $k \leftarrow k + 1$
    $n_k = 1$
    $\boldsymbol{\mu}_k = \mathbf{t}$
  else
    $n_c \leftarrow n_c + 1$ ▷ update closest cluster
    $\boldsymbol{\mu}_c \leftarrow \boldsymbol{\mu}_c + \frac{1}{n_c}(\mathbf{t} - \boldsymbol{\mu}_c)$
  end if
  incrementally update $\boldsymbol{\Sigma}$
  for each cluster pair (i,j) **do** ▷ merge clusters
    if $D_T(\boldsymbol{\mu}_{c_i}, \boldsymbol{\mu}_{c_j}) < \theta_d \bar{D}_T$ **then**
      $\boldsymbol{\mu}_i = (n_i \boldsymbol{\mu}_i + n_j \boldsymbol{\mu}_j)/(n_i + n_j)$
      $n_i = n_i + n_j$
      remove cluster j
      $k \leftarrow k - 1$
    end if
  end for
end for

equation:

$$a_{c_{t-1},j} = \gamma \mathbb{1}_{j=c_t} + (1 - \gamma) a_{c_{t-1},j} \qquad \forall j \in 1..k, \tag{5.46}$$

where $\mathbb{1}_x$ denotes the indicator function, and the constant $\gamma = 0.001$. Thus, the transition probability from c_{t-1} to c_t is increased, and from c_{t-1} to any other cluster j is decreased. Also, a new row and column is added if a new cluster is created and inversely if a cluster is removed. At the end of each iteration, the row c_{t-1} that has been updated is normalised to sum up to 1.0. Algorithm 2 summarises the overall procedure. In many cases, the learnt transition matrix will

Algorithm 2 Incremental learning of the transition matrix

initialise $\boldsymbol{A}$ to uniform distribution: $a_{ij} = \frac{1}{k}$ $\quad i, j \in 1..k$
for $t = 1$ to T **do**
  adapt the size of $\boldsymbol{A}$ to $k \times k$
  $c_t = \text{argmin}_i D(\mathbf{t}, \boldsymbol{\mu}_i)$
  $a_{c_{t-1},j} = \gamma \mathbb{1}_{j=c_t} + (1 - \gamma) a_{c_{t-1},c_t}$
  normalise row c_{t-1} to sum up to 1.0
end for

have high values on the diagonal (staying in the same state most of the time) and low values elsewhere. Of course, this depends on the dynamics of the scene. In our formal meeting setting, people are interacting frequently and changing their attention targets quite often. Thus, this seems not to be a limitation. But even in more static settings (*e.g.* a person giving a talk), this model is still appropriate. And we can observe this with less active persons in some videos in

Figure 5.12: Example frames from the three datasets that have been used for VFOA evaluation.

our experiments. Clearly, transitions with very low probabilities can still be "triggered" if the observation likelihood of the target state is high enough. Nevertheless, to prevent extreme cases where a transition probability becomes zero and thus a state inaccessible, in our experiments, we set a very small lower boundary (10^{-3}) for the transition probabilities.

5.3.5 Experiments

We evaluated the proposed approach on three public datasets from different scenarios, each containing a certain number of persons sitting around a table and filmed roughly from the front (see Fig. 5.12): TA2[1] [39], IHPD[2] [70], PETS 2003[3].

Note that we did not evaluate the accuracy of face or head pose tracking, as this is not our main interest here. Our goal is to correctly estimate the VFOA of a person, which requires a robust face tracking system. The VFOA targets are different for each dataset, due to the scenario and the layout of the room. Annotation has been done manually and frame-by-frame, where frames with ambiguous visual focus and transition phases have not been annotated.

First, we will show some qualitative results on the clustering that is obtained on some of the videos. Fig. 5.13 illustrates the result of the proposed on-line clustering algorithm (Alg. 1) for six different persons and videos. Each point represents a 2D projection of the 416-dimensional gradient feature vectors $\mathbf{t}_r(\bar{\mathbf{X}}_t)$ extracted from the mean state at time t (after the training phase). The linear embedding has been performed by applying multi-dimensional scaling with Euclidean distance measure on the whole data. Different colours (and point shapes) correspond to different labels produced by a k-Nearest Neighbour classifier using the normalised Euclidean distance, Eq. 5.45, and the learnt cluster means $\boldsymbol{\mu}_i$ as references. Note that the clusters means have been trained during the training phase, *i.e.* the first few minutes of a video. There are two difficulties that we want to emphasise here: first, the *test* data might be distributed slightly differently (*e.g.* the person's main focus changes), and second, the training data arrives sequentially and in a non-random order, *i.e.* a person's focus changes slowly and might be static for long periods. Note also, that the 2D projection of all points suggests that clustering is difficult in many cases, like in the top middle, bottom left, and bottom right example where cluster centres and frontiers are not so clear. Nevertheless, the output of the algorithm looks reasonable, apart from the bottom right example.

Note that, as our algorithm is unsupervised, we do not have the actual estimated VFOA targets (*i.e.* meaningful labels) that we can directly compare to the ground truth. For evaluation

[1] https://www.idiap.ch/dataset/ta2
[2] https://www.idiap.ch/dataset/headpose
[3] http://www.cvg.rdg.ac.uk/slides/pets.html

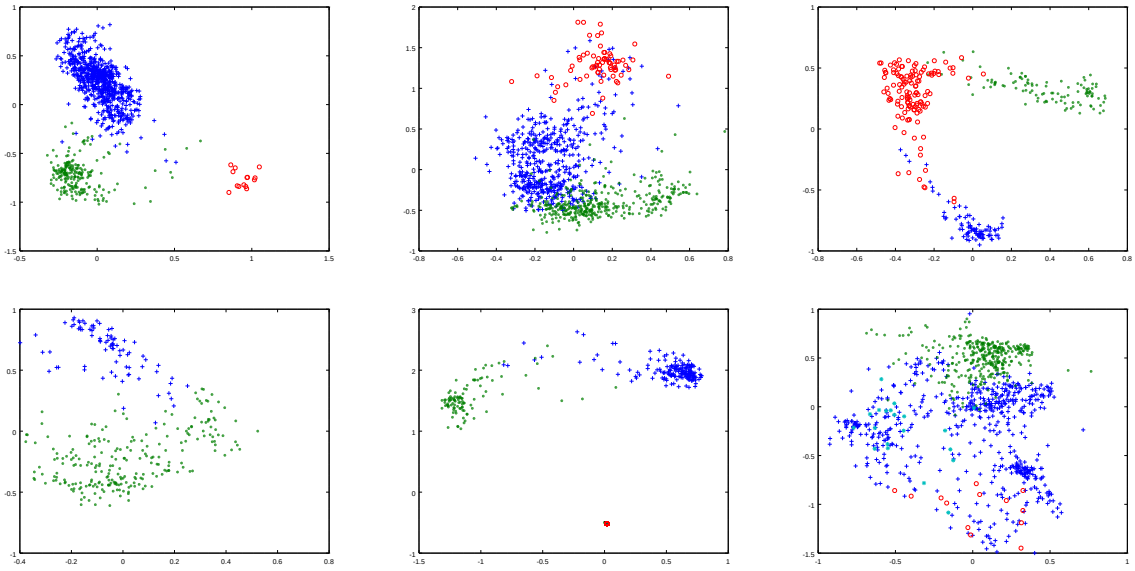


Figure 5.13: Visualisation of the clustering of low-level features produced by the proposed incremental learning algorithm (Alg. 1) for some examples. (Best viewed in colour.) From left to right, top to bottom: TA2 room 1, TA2 room 2, 2 x IHPD, PETS, and the last example shows a poor clustering result for one TA2 example.

purposes, after running our method on a whole video, we therefore assign to each cluster the target that maximises VFOA accuracy, *i.e.* we assume that we know which target label each cluster corresponds to. We believe that this is not a very restrictive assumption, as the labels could be assigned in a separate processing step, for example by incorporating a more general discriminative classifier trained beforehand.

In that way, we quantitatively evaluated our algorithm initialising it manually with a bounding box around the face and measuring the Frame-based Recognition Rate (FRR) of the VFOA for all the videos and averaging it over each dataset and over several runs. The FRR is simply the proportion of frames with correctly recognised VFOA:

$$FRR = \frac{N_c}{N_t} \,, \tag{5.47}$$

where N_c is the number of correct classifications, and N_t is the total number of annotated video frames. As our algorithm is learning the VFOA model *incrementally*, we need to account for a certain training phase, which we do not include in the evaluation. We used 8 000 ($\sim$ 5 min.) training frames in the beginning of the videos (not annotated), and evaluated the FRR on the following sequence with annotation.

We compared the proposed approach with three other approaches:

- **supervised**: a state-of-the-art supervised approach, that uses a specific face detection and tracking algorithm, a head pose estimator as in [321], and Gaussian Mixtures Models (GMM) to model different VFOA targets in terms of head pose pan and tilt angles as in [71, 353]. In this approach, the head pose model is trained beforehand in a supervised way, and the GMM parameters have been partly trained and partly defined manually.

- **no PF**: a variant of the proposed approach that does not integrate the VFOA estimation

	TA2	IHPD	PETS 2003	average
supervised [72]	0.59	0.49	0.26	0.4489
no PF	0.7663	0.5163	0.4437	0.5754
PF, fixed TM	0.6577	0.5235	0.4379	0.5397
PF, learnt TM	**0.7915**	**0.5282**	**0.4668**	**0.5955**

Table 5.2: VFOA recognition rate of the proposed algorithm with and without Particle Filter integration, and with fixed or learnt transition probability matrix $\boldsymbol{A}$ compared to a classical supervised approach.

into the Particle Filter tracking, *i.e.* v is not included in the state vector and is estimated frame-by-frame by a k-NN classifier using the feature vectors $\mathbf{t}(\bar{\mathbf{X}}_t)$ of the mean state and the cluster means $\boldsymbol{\mu}_i$, as in our first work [35].

- **PF, fixed TM**: a variant of the our approach with Particle Filter VFOA tracking and a fixed, uniform transition probability matrix $\boldsymbol{A}$.
- **PF, learnt TM**: the complete proposed approach, *i.e.* with Particle Filter VFOA tracking and learnt transition matrix $\boldsymbol{A}$

Table 5.2 shows the average FRR for these different approaches. One can see that the proposed approach outperforms the supervised method with an average FRR of ∼60% compared to ∼45%. Tracking the VFOA with a Particle Filter, as opposed to a frame-by-frame estimation, and learning the transition probability matrix on-line also improves the recognition performance on the three tested datasets. These results are comparable or superior to those published in the literature, although the evaluation protocols are not exactly the same due to the unsupervised and incremental nature of our method. Note that we do not include any contextual information like speaking status or other external events in the VFOA estimation process as in other existing work. This may additionally improve the overall performance.

5.3.6 Conclusion

We presented a VFOA tracking algorithm that incrementally, and in an unsupervised way, learns a VFOA model from directly low-level features extracted from a stream of face images coming from a tracking algorithm. The VFOA estimation is based on an HMM whose parameters are learnt incrementally and which is tightly integrated into a global Particle Filter framework that is used for face tracking. In a meeting room or video-conferencing setting, the proposed method is able to automatically learn the different VFOA targets of a person without any prior knowledge about the number of persons or the room configuration. By assigning a VFOA label to each cluster *a posteriori*, we evaluated the VFOA recognition rate for three different datasets and almost two hours of annotated data. The obtained results are very promising and show that this type of unsupervised learning can outperform traditional supervised approaches.

5.4 Conclusion

Considerable progress has been made in the last decades on MOT in more and more dynamic and difficult environments. Recent work has mostly been focusing on improving visual observation

models and shifted more and more towards learning of powerful discriminative appearance models and the use of tracking-by-detection approaches, leading to highly robust tracking methods. However, despite our contributions to robust multiple object and face tracking, many problems tackled in this chapter are still remaining to some extent. For instance, how to robustly handle false and missing detections and occlusions in on-line MOT? Or how to incrementally learn appearance models in a weakly supervised or unsupervised way with few data? Or how to adapt the appearance and tracking models to a given context and a given environment (indoor vs. outdoor, changing and possibly unknown meteorological or lighting conditions *etc.*). Such adaptive MOT algorithms are of great importance in many practical applications related to computer vision, such as social robotics, autonomous vehicles, video-surveillance or Human-Computer interaction and games.

In the next chapter, we will describe several of our research works more related to the on-line learning of powerful appearance and scene models applied to SOT. Most of them could be extended or integrated to MOT. But, as is common in the literature, for evaluation and comparison with the state-of-the-art, we do not treat and include the additional challenges related to the tracking of multiple objects.

6 On-line learning of appearance models for tracking arbitrary objects

6.1 Introduction

A crucial part in visual object tracking is the appearance model. In the works described in the previous chapter, we mainly used relatively simple models based on colour and texture histograms. The reason for this choice was their low computational complexity allowing for real-time applications, their high degree of invariance to different viewing angles, lighting conditions, occlusion and other deformations, and finally the relatively simple integration into a generative tracking framework by well-conditioned likelihood functions.

In the last years, much progress has been made on the definition of appearance models that are more robust to lighting and pose variations, background clutter, object deformations, partial occlusion and motion blur. Most of them are *discriminative models* – binary classifiers – that are trained to distinguish between the object to track and the background. Note that, in the previous chapter, we used such a model, the face detector, for track creation and termination and in the proposal function of the Particle Filter but it was not integrated in the appearance likelihood function.

Although different models can be built for each specific category of objects to track, like faces, persons, cars, it is of great interest to develop *generic models* and methods that are capable of tracking (on-line) any object in a given video stream, *e.g.* by using a designated image region in the first frame. This avoids the laborious and difficult step of defining or learning a model beforehand that is suited for any given environment and context of operation. Besides their sub-optimality in a given visual scene, such models are relatively complex and demanding in terms of computational and memory resources.

An object to track commonly undergoes a certain number of visual deformations in the image. On the one hand, this is due to the environment, like different lighting conditions, changing background or acquisition conditions. On the other hand, the object itself can change its shape, orientation or colour in the image. Therefore, to be effective in on-line tracking, appearance models need to *adapt* to these changing conditions over time. Usually, with discriminative models, this is achieved by some type of *on-line learning*, *i.e.* the model parameters are updated according to new observations from the video stream.

However, there are a few major challenges that arise with this approach:

- On-line learning methods need to adapt continuously throughout the video, because, firstly, many (different) observation samples are needed to built effective discriminative models, and, secondly, from a statistical signal processing point of view, most environments are non-stationary, and more recent observations should be prioritised. This bears the risk of gradually "forgetting" the object's appearance from the beginning.

- The on-line learning approach should be able, from continuously arriving data, to incrementally build a model that generalises well over all previous data but does not grow indefinitely.
- When updating the model, noise it introduced. For example, information from the background is considered belonging to the object or vice versa. On the one hand, updating *quickly* results in a highly adaptive model that is able to cope with abrupt appearance variations, but may lead the tracker to "drift" or, ultimately, even to lose the object. On the other hand, updating more slowly leads to a more stable model but less capable of adapting to dynamic changes. This is known as the "stability-plasticity" dilemma.

In this chapter, I will present three different approaches that we proposed to tackle these problems – all applied in an on-line SOT context. First, an efficient method to track arbitrary, deformable objects. Second, an approach to include scene information in the discriminative on-line learning of appearance models. And third, another approach to integrate scene context by dynamically selecting appropriate tracking models. The first two are joint work with Christophe Garcia at LIRIS, and the third one has been performed in the context of the PhD thesis of Salma Moujtahid [283] co-supervised with Atilla Baskurt at LIRIS.

6.2 Tracking deformable objects

6.2.1 Introduction

Tracking arbitrary objects that are non-rigid, moving or static, rotating and deforming, partially occluded, under changing illumination and without any prior knowledge is a challenging task. The problem of *model-free on-line tracking* is generally studied in the literature with benchmark videos where the goal is to follow a single object throughout the whole sequence. Given the object's initial position or bounding box in the first frame, the task is to estimate its state in the rest of the frames while sequentially processing the data. When no prior knowledge about the object's shape and appearance as well as motion is available, one of the main difficulties is to incrementally learn a robust model from consecutive video frames.

6.2.2 State of the art

Earlier works [123, 174, 175, 198, 277, 293, 304, 417] on visual object tracking mostly consider a bounding box representation (or some other simple geometric model) of the object to track with a global appearance model. These classical methods are very robust to some degree of appearance change and local deformations (as in face tracking), and also allow for a fast implementation. However, for tracking *non-rigid* objects that undergo a large amount of deformation and appearance variation, *e.g.* due to occlusions or illumination changes, these approaches are less suitable. Although some algorithms effectively cope with object deformations by tracking their contour, *e.g.* [119, 150, 302, 316, 422], most of them require the object to be moving or need prior shape knowledge [124]. Other approaches describe an object by a relatively dense set of key-points that are matched in each frame [176, 192, 235, 306] to track the object. However, these methods have mostly been applied to relatively rigid objects.

As mentioned above, many existing methods, follow a tracking-by-detection approach, where a discriminative model of the object to track is built and updated "on-line", *i.e.* during tracking, in order to adapt to possible appearance changes. For example, Adam et al. [59] use a patch-based appearance model with integral histograms of colour and intensity. The dynamic

patch template configuration allows for modelling spatial structure and to be robust to partial occlusions. Grabner et al. [165] proposed an Online Adaboost (OAB) learning algorithm that dynamically selects weak classifiers that discriminate between the object image region and the background. Later, they extended this method to a semi-supervised algorithm [166] that uses a fixed (or adaptive [349]) prior model to avoid drift and an on-line boosting framework learning with unlabelled data. Babenko *et al.* [74, 415] presented another on-line method based on Multiple Instance Learning (MIL), where the positive training examples are bags of image patches containing at least one positive (object) image patch. Besides boosting algorithms, Online Random Forests have been proposed for adaptive visual object tracking [327, 329], where randomised trees are incrementally grown to classify an image region as object or background. Kalal et al. [204, 206] also use randomised forests which they combine effectively with a Lucas-Kanade tracker in a framework called Tracking-Learning-Detection (TLD) where the tracker updates the detector using spatial and temporal constraints and the detector re-initialises the tracker in case of drift.

In order to cope with changing appearance, Mei and Ling [277] introduced the ℓ_1 tracker that is based on a sparse set of appearance templates that are collected during tracking and used in the observation model of a particle filter. Recently, several extensions or other sparse-representation-based methods have been proposed [80, 190, 366, 435, 446, 452] to improve the robustness or reduce the computational complexity. However, these approaches are still relatively time-consuming due to the complex ℓ_1 minimisation. A sparse set of templates has also been used by Liu et al. [254], but with smaller image patches of object parts, and by Kwon and Lee [222] in their Visual Tracking Decomposition (VTD) method. In a similar spirit, Ross et al. [323] proposed a particle filter algorithm called IVT that uses an observation model relying on the eigenbasis of image patches computed on-line using an incremental PCA algorithm. Wang et al. [385] also used a linear subspace appearance representation and employed a specific mechanism to eliminate outliers. As a compromise between discrimination power and ability to model deformable object, *image patches* representing object parts have been proposed by several previous works [120, 140, 409]. However, their spatial structure needs to be modelled and updated in addition, and patch-based appearance representations are sensitive to object rotations and sometimes difficult to adapt over time. Other approaches, more similar to ours, consist in using a pixel-based classifier [68, 105]. Avidan [68], for example, proposed an ensemble tracking method that labels each pixel as foreground or background with an Adaboost algorithm that is updated on-line. However, all of these methods still operate on image regions described by bounding boxes and inherently have difficulties to track objects undergoing large deformations.

To overcome this problem, recent approaches integrate some form of segmentation into the tracking process. For example, Nejhum et al. [290] proposed to track articulated objects with a set of independent rectangular blocks that are used in a refinement step to segment the object with a graph-cut algorithm. Similarly, although not segmenting the object, Kwon and Lee [224] handle deforming objects by tracking configurations of a dynamic set of image patches, and they use Basin Hopping Monte Carlo (BHMC) sampling to reduce the computational complexity. Other approaches [319, 393, 413] apply a segmentation at the super-pixel level, or at several different levels [191]. Bibby and Reid [94] proposed an adaptive probabilistic framework separating the tracking of non-rigid objects into registration and level-set segmentation, where posterior probabilities are computed at the pixel level. Aeschliman et al. [60] also combined tracking and segmentation in a Bayesian framework, where pixel-wise likelihood distributions of several objects and the background are modelled by Gaussian functions whose parameters are learnt on-line. Čehovin et al. [377] used a similar adaptive pixel-wise probabilistic colour segmentation of object and background with histograms where the output is used in the observa-

tion likelihood function of a particle filter. More recently, Wen et al. [400] proposed a multi-part segmentation and tracking approach based on a joint energy-minimisation framework. Several other recent method [92, 312] effectively employ an adaptive segmentation based on colour histograms to improve the robustness to object deformations. In a different application context, pixel-based descriptors have also been used for 3D articulated human-body detection and tracking by Shotton et al. [340] on segmented depth images. In the approach proposed by Belagiannis et al. [84], a graph-cut segmentation is applied separately to the image patches provided by a particle filter. The work of Godec and Roth [162] is similar to ours. The authors proposed a patch-based voting algorithm with Hough forests [154]. By back-projecting the patches that voted for the object centre, the authors initialise a graph-cut algorithm to segment foreground from background. The resulting segmentation is then used to update the patches' foreground and background probabilities in the Hough forest. This method achieves good tracking results on many challenging benchmark videos. However, due to the graph-cut segmentation it is relatively slow. Also, the segmentation is discrete and binary, which can increase the risk of drift due to wrongly segmented image regions.

Much more efficient are correlation filter-based tracking methods (*e.g.* [63, 92, 120, 127–129]), originally proposed by Bolme et al. [96] and its kernelised version by Henriques et al. [182]. In these approaches, tracking can be performed in the frequency domain without exhaustive sliding window search around the previous object position. However, despite recent advances they remain sensitive to local object deformations and fast appearance changes.

Recently, several methods based on Convolutional Neural Networks (CNN) have provided state-of-the-art results, *e.g.* [242, 265, 289, 390]. These networks are mostly initialised from pre-trained models (*e.g.* on the ImageNet dataset) and fine-tuned on-line during tracking [289]. Or the first pre-trained layers are used as feature extractors and combined with other tracking approaches, like correlation filters [265]. The employed neural network architectures are relatively complex and computationally expensive and rely heavily on GPU computing in order to be practical.

To cope with object deformation, several works propose approaches that incorporate a shape model. One can distinguish two families: *parametric* and *non-parametric* models. Parametric shape models employ a geometric shape, like an ellipse [95], splines [198] or other parametric curves [198, 331], or even 3D meshes [179] that are fit to the object in each image of the sequence. Usually, these models are designed for specific types of objects, like heads, faces, hands *etc.* and thus cannot be applied to general object tracking. The same limitation holds for some non-parametric models that are based on exemplar shapes, for example for pedestrians [285, 368]. On the contrary, most non-parametric models are generic. Early works proposed active contours that are based on level sets [119, 150, 302, 316, 422] allowing for arbitrary shapes and topologies. More recent approaches [60, 92, 94, 312, 377], as cited above, are based on segmentation maps as a generic discrete shape representation.

6.2.3 An adaptive, pixel-based tracking approach

Our approach, called PixelTrack [5, 34], is inspired by works on combined tracking and segmentation, which proved to be beneficial for tracking non-rigid objects while reducing the risk of model drift as opposed to template-based approaches. Furthermore, local descriptors have shown state-of-the-art performance due to their capability of handling appearance changes with large object deformations as well as partial occlusions.

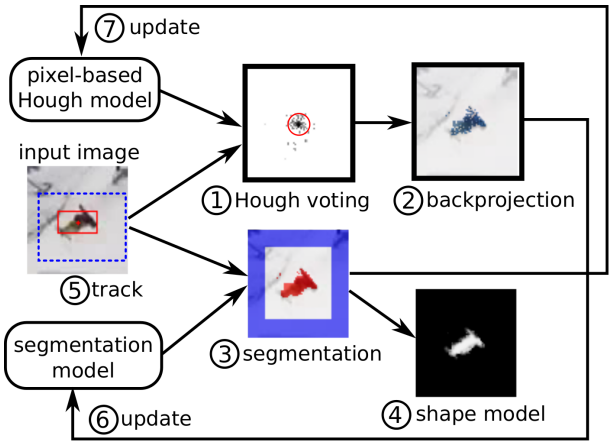


Figure 6.1: The overall tracking procedure for one video frame.

6.2.3.1 Overview

We integrated these concepts and developed a novel tracking-by-detection algorithm illustrated in Fig. 6.1. The algorithm receives as input the *current* video frame as well as the bounding box and segmentation from the tracking result of the *previous* frame. A pixel-based Hough transform is applied on each pixel inside the search window, where each pixel votes for the centre position of the object according to the learnt model, which gives the most likely position of the object's centre. Then, the pixels that have contributed to the maximum vote are selected. This process is called *backprojection*. In parallel, a pixel-based probabilistic segmentation of the image in the search window is obtained with a colour-based model , and this segmentation is used to compute a long-term shape model. The position of the tracked object is updated using the maximum vote position, the centre of mass of the segmentation output, and the shape model. Finally, the models are adapted in a co-training manner to avoid drift. In the following, each of the processing steps is explained in more detail.

6.2.3.2 Pixel-based Hough Voting

We developed a new detection algorithm relying on the generalised Hough transform [79]. In contrast to existing models developed recently for similar tasks (*e.g.* [154, 162]) which use Hough forests, *i.e.* Random Forests trained on small *image patches*, or the Implicit Shape Model (ISM) [233], our method operates at the *pixel level*.

This has the following advantages:

- pixel-based descriptors are more suitable for detecting objects that are extremely small in the image (*e.g.* for far-field vision),
- the feature space is relatively small and does not depend on spatial neighbourhood, which makes training and updating of the model easier and more coherent with the object's appearance changes,
- the training and the application of the detector is extremely fast, as the complete feature space is relatively small and can be implemented with look-up tables.

Let us now consider the model creation and application in detail. Figure 6.2 illustrates the model creation (training) and the detection process.

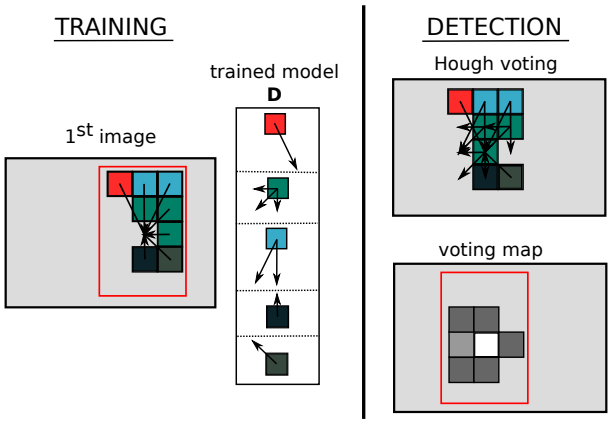


Figure 6.2: Training and detection with the pixel-based Hough model. In this example, the red, blue, green, and black pixels belong to the object. *Left:* in the first frame, the model $\mathbf{D}$ is constructed by storing for each quantised pixel value in the given (red) bounding box all the displacement vectors to the object's centre position (here only colour is used for illustration). *Right:* in a new frame, the object is detected in a search window by cumulating the displacement votes of each pixel in a voting map (bright pixels: *many* votes, dark pixels: *few* votes).

Training: The detector is trained on the first video frame. Let us denote $\mathbf{x} = (x_1, x_2)$ the position of a pixel $I(\mathbf{x})$ in an image I. In the training image, the pixels inside a given initial bounding box $\mathbf{B}_0 = (b_0^x, b_0^y, b_0^w, b_0^h)$ are quantised according to the vector composed of its HSV colour values and its gradient orientation (see Fig. 6.2 left). This amounts to computing $\mathbf{D} = \mathbf{D}_z$ ($z = 1..N$), an N-dimensional distribution similar to a histogram, which is referred to as *pixel-based Hough model* in the following. The vectors $\mathbf{D}_z = \{\mathbf{d}_z^1, \ldots, \mathbf{d}_z^{M_z}\}$ contain M_z displacement vectors $\mathbf{d}_z^m = (\mathbf{x}_{zm}, w_{zm})$, each associated with a weight $w_{zm} = 1.0$. Thus, training consists in constructing $\mathbf{D}$ by traversing each pixel $I(x_1, x_2)$ inside the bounding box $\mathbf{B}_0$, quantising the pixel value as $z = z_{\mathbf{x}}$, computing a displacement vector $\mathbf{x}_z = (b_0^x - x_1, b_0^y - x_2)$ pointing to the centre of the bounding box: (b_0^x, b_0^y) (illustrated by arrows in Fig. 6.2), and finally adding the vector $\mathbf{d}_\mathbf{z} = (\mathbf{x}_z, 1.0)$ to the set D_z.

Detection: in a new video frame, the object can be detected by letting each pixel $I(\mathbf{x})$ inside the search window vote according to $\mathbf{D}_z$ corresponding to its quantised value $z_{\mathbf{x}}$. The right part of Fig. 6.2 illustrates this. Each vote is a list of displacements $\mathbf{d}_z^m$ that are weighted by w_{zm} and cumulated in a voting map. The detector's output is then simply the position in the voting map with the maximum value $\mathbf{x}_{max}$.

Backprojection: we can determine how much each pixel inside the search window Ω_t contributed to the maximum value of the voting map $\mathbf{x}_{max}$. This process is illustrated in Fig. 6.1 and is called *backprojection*. More precisely, let z be the quantised value of pixel $I(\mathbf{x})$. Then, the backprojection b at each position $\mathbf{x} \in \Omega_t$ is defined as:

$$b_\mathbf{x} = \begin{cases} w_{zm} & \text{if } \exists\, \mathbf{d}_z^m \in \mathbf{D}_z \text{ s.t. } (b^x, b^y) + \mathbf{x}_{zm} = \mathbf{x}_{max}\ , \\ 0 & \text{otherwise,} \end{cases} \tag{6.1}$$

where (b^x, b^y) is the top-left corner of the current bounding box.

6.2.3.3 Segmentation

Complementary to the local pixel-wise Hough model, a probabilistic soft segmentation approach is adopted, similar to the ones from Aeschliman et al. [60], Čehovin et al. [377] or [312]. Let $c_{\mathbf{x},t} \in \{0,1\}$ be the class of the pixel at position $\mathbf{x}$ at time t: 0 for background, and 1 for foreground, and let $y_{\mathbf{x},0:t}$ be the pixel's colour observations from time 1 to t. For clarity and as pixels are independent in our approach, we will drop the index $\mathbf{x}$ in the following. In order to incorporate the segmentation of the previous video frame at time $t-1$ and to make the estimation more robust, we use a recursive Bayesian formulation, where, at time t, each pixel (in the search window) is assigned the probability to belonging to class $C \in \{0,1\}$ (foreground/background) :

$$p(c_t = 1|y_{0:t}) = Z^{-1} p(y_t|c_t = 1) \sum_{c'_{t-1}} p(c_t = 1|c'_{t-1})\, p(c'_{t-1}|y_{0:t-1}) \;, \tag{6.2}$$

where Z is a normalisation constant that makes the probabilities sum up to 1. The distributions $p(y_t|c_t)$ are represented with HSV colour histograms. At $t = 0$, the foreground histogram is initialised from the pixels in the image region defined by the bounding box around the object in the first frame. The background histogram is initialised from the image region surrounding this rectangle. The transition probabilities for foreground and background are set to:

$$p(c_t = 0|c_{t-1}) = 0.6 \qquad\qquad p(c_t = 1|c_{t-1}) = 0.4 \;. \tag{6.3}$$

As opposed to recent work on image segmentation (*e.g.* [324]), we treat each pixel independently, which, in general, leads to a less regularised solution but at the same time reduces the computational complexity considerably.

6.2.3.4 Shape Model

The segmentation gives a rough estimate of the current 2D shape of the tracked object in the image (see step (3) in Fig. 6.1). We use this estimate to gradually construct a non-parametric longer-term shape model (step (4) in Fig.6.1), which further helps in the overall tracking, especially for less deforming, *i.e.* more rigid, objects. The segmentation output is the posterior probability of $\mathbf{x}$ belonging to the foreground: $p(c_{\mathbf{x},t} = 1|y_{\mathbf{x},0:t})$. By computing this for all $\mathbf{x}$ inside the current search window Ω_t, we obtain a 2D segmentation map of the size of Ω_t. The shape $s_{\mathbf{x},t}$ at time t is then recursively estimated as:

$$s_{\mathbf{x},t} = \lambda\, \phi\big(p(c_{\mathbf{x}',t} = 1|y_{0:t})\big) + (1-\lambda) s_{\mathbf{x},t-1} \quad \forall \mathbf{x} \in \Omega^s \;, \tag{6.4}$$

where $\phi(\cdot)$ re-samples the segmentation map to a canonical size $\Omega^s = \Omega_0$ (*i.e.* the initial search window size), and $\lambda = 0.02$ is a small update factor. The shape model is initialised at the first frame with $s_{\mathbf{x},0} = \phi\big(p(c_{\mathbf{x}',t} = 1|y_0)\big)$. It constitutes a scale-invariant discretised representation of the 2D shape of the object. Figure 6.3 shows illustrations of the automatically learnt object shape prior models for some of our evaluation videos.

6.2.3.5 Tracking

In a new video frame, pixel-based detection and segmentation are performed inside a search window Ω_t, which we set to $\eta = 1.5$ times the size of the object's bounding box. Then, the long-term shape model is updated and the new object's position $\mathbf{X}_t = (b_t^x, b_t^y)$ and size (b_t^w, b_t^h) can be re-estimated. To this end, we utilise not only the output of the detector, *i.e.* the maximum

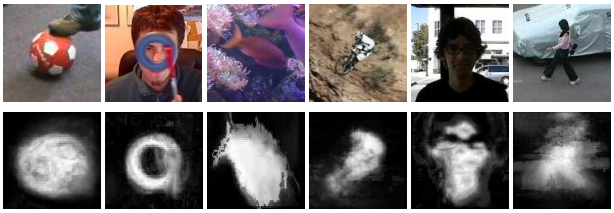


Figure 6.3: Some examples of on-line learnt shape prior models of the videos "ball", "torus", "fish2", "mountainbike", "trellis", and "woman".

position in the voting map, but also the segmentation and shape model, as described in the following. Clearly, this makes the tracking algorithm more robust to non-rigid deformations.

Segmentation map tracking: to infer an estimate of the object's position from the current segmentation, we calculate the centre of mass of the soft segmentation produced by Eq. 6.2:

$$\mathbf{x}_s = \frac{1}{S} \sum_{\mathbf{x} \in \Omega_t} p(c_{\mathbf{x}} = 1|y)\ \mathbf{x}, \tag{6.5}$$

where S is the sum of all foreground probabilities $p(c_{\mathbf{x}} = 1|y)$ in the search window Ω_t.

Shape correlation: we further exploit the longer-term shape model for tracking the object's position by computing the cross-correlation between the shape "map" and the segmentation map (scaled to the same size by ϕ):

$$\mathbf{x}_p = \operatorname*{argmax}_{\mathbf{x}} \sum_{\delta \in \mathcal{I}} \phi(p(c_{\mathbf{x},t} = 1|y_{\mathbf{x},0:t}))s_{\mathbf{x}+\delta,t}\ , \tag{6.6}$$

with $\mathcal{I}$ describing the offset in the image, *i.e.* a discrete 2D displacement of the shape map within some reasonable bounds. Thus, we match the current shape model with the segmentation map in order to obtain a position estimate.

Overall tracking result: At each frame, the new object position is set to a linear combination of the voting map maximum $\mathbf{x}_{max}$ and the mean of segmentation and shape positions $\mathbf{x}_s$ and $\mathbf{x}_p$:

$$\mathbf{X}_t = (b_t^x, b_t^y) = \alpha \frac{\mathbf{x}_s + \mathbf{x}_p}{2} + (1 - \alpha)\mathbf{x}_{max}\ . \tag{6.7}$$

The factor α determines how much we trust in the segmentation/shape position compared to the Hough model's estimation. It is computed dynamically at each frame by a simple reliability measure that is defined as the proportion of pixels in the search window Ω_t that change from foreground to background or vice versa, *i.e.* crossing the threshold $p(c_{\mathbf{x}} = 1|y) = 0.5$.

We further re-estimate the overall scale of the object at each frame using a recursive probabilistic inference. See [5] for more details.

6.2.3.6 Model adaptation

Both pixel-based Hough model and segmentation model are updated at each frame in a co-training manner, *i.e.* the output of one model is used to update the other one. To update the Hough model, only foreground pixels are used, that is pixels for which $p(c_{\mathbf{x}} = 1|y) > 0.5$. For

each of these pixels $\mathbf{x}$ the displacement $\mathbf{d}$ to the new object's centre is calculated, and its weight w is set according to its foreground probability:

$$w \leftarrow \begin{cases} \gamma\, p(c_{\mathbf{x}} = 1|y) + (1-\gamma)\, w & \text{if } \mathbf{d} \in \mathbf{D}_z\ , \\ p(c_{\mathbf{x}} = 1|y) & \text{otherwise,} \end{cases}$$

where $\gamma = 0.1$ is the update factor. In the second case (*i.e.* if $\mathbf{d} \notin \mathbf{D}_z$), $\mathbf{d}$ is added to $\mathbf{D}_z$. For computational and memory efficiency, we limit the size of each $\mathbf{D_z}$ and only keep the K displacements with the highest weights ($K = 20$ in our experiments).

The foreground and background distributions of the segmentation model are adapted using the backprojection $b_{\mathbf{x}}$. That is, the colour distribution $p(y|b > 0.5)$ of the backprojected pixels is calculated, and used to linearly update the current foreground colour distribution:

$$p(y_t|c_t = 1) = \delta\, p(y|b > 0.5) + (1-\delta)\, p(y_{t-1}|c_{t-1} = 1)\ , \tag{6.8}$$

where $\delta = 0.1$ is the update factor. The background colour distribution is updated in the same way but using the colour distribution from a rectangular frame surrounding the object borders (as for the initialisation step).

6.2.4 Experiments

We conducted quantitative evaluation on two sets of challenging standard videos that are commonly used in the literature: the "Babenko sequences" [74][4] and the "Non-rigid object dataset" [162][5], as well as the public tracking benchmark VOT2014[6] [218]. The tracking accuracy and speed on these datasets has been measured and compared to several state-of-the-art tracking methods.

Using the first two datasets, we compared our algorithm, called PixelTrack+, to 11 state-of-the-art methods: HoughTrack (HT) [162], Tracking-Learning-Detection (TLD) (CVPR version) [204], Incremental Visual Tracker (IVT) [323], the Multiple Instance Learning tracker (MIL) (CVPR version) [74], the ℓ_1 tracker using the Accelerated Proximal Gradient method (ℓ_1 APG) [80], the structured output tracker: Struck (ICCV version) [175], Multi-Level Quantisation tracker (MQT) [191], the Deep Learning Tracker (DLT) [391], MEEM [430], DSST [127] and MUSTer [192].

To measure the performance of the different tracking algorithms, we determine, for each video, the proportion of frames in which the object is correctly tracked. And, to measure the tracking precision, we computed the average Normalised Centre Error (NCE) on each video, *i.e.* the Euclidean distance between the ground truth rectangle's centre and the corresponding tracked bounding box in a frame. Table 6.1 summarises the results. Although our method is not designed for grey-scale videos and, thus, does not show its full potential, it still performs better on average than most of the state-of-the-art methods. Only MEEM outperforms PixelTrack+ on both measures. DSST and MUSTer also generally have a higher proportion of successfully tracked frames as they rely less on colour segmentation which is not useful in grey-scale videos.

For the VOT2014 data, we followed the evaluation protocol of the benchmark [219], using the accuracy (same as our correctly tracked frame measure above) and the robustness (number of tracking failures). Figure 6.4 shows the accuracy-robustness plots for the baseline experiment of the VOT2014 benchmark [218] comparing 33 state-of-the-art tracking methods. Our method

[4] http://vision.ucsd.edu/ bbabenko/project_miltrack.html
[5] http://lrs.icg.tugraz.at/research/houghtrack/
[6] http://votchallenge.net/

	HT	TLD	IVT	MIL	ℓ_1 APG	Struck
Babenko	0.65 / 0.55	0.70 / 0.23	0.61 / 0.29	0.63 / 0.31	0.76 / 0.29	0.86 / **0.18**
Non-rigid	0.79 / 0.23	0.55 / 0.26	0.55 / 0.32	0.71 / 0.24	0.56 / 0.32	0.74 / 0.26

MQT	DLT	MEEM	DSST	MUSTer	**ours**
0.54 / 0.54	0.84 / 0.21	**0.91** / **0.18**	0.89 / 0.44	0.90 / 0.23	0.86 / 0.19
0.73 / 0.19	0.66 / 0.24	0.83 / 0.28	0.66 / 0.28	0.74 / 0.32	**0.90** / **0.18**

Table 6.1: Babenko and Non-rigid objects sequences: proportion of correctly tracked frames (1st number), and Normalised Centre Error (NCE) (2nd number).

algorithm	HT	TLD	IVT	MIL	ℓ_1 APG	Struck	MQT	DLT	MEEM	DSST	MUSTer	ours
speed	2.3	5.2	8.4	5.2	7.1	9.9	1.0	15	10	**59.4**	4.0	44.2

Table 6.2: Comparison of the average processing speed in frames per second.

compares favourably with the top performing methods, especially in terms of robustness. Superior in terms of accuracy and robustness are only PLT, a structured output SVM-based tracker, DGT [104], that tracks super-pixels using graph-matching, and MatFlow [272], a key-point-based approach. Correlation filter-based approaches like DSST [127] show a very high accuracy but slightly lower robustness.

Figure 6.5 shows tracking results of PixelTrack+ compared to several state-of-the-art methods for some very challenging videos.

Finally, we measured the average processing speed of each algorithm for the 19 videos of the first two datasets on a 3.4 GHz Intel Xeon processor (using a single core). The results are shown in Table 6.2. The execution speed of the proposed method is at least 3 times faster than most of the other state-of-the-art methods, except for DSST which is slightly faster.

6.2.5 Conclusion

We designed a fast algorithm for tracking generic deformable objects in videos without any prior knowledge, *i.e.* without any learnt appearance model of the object or the surrounding scene and without any specific motion model. It is an effective combination of a pixel-based detector based on a Hough voting scheme and a global probabilistic segmentation method that operate jointly and update each other in a co-training manner. Moreover, a novel long-term non-parametric shape model has been proposed to further improve the robustness of the approach. Our algorithm has two strengths compared to existing state-of-the-art methods: firstly, using a pixel-wise soft segmentation it is able to track objects that are highly deformable, such as articulated objects, as well as objects that undergo large in-plane and out-of-plane rotations. And secondly, it is very fast, which makes it suitable for real-time applications, or tasks where many objects need to be tracked at the same time, or where large amounts of data need to be processed (*e.g.* video indexation). Our experimental results show that the method outperforms state-of-the-art tracking algorithms on challenging videos from standard benchmarks.

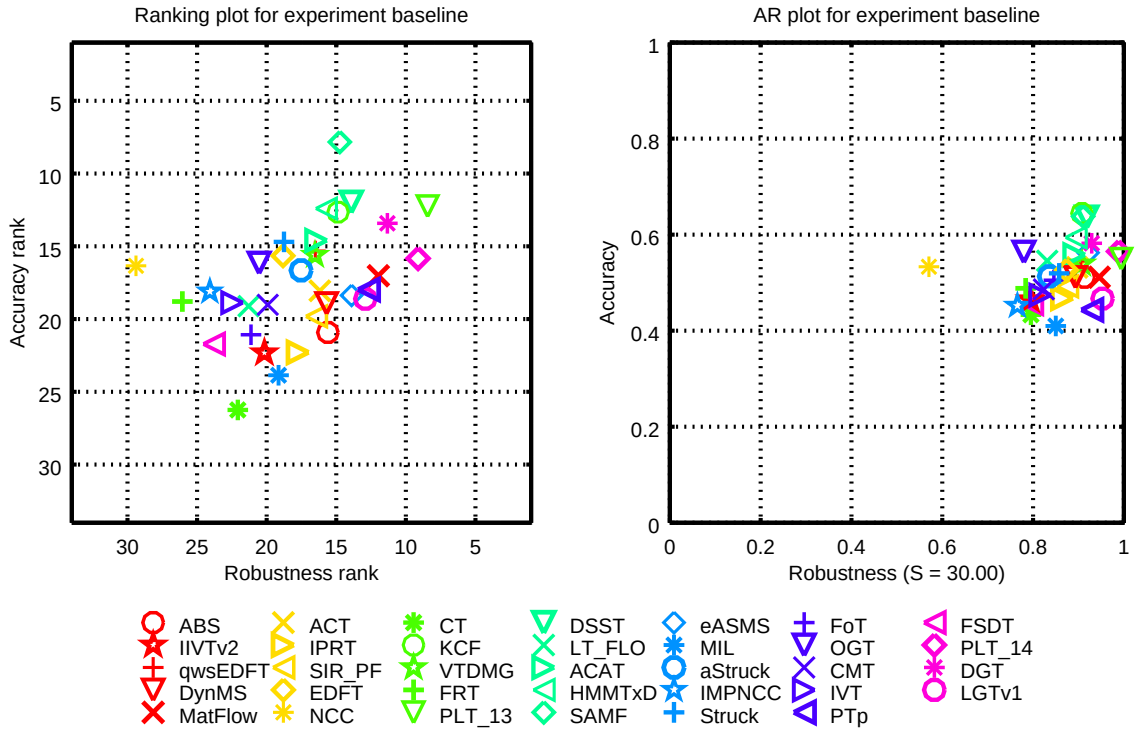


Figure 6.4: Comparison with 33 state-of-the-art methods of the VOT2014 benchmark [218]. The proposed method "PTp" is among the top-performing methods with the fifth-best robustness.

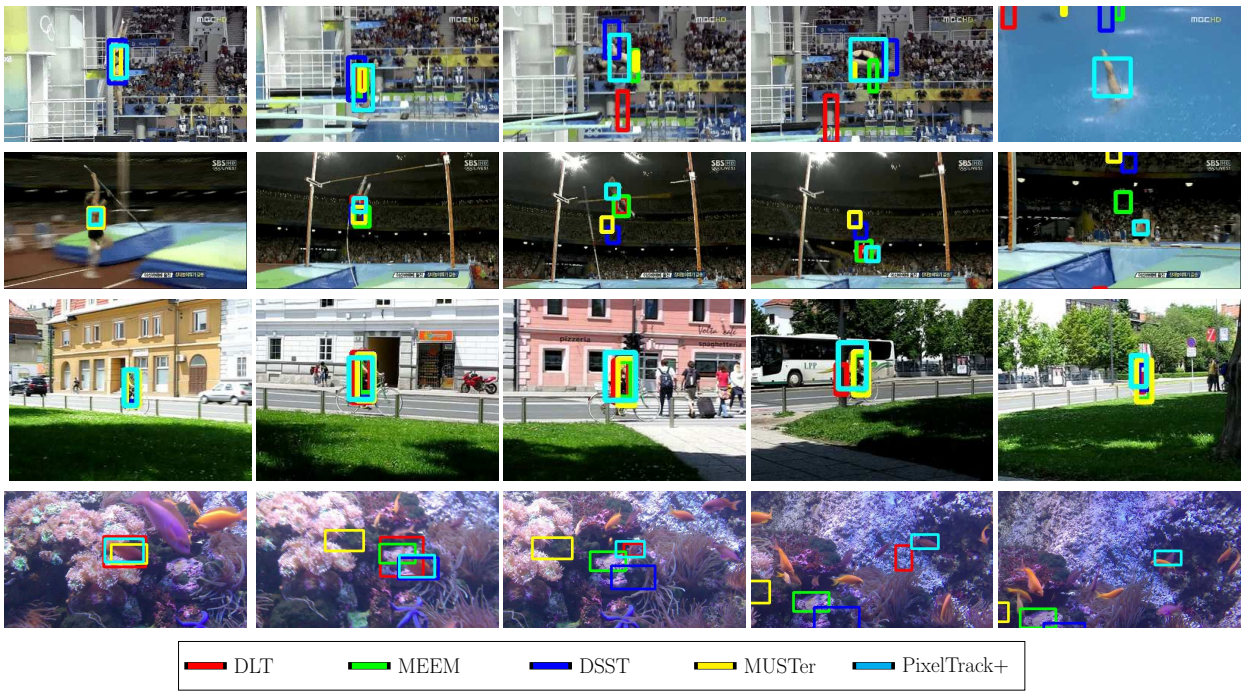


Figure 6.5: Tracking results of DLT [391], MEEM [430], DSST [127], MUSTer [192] and the proposed PixelTrack+ on some of the more challenging videos from the non-rigid object dataset and the VOT2014 benchmark (best viewed in colour).

6.3 On-line learning of motion context

6.3.1 Introduction

As experimental results show, the previous work addressed well the problem of tracker drift and is able to learn on-line a stable appearance model in dynamic environments. However, when analysing the behaviour and the results of our tracking algorithms and the ones from the literature, we realised that there are still certain failure modes, *e.g.* certain situations that have not been considered in the design of these algorithms. One of these situations is when, in the background, there are very similar objects to the one(s) that we want to track, and when one or several of these distracting objects comes close to it. Then many algorithms might lose track and "jump" to the image region corresponding to the most similar object. This happens, for instance, when tracking a person in sports videos or an animal in a herd or swarm. Thus, in a SOT problem, it seems important to not only consider the object to track but also its global environment.

For this reason, together with Christophe Garcia, we studied methods to incorporate the *context* of the visual scene into the tracking process. In many practical applications, the camera is moving. Thus, the background and the visual context is also changing over time. Therefore, we adopted again an on-line learning approach in order to be able to dynamically adapt to new environments and situations. In contrast to the tracking-by-detection method PixelTrack+ described in the previous section, we opted here for a probabilistic Bayesian approach, as it allows for a straightforward integration and combination of different appearance models and observation likelihoods as well as motion models, and for an effective inference with particle filtering.

6.3.2 State of the art

One of the first works to explicitly include context in visual object tracking has been the Context-Aware Tracker (CAT) by Yang *et al.* [414]. Their method operates on-line throughout the video and continuously discovers objects that move in the same direction as the tracked object by performing a motion correlation analysis. These auxiliary objects help to support and improve tracking by performing inference in a star-structured graphical model that includes the objects' states. Similarly, Zhang *et al.* [431] modelled the spatio-temporal relationships and correlations between the object and its locally dense contexts in a Bayesian framework.

Spatial context has also been exploited by using *supporters*, *i.e.* other objects or feature points around the target in the image. For example, this may be useful in videos, where a hand holds the object to track and thus exhibits the same visible motion pattern. Or when groups of people walk in the same direction. Grabner *et al.* [167], for example, extended the well-known Implicit Shape Model by detecting feature points in the image that have a correlated motion with the target. These supporters are matched from frame to frame, and their relative displacement vectors are updated on-line. Wen *et al.* [399] also proposed a method that detects supporters (here called contributors) which are interest points within a local neighbourhood around the target, in order to improve the tracking performance. In addition, their method makes use of a longer-term temporal context using an on-line sub-space learning method that groups together observations from several frames. Similarly, the approach proposed by Sun *et al.* [362] tracks "helper" objects using an on-line Adaboost detector, initialised manually at the first frame. Their relative position is learnt on-line and used to predict the target object's position.

Dinh *et al.* [136] proposed a method using supporters as well as distractors, which are objects with similar appearance to the target. The distractors help to avoid confusion of the tracker with

other similar objects in the scene, and they can possibly be used to reason about the objects' mutual occlusion. In the work of Dinh *et al.*, supporters are not used directly for the target's state estimation but only to disambiguate between the target and its distractors. Hong *et al.* [190] recently proposed an approach based on the ℓ_1 tracker [277] that deals with distractors by automatically learning a metric not only between positive and negative examples but also within the collected negative examples, effectively replacing the originally proposed Euclidean distance. Finally, Supančič and Ramanan [363] presented a self-paced learning tracker that also selects training examples from video frames in the past to perform long-term tracking, an idea that has also been used in the recent work of Hua *et al.* [195].

The disadvantage with using supporting and distracting objects is that several objects need to be detected and tracked, which can be computationally expensive especially with a larger number of objects. Moreover, the success or failure of data association or, in some methods, matching local features points in successive video frames, heavily depends on the type of object to track and the surrounding background. This process can be error-prone and, in some situations, may rather harm the overall tracking performance. Finally, modelling the spatial, temporal, or appearance-based pairwise relationships between objects and/or interest points can lead to a combinatorial explosion and make the inference on the state space difficult.

To alleviate this problem, in this work, we propose a probabilistic method that dynamically updates the foreground and background model depending on distracting objects or image regions in the scene background. This contextual appearance information is extracted from moving image regions and used to train on-line a discriminative binary classifier that, in each video frame, detects the image region corresponding to the object to track.

Traditionally, these discriminative on-line classifiers used in tracking-by-detection approaches [74, 154, 162, 165, 175, 277] learn negative examples extracted from the image region surrounding the current target object region. This choice is motivated by the fact that the object will move only slightly from one frame to the other w.r.t. the background or other objects, and by computational speed. In contrast, our method uses a stochastic sampling process to extract negative examples from image regions that move. We call these: *contextual motion cues* (see Fig. 6.6). In that way, regions that correspond to possibly distracting objects are detected efficiently and early, *i.e.* without them having to be inside a search window and without scanning the whole image at each point in time.

6.3.3 Tracking framework with discriminative classifier

In this work, we used a traditional recursive Bayesian model for tracking and a particle filter, *i.e.* sampling importance resampling (SIR) or bootstrapping, for the inference [139, 198]. As we already described the principal model before (*c.f.* section 5.2.3 for MOT and section 5.3.3 for SOT), we only concentrate on the original parts and on our contributions, *i.e.* the on-line learning of a discriminative classifier using motion context cues and its integration in the particle filter framework using an effective likelihood model and proposal functions.

6.3.3.1 Object state representation and inference

The state $\mathbf{X} = (x, y, v_x, v_y, s, e) \in \mathbb{R}^6$ of the object to track is described by an upright bounding box defined by the object's centre position (x, y) in the image, its 2D speed (v_x, v_y) in the image plane, scale (s), and eccentricity (e), *i.e.* the ratio of height and width. The state $\mathbf{X}_0$ is initialised manually by providing a bounding box around the object in the first frame. Then, for each video frame, the particle filter performs its classical steps of *predicting* particles $\mathbf{X}^{(i)}$ sampled from

the proposal distribution $q(\mathbf{X}_t|\mathbf{X}_{t-1})$ and *updating* their weights according to the observation likelihood $p(\mathbf{Y}_t|\mathbf{X}_t)$, state dynamics $p_m(\mathbf{X}_t|\mathbf{X}_{t-1})$ and proposal (see Section 6.3.3.2): $w_i = p(\mathbf{Y}_t|\mathbf{X}_t)\frac{p_m(\mathbf{X}_t|\mathbf{X}_{t-1})}{q(\mathbf{X}_t|\mathbf{X}_{t-1})}$, for each particle $i \in 1..N$. At the end of each iteration, the observation likelihood model parameters are updated using the mean of the posterior distribution $p(\mathbf{X}_x|\mathbf{Y}_{1:t})$. And finally, systematic resampling is performed.

6.3.3.2 State dynamics and proposal function

In order to cope with fairly complex motion of arbitrary objects in videos from a possibly moving camera, we used a proposal function composed of a mixture of three distributions:

$$\begin{aligned} q(\mathbf{X}_t|\mathbf{X}_{t-1}) = \beta_m p_m(\mathbf{X}_t|\mathbf{X}_{t-1}) \\ + \beta_f p_f(\mathbf{X}_t|\mathbf{X}_{t-1}) \\ + \beta_d p_d(\mathbf{X}_t|\mathbf{X}_{t-1}) \ , \end{aligned} \tag{6.9}$$

where $\beta_m, \beta_f, \beta_d < 1$ define the mixture weights ($\sum_i \beta_i = 1$), and $p_m(\mathbf{X}_t|\mathbf{X}_{t-1})$ is the state dynamics model (similar to the ones in sections 5.2.3.2 and 5.3.3), $p_f(\mathbf{X}_t|\mathbf{X}_{t-1})$ is an optical flow-like motion-based proposal function, and $p_d(\mathbf{X}_t|\mathbf{X}_{t-1})$ proposes states coming from a discriminative on-line trained detector described in more detail in section 6.3.4.

6.3.3.3 Observation likelihood

The observation likelihood function $p(\mathbf{Y}|\mathbf{X})$ that we proposed is designed to be robust against object deformations, pose and illumination changes as well as partial occlusions. It is a geometric mean of three distributions corresponding to different visual cues:

$$p(\mathbf{Y}_t|\mathbf{X}_t) = \left(p_H(\mathbf{Y}_t|\mathbf{X}_t)\, p_S(\mathbf{Y}_t|\mathbf{X}_t)\, p_T(\mathbf{Y}_t|\mathbf{X}_t)\right)^{1/3} \ , \tag{6.10}$$

where p_H computes a local colour histogram likelihood ratio, p_S measures the global colour distribution similarity, and p_T is a texture likelihood based on the on-line learnt discriminative classifier that we will explain in the following section. Taking the cube root of the product ensures that the overall likelihood distribution does not become too peaked. For more details, refer to [6].

6.3.4 Model adaptation with contextual cues

In this section, we will describe the main contribution of the proposed approach: a method to exploit motion context effectively for visual object tracking using a discriminative classifier that is trained on-line on specific parts of the input video. Our approach is different from previous work, where motion context or background motion has been integrated tightly in the tracking process (*e.g.* in [160, 293], or where specific appearance models are used to avoid distractions in the background [136, 190].

In our particle filter framework outlined above, we used a binary discriminative classifier based on the On-line Adaboost (OAB) algorithm [165] (based on Haar-like features) for proposing new particles as well as for evaluating the observation likelihood. Any other on-line classifier could have been used as well. The classifier is trained with the first video frame using the image patch inside the object's bounding box as a positive example and surrounding patches within a search window as negative examples (as we cannot infer motion from a single image). Then, the classifier is updated at each tracking iteration using the same strategy for extracting positive and negative examples. We refer to [165] for details on the model and how it is trained.

Figure 6.6: Illustration of different sampling strategies of negative examples (blue). *Left:* traditional sampling at fixed positions within a search window (red). *Middle:* the motion probability density function m (Eq. 6.11). *Right:* the proposed negative sampling from m.

6.3.4.1 Background sampling

We propose to sample negative examples from image regions that contain motion and thus likely correspond to moving objects (see Fig. 6.6). The idea is that these regions may distract the tracker at some point in time. Therefore, it is preferable to incorporate these distracting image regions in the classifier training in the form of negative examples and learn them as early as possible, *i.e.* as soon as they appear in the scene. One can see this as a kind of long-term prediction of possible negative samples, in contrast to the much shorter (frame-by-frame) time scale of the proposal function. To perform this negative sampling, we first compensate for camera motion between two consecutive frames using a classical parametric motion estimation approach [292]. We apply a three-parameter model to estimate the translation and scale factor between the images, and then compute the intensity differences for each pixel in the current image with its corresponding pixel in the previous frame. This gives an image $M(x, y)$ approximating the amount of motion present at each position (x, y) of the current frame of the video. We then transform this image into a probability density function (PDF) $m(x, y)$ over the 2-dimensional image space:

$$m(x, y) = Z^{-1} \sum_{(u,v)\in\Omega(x,y)} M(u, v) \ , \tag{6.11}$$

where $\Omega(x, y)$ defines an image region of the size of the bounding box of the object being tracked, centred at (x, y), and Z is a constant normalising the density function to sum up to 1. Thus, $m(x, y)$ represents the relative amount of motion inside the region centred at (x, y). Finally, N^- image positions (x, y) are sampled from this PDF corresponding to rectangles centred at (x, y). That is, statistically, regions with high amount of motion are sampled more often than static image regions. This process is illustrated in Fig. 6.6.

6.3.4.2 Classifier update

The N^- image patches corresponding to the sampled regions as well as the positive example coming from the mean particle of the tracker are then used to update the classifier. In this case, the OAB method needs a balanced number of positives and negatives, thus the positive example is used N^- times, alternating positive and negative updates.

The advantage of sampling positions from these motion cues is that we do not need to care about explicitly detecting, initialising, tracking, and eventually removing a certain number of distracting objects at each point in time. Note that we could also sample regions of different scales but as scale does not change rapidly in most videos the benefit of this is relatively small.

	fixed	fixed+random	motion	fixed+motion
Babenko	73.28	74.06	82.30	**85.25**
Non-rigid	68.71	70.30	74.29	**80.87**

Table 6.3: Average percentage of correctly tracked frames with the proposed method using different negative sampling strategies.

Note also that the PDF could as well include appearance similarity with the tracked target. However, this would considerably increase the computational complexity.

6.3.5 Experiments

We performed a quantitative evaluation of our proposed approach, that we called "Motion Context Tracker" (MCT), on four challenging public tracking datasets: the Babenko and Non-rigid object sequences also used to evaluate our PixelTrack+ method, described in the previous section, and the VOT2013 and the VOT2014 datasets. Again, we measured the accuracy ($\equiv$ the percentage of correctly tracked frames) and the robustness ($\equiv$ number of tracking failures).

In the first experiments, we evaluated different strategies for the collection of negative examples of the discriminative OAB classifier, as explained in Section 6.3.4. We compared four different strategies:

- **fixed:** N^- negatives are taken from fixed positions around the positive example inside the search window, which is twice the size of the object's bounding box.
- **fixed+random:** $N^-/2$ examples are taken from fixed position (as for "fixed"), and $N^-/2$ examples are sampled from random image positions.
- **motion:** N^- negative examples are sampled from the contextual motion distribution m (Eq. 6.11).
- **fixed+motion:** $N^-/2$ examples are taken from fixed positions, and $N^-/2$ examples are sampled from the contextual motion distribution.

In any case, the negative examples do not overlap more than 70% with the positive ones in the image. Table 6.3 shows the results for the first two datasets in terms of the percentage of correctly tracked frames. On average, the best strategy is "fixed+motion", with a relative improvement of around 7.5%.

Using the VOT2013 benchmark, we compared MCT with 27 other state-of-the-art tracking methods. Table 6.4 lists the top 7 ranks for the experiments baseline, region-noise (*i.e.* with perturbed initial bounding boxes), and greyscale (*i.e.* videos converted to grey-scale), combining accuracy and robustness. The results of MCT are very competitive, being the second-best method for baseline and region-noise and the third-best for greyscale. We also added the method PF (MCT without the detector) to the VOT2013 evaluation. Its overall ranks for the baseline, region-noise, and greyscale experiments are 16.1, 14.5, and 14.4 respectively. This clearly shows that the benefit of the motion context-based discriminative classifier.

Finally, Table 6.5 lists the 10 best methods for VOT2014 and the respective accuracy ranks, robustness ranks, and overall ranks. Overall, the results of our MCT approach are very competitive. Taking the average of accuracy and robustness ranks, PLT and its extension PLT_14 are still slightly better, as well as the correlation filter-based method SAMF [413], and the method

baseline		region-noise		greyscale	
PLT	4.96	PLT	3.58	PLT	3.96
MCT	**6.62**	**MCT**	**5.08**	FoT [384]	4.75
FoT [384]	8.25	CCMS	8.33	**MCT**	**6.25**
EDFT [146]	9.5	FoT [384]	9.04	EDFT [146]	7.5
CCMS	9.54	LGT++ [408]	9.04	GSDT [155]	9.5
LGT++ [408]	10.2	EDFT [146]	9.08	LGT++ [408]	9.58
DFT [336]	11.1	LGT [377]	10.5	Matrioska [272]	10.7

Table 6.4: Overall ranking result with the VOT2013 dataset. Only the first 7 out of 28 ranks are shown.

	accuracy rank	robustness rank	overall rank
SAMF [413]	8.16	16.49	12.33
PLT	14.28	10.41	12.35
DGT [104]	11.42	13.44	12.43
PLT_14	17.46	10.77	14.12
MCT	**13.52**	**14.76**	**14.14**
PF	13.70	14.74	14.22
DSST [127]	13.51	15.54	14.53
KCF [182]	13.62	16.82	15.22
HMMTxD [218]	13.18	17.57	15.38
MatFlow [272]	16.90	15.29	16.10

Table 6.5: Overall ranking result with the VOT2014 dataset. Only the first 10 out of 39 ranks are shown.

DGT [104] which relies on graph matching and super-pixel representations. The method PF, *i.e.* MCT without the discriminative classifier, is only slightly worse on average with this benchmark. This might be due to the more challenging type of videos with deformable objects for which the texture-based classifier is not powerful enough.

In terms of execution speed, our algorithm runs at around 20fps for a frame size of 320×240 on an Intel Xeon 3.4GHz (single core).

6.3.6 Conclusion

We proposed a new efficient particle filter-based approach for tracking arbitrary objects in videos. The method combines generative and discriminative models, by effectively integrating an on-line learning classifier. We introduced a new method to train this classifier that samples the position of negative examples from contextual motion cues instead of a fixed region around the tracked object. Our extensive experimental results show that this procedure improves the overall tracking performance with different discriminative classification algorithms. Further, the proposed tracking algorithm gives state-of-the-art results on four different challenging tracking datasets, effectively dealing with large object shape and appearance changes, as well as complex motion, varying illumination conditions and partial occlusions. Note that we also officially participated in the VOT2014 challenge [32] and the associated workshop held in conjunction

with ECCV 2014 [31]. As can be seen from the results (*c.f.* 6.5), our MCT method achieved an excellent position in the ranking.

In summary, we have shown experimentally that dynamically integrating context information from the visual scene using on-line learning can largely improve the overall performance of the tracking algorithm. In the following section, we will outline our research work that develops this idea further by constructing finer global scene context models in an on-line manner.

6.4 Dynamic adaptation to scene context

6.4.1 Introduction

In this section, I will present our research work performed in the context of the PhD thesis of Salma Moujtahid [23, 28, 29, 283]. We worked on another approach of including contextual scene or background information in the tracking process, and we applied it to the SOT scenario, as in the previous section. However, here, scene context is not used for the on-line learning of robust appearance models, as with MCT, but for a dynamic selection of trackers among a set of methods depending on their suitability for the given environment at each point in time.

Many tracking algorithms have been proposed in the literature, and each of them has strengths and weaknesses depending on the appearance, motion or state models or the type of inference.

Our assumption was that their performance varies in different types of settings and environments and, thus, for a given context, we would be able to select the algorithm and model that is most appropriate. As these variations might occur not only from one type of video or application to another but also *within* a given video stream, we developed a method that dynamically selects one tracker from a pool of supposedly complementary trackers depending on the scene context at a given point in time. To this end, we conceived a set of general scene context features, and we extracted these features from each frame of a set of training videos. Then, a classifier was trained (off-line) on these features to predict which tracker is the most suitable for a given moment in the video. Finally, we proposed an original tracking framework that selects the most appropriate tracker at each frame based on this classifier. In the following, we will describe this approach in more detail.

6.4.2 State of the art

Many ways of combining, fusing or selecting visual models or features for tracking exist in the literature. They can be categorised into low-level and high-level fusion approaches.

6.4.2.1 Low-level fusion

Fusion at a low level means the combination of multiple visual features in a single tracking model. Early works like the one from Birchfield *et al.* [95] used the sum of colour histograms and intensity gradient likelihoods. Collins *et al.* [122] also used likelihood maps to rank features and select the most discriminant ones. Triesh *et al.* [369] weight features in other to fuse them in a democratic integration for face tracking. A Bayesian framework introduced by Yilmaz *et al.* [422] fuses probabilistic density functions based on texture and colour features for object contour tracking. Other existing works (*e.g.* [305, 349, 423]) fuse different modalities, like motion or shape, in order to improve the overall foreground-background discrimination. The low-level fusion of features might lead to problems because of the interdependence of the features in the

tracking model. If some of the visual cues are altered or occluded because of changing scene conditions, the whole model is prone to drift.

6.4.2.2 High-level fusion

Other approaches consider the combination of the output of multiple trackers. For example, a probabilistic combination was proposed by Leichter *et al.* [234] with multiple synchronous trackers using different features where each tracker estimates a probability density function of the tracked state. A sampling framework is introduced by Kwon *et al.* [222, 223] integrating estimates from basic complementary trackers using different observation models and motion models. Moreover, tracker performance within a parallel framework can be measured as the disagreement of a tracker with respect to the other trackers. Li *et al.* [244] exploited this idea to seek a balance between the trackers. The recent work by Khalid *et al.* [212] fuses the output of multiple trackers based on their estimated individual performance and the spatio-temporal relationships of their results. Using the object trajectory as a fusion criteria is another possibility. For example, Bailer *et al.* [76] used trajectory optimisation to fuse the tracking results (bounding boxes) from different tracking algorithms. Wang *et al.* [392] also modelled the object trajectory and the reliability of each of five independent trackers combining them with a factorial Hidden Markov Model (HMM). More recently, Vojir *et al.* [383] also utilised a HMM to fuse observations from complementary trackers and a detector. The HMM's latent states correspond to a binary vector expressing the failure of the individual trackers.

In a different manner, rather than using multiple trackers with different cues, Zhang *et al.* [430] retain snapshots of a base tracker (an on-line SVM) in time, constituting an ensemble of its past models. Then a model is restored according to an entropy criterion.

In terms of tracker *selection* rather than *fusion*, our previous work [29] presented a simple selection framework using the trackers confidences and a spatio-temporal criteria. Similarly Stenger *et al.* [351] also used confidences to select the best tracker applied to face tracking. The main advantage of selection algorithms is that the resulting tracking output is not altered by one or more trackers that may have drifted as long as they are not selected.

In contrast to these fusion frameworks, our proposed selection framework does not rely solely on the performance or tracking results of our trackers. It differs essentially in the use of the scene context in the decision of selecting the most suitable tracker at each point in time. Moreover, selecting from a pool of independent trackers allows to adapt to rapid scene changes and quickly switch between different models.

6.4.2.3 Context in tracking

Scene context has been used previously in visual tracking and detection. Especially in on-line detection or tracking scenarios, where information for model construction is very limited to the number of frames, context can provide important information and greatly improve the performance. Context can be used in many ways. As mentioned in section 6.3.2, some works (*e.g.* [167, 399, 414, 431]) use "supporters", *i.e.* image regions or interest points moving similarly to the tracked object and assisting the tracking, or "distractors" [136, 190], *i.e.* image regions with similar appearance to the object, in order to avoid confusion in tracking. However, the modelling of spatial and temporal relationships between the different tracked objects or interest points is computationally expensive, due to the more complex data association. In a different manner, Maggio *et al.* [270] used contextual event cues such as target births (objects entering the scene) and spatial clutter of objects. The spatial distribution of these events is incrementally

learned using tracker feedback for MOT.

In our approach, supporters, distractors or contextual events are not used because they make inference more complex and error-prone due to detection or tracking errors. We rather classify the general scene context and conditions in order to select the most appropriate visual cue or tracker for a given situation. To this end, we compute global image descriptors based on colour, intensity and motion at each video frame. In the past, other global image descriptors (sometimes called gist features) have been proposed (*e.g.* [298, 341, 367]) mostly for fixed images to classify scenes into different semantic categories, such as open, closed environments, indoor, outdoor *etc.*

6.4.3 Visual scene context description

One of our main goals is to globally characterise the scene at a given point in time through descriptors extracted from the image. For instance, we want to capture if the overall environment is rather dark or light, if it is cluttered, if the motion in the background is homogeneous *etc.* We also need to correlate these descriptors with the features used in the individual trackers from our pool of trackers $T_n, (n \in 1..N)$. We thus designed descriptors, called "scene features", based on first and second order statistics of the main image characteristics (*e.g.* intensity, hue, motion vectors). We defined these low-level features such that they can be easily interpreted, and, at the same time, help the process of learning and correlating a scene condition to a tracker.

Using Equations 6.12-6.15, we define our scene features f_k^Ω over an image region Ω as:

Intensity and texture features:

- *Average brightness* (f_1^Ω): the mean grey-scale pixel value (see Eq. 6.12).
- *Average contrast* (f_2^Ω): the mean squared value of the difference of each grey-scale pixel and the average brightness (see Eq. 6.15).

Chromatic features:

- *Average saturation* (f_3^Ω): the mean pixel value of the saturation channel in HSV colour space (Eq. 6.12).
- *Saturation variance* (f_4^Ω): the variance of saturation (Eq. 6.13).
- *Dominant hue* (f_5^Ω): the dominant colour extracted from a histogram of quantised hue pixel values in HSV colour space (Eq. 6.14).
- *Hue variance* (f_6^Ω): the variance of the pixel values in the hue channel (Eq. 6.13).

Motion features:

- *Average motion* (f_7^Ω): the mean of the norm of optical flow vectors (Eq. 6.12).
- *Motion variance* (f_8^Ω): the variance of the norm of dense optical flow vectors (Eq. 6.13).

Let p_i denote the pixel value of a given image channel (*e.g.* H,S,V) and $\|\Omega\|$ the number of pixels in the region Ω. The above mentioned features are then defined as follows:

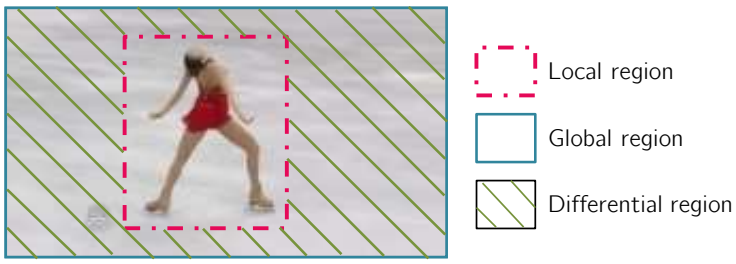


Figure 6.7: The different image regions used to compute scene features.

$$\text{AVERAGE:} \quad f_k^\Omega = \frac{\sum_{i\in\Omega} p_i}{\|\Omega\|}, \qquad \text{for } k = 1, 3, 7 \tag{6.12}$$

$$\text{VARIANCE:} \quad f_k^\Omega = \frac{\sum_{i\in\Omega} (p_i)^2}{\|\Omega\|} - \left(\frac{\sum_{i\in\Omega} p_i}{\|\Omega\|}\right)^2, \qquad \text{for } k = 4, 6, 8 \tag{6.13}$$

$$\text{DOMINANT CUE:} \quad f_k^\Omega = \underset{i\in\Omega}{\text{argmax}}(p_i), \qquad \text{for } k = 5 \tag{6.14}$$

$$\text{CONTRAST:} \quad f_k^\Omega = \frac{1}{\|\Omega\|} \sum_{i\in\Omega} \left(p_i - \frac{\sum_{i\in\Omega} p_i}{\|\Omega\|}\right)^2, \qquad \text{for } k = 2. \tag{6.15}$$

Each of these features k is computed on three different image regions Ω as shown in Fig.6.7. We define a **global value** as the feature computed on the whole image: f_k^G. The **local value** is the feature computed on the Region Of Interest (ROI): f_k^L. And a **differential value** as the difference between the feature computed on the foreground region (*i.e.* the ROI) and the background region (*i.e.* the image not including the ROI): f_k^D.
Not every combination of feature and region is used as some of them have little semantic meaning. The concatenation of these features gives us:

$$\begin{aligned} \mathbf{f}^G &= \left\{f_1^G, \ldots, f_3^G, f_6^G, \ldots, f_8^G\right\}, \\ \mathbf{f}^L &= \left\{f_1^L, \ldots, f_8^L\right\}, \\ \mathbf{f}^D &= \left\{f_1^D, \ldots, f_7^D\right\}. \end{aligned} \tag{6.16}$$

Finally, we obtain $M = 21$ scene context features $\mathbf{f}_t = \left\{\mathbf{f}_t^G, \mathbf{f}_t^L, \mathbf{f}_t^D\right\}$ for frame t.

6.4.4 A scene context-based tracking approach

6.4.4.1 Supervised learning of scene context

The scene context classifier's goal is to learn the different patterns that show high correlation between the information extracted from the scene context (described by our scene features) and the performance of a tracker in the particular set of conditions. As multi-class classifier, we chose a fully connected Multi-Layer Perceptron (MLP) with one hidden layer, N output neurons and sigmoid activation functions. Any other algorithm could be used as well. In fact, a multi-class SVM showed equivalent performance in our experiments.

As shown in Fig.6.8, the classifier's input $\mathbf{i}_t$ at a frame t consists of several components: The scene features $\mathbf{f}_t$ extracted from the scene characterise the scene, the *confidence* values of the

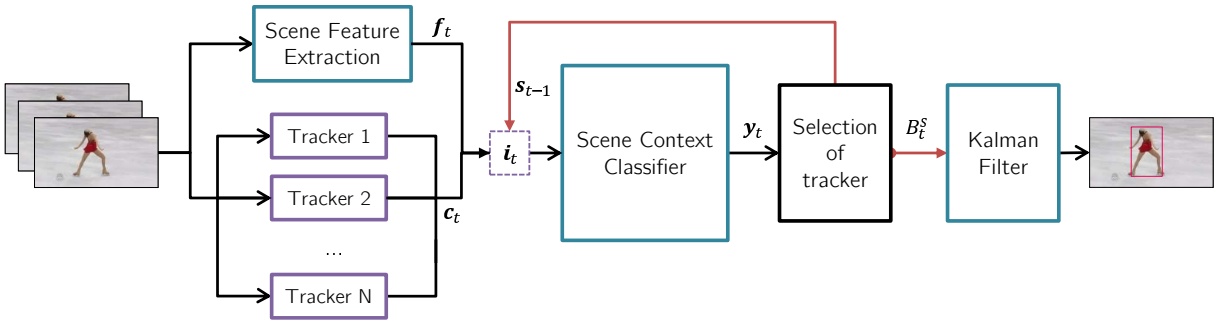


Figure 6.8: The overall framework of the proposed Scene Context-Based Tracker (SCBT)

N trackers $\mathbf{c}_t = (c_{t,1}..c_{t,N})$ providing the classifier with a measure of reliability of each tracker's result, and finally, the *identifier* s_{t-1} of the tracker that has been selected in the previous frame. We experimentally showed that this recursion highly contributes to learning the correlation between the scene context features and the selected tracker in a given frame.

Furthermore, in order to give the classifier information on the evolution of the scene context over time, we additionally provided it with the features from the two previous frames $t-1$ and $t-2$. We hence form a sliding window with the following vectors:

$$\mathbf{F}_t = \{\mathbf{f}_t, \mathbf{f}_{t-1}, \mathbf{f}_{t-2}\}$$
$$\mathbf{C}_t = \{\mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{c}_{t-2}\}$$
$$\mathbf{S}_{t-1} = \{s_{t-1}, s_{t-2}, s_{t-3}\}$$

And the final feature vector given as input to the classifier is the following:

$$\mathbf{i}_t = \{\mathbf{F}_t, \mathbf{C}_t, \mathbf{S}_{t-1}\}.$$

The classifier is trained off-line on a dataset with annotated object bounding boxes. Let us consider a training sample $\left\{\mathbf{i}_j, o_j^*\right\}$, $(j \in 1..N_{train})$ where N_{train} is number of training samples, $\mathbf{i}_j$ is the input vector and o_j^* is the *label* for the sample j, described below. In order to construct the classifier input vector $\mathbf{i}_j = \{\mathbf{F}_j, \mathbf{C}_j, \mathbf{S}_{j-1}\}$, we run the N trackers on each video, and at each frame (*i.e.* training sample j), we extract the scene context features $\mathbf{F}_j$ as well as the trackers' confidences $\mathbf{C}_j$ and save the identifier of best tracker to be used as the "previously selected tracker" $\mathbf{S}_{j-1}$ in the following frame.
The best tracker for each sample j, *i.e.* the label o_j^*, is determined by computing the F-scores of each tracker between its output bounding box and the ground truth.

The tracker with the highest F-score is considered the label o_j^* of the sample. We optimise the neural network parameters with standard stochastic gradient descent by minimising the mean squared error between the network's response vector $\mathbf{y}_j = (y_{j,1}..y_{j,N})$ and the desired output vector $\mathbf{y}_j^*$. We studied different ways of defining the desired classifier output $\mathbf{y}_t^*$:

- **One-of-N** : $\mathbf{y}_t^* \in \{-1, +1\}^N$ It is the usual output strategy in classification where $+1$ is assigned to the class label o_t^* and -1 otherwise.
- **Threshold** : $\mathbf{y}_t^* \in \{-1, 0, +1\}^N$ We threshold the F-score of each class and assign $+1$ if the F-score is higher than the threshold, 0 if it is lower, and -1 if the F-score is null (*i.e.* the tracker is completely lost). Here, the threshold value was set to 0.6, determined empirically.

- **Ranking** : $\mathbf{y}_t^* \in \{-1, -0.3, +0.3, +1\}^N$ We rank the F-scores of our classes, and assign respectively $+1$, $+0.3$ and -0.3 from highest to lowest F-score. If the F-score value is 0, than -1 is assigned to the corresponding class.

- **Regression** : $\mathbf{y}_t^* = \{Fscore_t^1, .., Fscore_t^N\}$ We directly use the F-score values, so the classifier trains to predict these values.

The network's final class prediction, *i.e.* the predicted best tracker index, is simply $o_j = \text{argmax}_{n \in N}\, \mathbf{y}_{j,n}$.

6.4.4.2 On-line tracker selection and learning

The proposed algorithm uses N independent on-line trackers that are initialised with the bounding box of the object in the first video frame. Then, as illustrated in Fig. 6.8, the trackers and the context feature extraction operate in parallel providing at each frame N confidence values $\mathbf{C}_t$ and M scene context features $\mathbf{F}_t$ respectively. At each video frame t, the scene context classifier estimates a score for each tracker $\mathbf{y}_t$ to perform best under the current scene context. We select the tracker with the highest score as the most suitable tracker $s_t = \text{argmax}_{n \in N}\, \mathbf{y}_{t,n}$. The bounding box B_t^s from the selected tracker T_t^s is then passed to a Kalman Filter to deal with imprecise estimations and to provide a smoother object trajectory. Finally, the last step is the *general update* of the trackers with the filtered bounding box. The individual trackers train their models on-line. They use the ground truth bounding box of the first frame to initialise their models and update them every frame once a prediction is made using the selected bounding box B_t^s processed by the Kalman filter.

6.4.5 Experiments

To evaluate our proposed approach we used a set of three trackers based on the Kernelized Correlation Filter (KCF) [182] and with three different types of complementary features: Histogram of Oriented Gradients (HOG), raw grey-scale intensities and quantised colours in the CIE-lab colour space. Note that our proposed method is completely independent of the underlying individual tracking algorithms.

We evaluated first the performance of the proposed scene context classifier based on the scene context features. To train the classifier we used the Princeton Tracking Benchmark Dataset [348], containing 100 videos, and a section of the ILSVRC2015 Dataset [325]. With a total of 397 videos (106 203 training samples and 12 700 validation samples), the dataset represents a diverse set of object types, background and scene conditions. For testing, we used the dataset of the VOT2013 benchmark [217]. The input features have been computed with the tracking output of the three KCF trackers. Table 6.6 shows the classification results for the different strategies for the output value $\mathbf{y}_t^*$. One can see the that, the classifier is able to predict the best tracker in around 80% of the video frames. The *One-of-N* training strategy gives the best result and *Ranking* the second best. The table shows also the overall tracking results of the proposed framework. Here, the *Ranking* strategy gives considerably better results.

Table 6.7 shows more detailed tracking results on VOT2013. The proposed tracker selection framework improves both accuracy and failure rate (robustness) compared to the individual trackers. Also the Kalman Filter and the general update, *i.e.* updating the individual models only using the bounding box of the selected tracker, increase the overall performance. More results as well as a comparison with the state-of-the-art can be found in [28].

Classifier output $\mathbf{y}_t^*$	Train rate	Test rate	Accuracy	Failures
One-of-N	89.56 %	79.56 %	0.583	1.313
Threshold	63.77 %	51.76 %	0,600	1,440
Ranking	88.95 %	77.82 %	**0.607**	**0.563**
Regression	69.80 %	62.24 %	0,590	1,130

Table 6.6: Correct classification rates on training (Princeton + ILSVRC15) and test (VOT2013) datasets for context classifiers trained using different desired output $\mathbf{y}_t^*$ strategies and the corresponding VOT2013 benchmark tracking results.

Method	Accuracy	Failures
KCF RAW	0.522	1.688
KCF HOG	0.590	0.875
KCF LAB	0.568	0.938
Context classifier	**0.607**	0.563
Context classifier + general Update	0.606	0.438
Context classifier + general Update + Kalman Filter	0.599	**0.375**

Table 6.7: VOT2013 benchmark[217] accuracy and failure rates for the individual KCF trackers and the proposed selection framework.

6.4.6 Conclusion

We proposed a novel selection framework that exploits scene context information in order to learn and predict the most suitable tracker. Using standard KCF trackers, we optimised the training of the scene context classifier by exploring multiple output strategies to select the most adapted one to our framework. We further evaluated the proposed framework on a standard benchmark proving the efficiency of the scene features and scene context classifier as well as the overall tracking framework. However, the tracking performance of our method is bounded by the performance of the individual trackers that are used. The proposed selection framework could be further improved by using more powerful individual tracker at the expense of computational efficiency and speed of the framework. Finally, adding new 'semantic' scene features that would characterise the type of object or type of scene would be an interesting future research direction.

6.5 Conclusion

This concludes the first part of this manuscript. We proposed several major contributions in the field of computer vision and machine learning applied to different problems of tracking a single and multiple objects in dynamic unconstrained environments. Our work was mainly focused on improving long-term MOT, especially multiple face tracking, by learning models for track initialisation and removal coping with the typical limitations of object detection algorithms (*i.e.* false and missing detection). Another focus was the design of robust and fast on-line learning and tracking algorithms coping with the challenging situations in unconstrained environments without any prior knowledge. And finally, we studied several approaches for effectively integrating visual scene context in the tracking process. Note that, we developed several complete tracking frameworks that were highly ranked in the very competitive international challenges VOT2014 and VOT2015, a challenge that is organised every year since 2013 as part of a workshop in conjunction with ECCV and ICCV.

PART II

Similarity metric learning and neural networks

7 Siamese Neural Networks for face and gesture recognition

7.1 Introduction

As shown by our research presented in the last part, and by many works in the literature in the last decades, discriminative approaches in machine learning are a very powerful tool in numerous Computer Vision problems, including visual object tracking. In this part, we will consider other applications that do not allow or are not suited for such supervised learning approaches. This is the case, for example, when:

- instance labels (*e.g.* positive/negative, foreground/background or a person identifier) are not available or too difficult to obtain for training or
- the number of classes is not fixed *a priori* or
- we want to model explicitly the relationship or similarity between instances and categories of instances.

If no information on instance classes or their relationships is given at all, *unsupervised approaches*, such as clustering methods, are most suitable to automatically learn and infer a general model of the data. However, in many settings, we have class labels for at least some of the training data, and we would like to learn a generic model that is applicable for all data of the same type. In this case, *weakly supervised* or *semi-supervised learning* algorithms are commonly employed.

One such weakly supervised approach is to automatically learn a similarity metric between instances of a given category (*e.g.* faces). That is, the instance labels are not explicitly learnt but rather used to model the distance[7] between similar and dissimilar instances – for example, by using pairs of instances.

In the work that I will present in this chapter, we followed this approach using Siamese Neural Networks (SNN). The original term "siamese" relates to the use of pairs of instances as introduced by Bromley *et al.* [99]. However, in the literature, this has been extended to triplet or tuple-based architectures, as in our approaches. As we will outline in the next section, there are other models, for example, based on statistical projections or Support Vector Machines. However, feed-forward neural networks have several properties that make them interesting and particularly suitable for the problems that we studied:

- They can model a wide variety of linear and non-linear functions, *c.f.* the universal approximation theorem [193], and well-established optimisation approaches can be used.

[7] sometimes the term *distance learning* is used but many existing models do not strictly fulfil the mathematical requirements of a distance, *e.g.* triangle inequality

- By carefully specifying the architectures, we can relatively easily "control" the complexity of the models (in terms of the number of parameters and at different abstraction levels, considering it as a data processing pipeline).
- Through multi-layered architectures and the error back-propagation algorithm, we can automatically construct models that simultaneously learn optimal features and projections into vector sub-spaces that best represent semantic similarities.
- Using multi-layer Convolutional Neural Network architectures, we can define models that are suitable and very powerful for (natural) image data.
- Finally, for a large variety of applications and data, neural networks showed a high generalisation capacity and robustness to different types of noise.

In the following, I will first outline the different algorithms and models that exist for similarity metric learning in the literature and describe the principal SNN approach. Then, I will present two major contributions that have been made to this field in the context of the PhD theses of Lilei Zheng [441] and Samuel Berlemont [90] that I have co-supervised. One is related to the definition of novel objective functions and learning strategies to improve the convergence for training and the performance at test time with applications to pairwise face verification. The other proposes a new SNN training framework with instance tuples to better condition the resulting similarity space applied to the problem of 3D gesture recognition using inertial data (from mobile phones).

7.2 Metric learning with Siamese Neural Networks

Most linear metric learning methods employ two types of metrics: the Mahalanobis distance or a more general similarity metric. In both cases, a linear transformation matrix W is learnt to project input features into a target space. Typically, distance metric learning relates to a Mahalanobis-like distance function [397, 410]: $d_W(x,y) = \sqrt{(x-y)^T W(x-y)}$, where x and y are two sample vectors, and W is not the covariance matrix as for the Mahalanobis distance but is to be learnt by the algorithm. Note that when W is the identity matrix, $d_W(x,y)$ is the Euclidean distance. In contrast, similarity metric learning methods learn a function of the following form: $s_W(x,y) = x^T W y / N(x,y)$, where $N(x,y)$ is a normalisation term [315]. Specifically, when $N(x,y) = 1$, $s_W(x,y)$ is the bilinear similarity function [108]; when $N(x,y) = \sqrt{x^T W x}\sqrt{y^T W y}$, $s_W(x,y)$ is the generalised cosine similarity function [186].

Non-linear metric learning methods are constructed by simply substituting the above linear projection with a non-linear transformation [121, 194, 210, 426]. For example, Hu et al. [194] and Chopra et al. [121] employed neural networks to accomplish this. These non-linear methods are subject to local optima and more inclined to overfit the training data but have the potential to outperform linear methods on some problems [85, 210]. Compared with linear models, non-linear models are usually preferred on a redundant training set to well capture the underlying distribution of the data [229]. A detailed survey and review of metric learning approaches has been published recently by Bellet et al. [85], and an experimental analysis and comparison by Moutafis et al. [284]. We will concentrate here on Siamese Neural Networks (SNN) that can represent linear or non-linear projections depending on the used activation function and number of layers.

A SNN essentially differentiates itself from classical feed-forward neural networks by its specific training strategy involving sets of samples labelled as similar or dissimilar. The capabilities

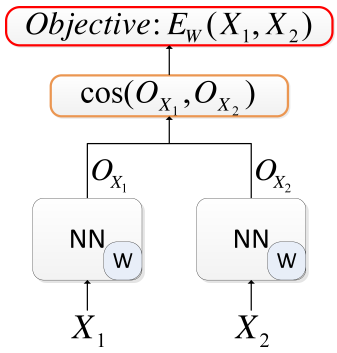


Figure 7.1: Original SNN training architecture.

of different SNN-based methods depend on four main points: the network architecture, the training set selection strategy, the objective function, and the training algorithm [99]. In the following, we will explain the three first points in more detail as they are most related to our contributions.

7.2.1 Architecture

A SNN can be seen as two identical, parallel neural networks *NN* sharing the same set of weights W (see Fig. 7.1). These sub-networks each receive an input sample $\mathbf{X}$, and produce output feature vectors $\mathbf{O}_X$ that are supposed to be *close* for samples from the same class and *far apart* for samples from different classes, according to some distance measure, such as the cosine similarity metric (*c.f.* section 7.2.3). During the training step, an objective function E_W, defined using the chosen distance measure over the output of all input sample combinations, is iteratively minimised.

Bromley *et al.* [99] introduced the Siamese architecture in 1994, using a Siamese CNN with two sub-networks for a signature verification system handling time-series of hand-crafted low-level features. In 2005, Chopra, Hadsell and LeCun [121] formalised the Siamese architecture applying a CNN on raw images for face verification, before adapting it to a dimensionality reduction technique [173]. More recently, Siamese CNNs have been used successfully for various tasks, such as person re-identification [419], speaker verification [112], and face identification [359].

CNN-based architectures are more specific to image inputs, and several research works propose to use feed-forward Multi-Layer Perceptrons (MLP) to handle more general vector inputs. For example, Yih *et al.* [421] apply SNNs to learn similarities on text data, Bordes *et al.* [97] on entities in Knowledge Bases, and Masci *et al.* [273] on multi-modal data. In this chapter, we will mostly focus on our work on MLP-based architectures and applications to face verification and gesture recognition. Whereas, in the following chapter, we will present our research on deeper CNN-based architectures applied to person re-identification in images.

7.2.2 Training Set Selection

The selection strategy for training examples depends mostly on the application and the kind of knowledge about similarities that one wants to incorporate in the model. For many applications, such as face or signature verification, the similarity between samples depend on their "real-world" origin, *i.e.* faces/signatures from the same person, and the neural network allows to determine the genuineness of a test sample w.r.t. a reference by means of a binary classification. Most

approaches use pairs of training samples $(\mathbf{X}_1, \mathbf{X}_2)$ and a binary similarity relation which takes different values for similar and dissimilar pairs. Lefebvre *et al.* [231] proposed to expand the information about the expected neighbourhood, and suggested a more symmetric representation: by considering a reference sample $\mathbf{X}_R$ for each known relation, it is possible to define triplets $(\mathbf{X}_R, \mathbf{X}_+, \mathbf{X}_-)$, with $\mathbf{X}_+$ forming a genuine pair with the reference $\mathbf{X}_R$, while $\mathbf{X}_-$ is a sample from another class – sometimes also called the *anchor*, the *positive* and the *negative* examples, respectively.

7.2.3 Objective Functions

The objective function computes a similarity metric between the higher-level features extracted from multiple input patterns. Minimising this function iteratively during training ensures that the distance between similar patterns gets smaller, and the one between dissimilar gets larger. In this regard, different metrics have been used in the literature:

Cosine pair-wise

Given a network with weights W and two samples $\mathbf{X}_1$ and $\mathbf{X}_2$ with their labels Y, a target $t(Y)$ is defined for the cosine value between the two respective output vectors $\mathbf{O}_{X_1}$ and $\mathbf{O}_{X_2}$ as "1" for similar pairs and "−1" (or "0") for dissimilar pairs [99]:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = (t(Y) - \cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2}))^2 \ . \tag{7.1}$$

A similar function is used in the Cosine Similarity Metric Learning (CSML) approach [186]:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = -t(Y)\cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2})\,. \tag{7.2}$$

Norm-based

Several works [121, 173, 273, 359] propose to use the norm, *e.g.* ℓ_2-norm, between the output vectors as a similarity measure:

$$d_W(\mathbf{X}_1, \mathbf{X}_2) = \|\mathbf{O}_{X_1} - \mathbf{O}_{X_2}\|_2 \ . \tag{7.3}$$

For example, Chopra *et al.* [121] define an objective composed of an "impostor" I (t(Y)=1) and a "genuine" G term (t(Y)=0):

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = (1 - t(Y))E_W^G(\mathbf{X}_1, \mathbf{X}_2) + t(Y).E_W^I(\mathbf{X}_1, \mathbf{X}_2) \tag{7.4}$$

$$\text{with } E_W^G(\mathbf{X}_1, \mathbf{X}_2) = \frac{2}{Q}(d_W)^2, E_W^I(\mathbf{X}_1, \mathbf{X}_2) = 2Qe^{(-\frac{2.77}{Q}d_W)} \ , \tag{7.5}$$

where Q is the upper bound of d_W.

Many recent works use the so-called *contrastive loss* [173], where

$$E_W^G(\mathbf{X}_1, \mathbf{X}_2) = d_W^2 \quad \text{and} \quad E_W^I(\mathbf{X}_1, \mathbf{X}_2) = \max(m - d_W^2, 0) \ , \tag{7.6}$$

with m being a fixed margin parameter.

Triplet

Weinberger et al. [397] introduced the triplet loss using simultaneously targets for genuine and impostor pairs by forming triplets of a reference $\mathbf{X}_R$, a positive $\mathbf{X}_+$ and a negative $\mathbf{X}_-$ sample:

$$E_W(\mathbf{X}_R, \mathbf{X}_+, \mathbf{X}_-) = \max(d_W(\mathbf{O}_R, \mathbf{O}_+)^2 - d_W(\mathbf{O}_R, \mathbf{O}_-)^2 + m, 0) \ . \tag{7.7}$$

Later, Lefebvre *et al.* [231] proposed a triplet similarity measure based on the cosine distance. Here, the output of the positive pair $(\mathbf{O}_R, \mathbf{O}_+)$ is trained to be collinear, whereas the output of the negative pair $(\mathbf{O}_R, \mathbf{O}_-)$ is trained to be orthogonal. Thus:

$$E_W(\mathbf{X}_R, \mathbf{X}_+, \mathbf{X}_-) = (1 - \cos(\mathbf{O}_R, \mathbf{O}_+))^2 + (0 - \cos(\mathbf{O}_R, \mathbf{O}_-))^2 \tag{7.8}$$

Deviance

Yi *et al.* [419] use the binomial deviance to define their objective function:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = \ln\left(\exp^{-2t(Y)\ \cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2})} + 1\right) \tag{7.9}$$

Two Pairs

Yih *et al.* [421] consider two pairs of vectors, $(\mathbf{X}_{p1}, \mathbf{X}_{q1})$ and $(\mathbf{X}_{p2}, \mathbf{X}_{q2})$, the first being known to have a higher similarity than the second. The main objective is then to maximise

$$\Delta = \cos(\mathbf{O}_{X_{p1}}, \mathbf{O}_{X_{q1}}) - \cos(\mathbf{O}_{X_{p2}}, \mathbf{O}_{X_{q2}}) \tag{7.10}$$

in a logistic loss function

$$E_W(\Delta) = \log(1 + \exp(-\gamma\Delta)) \ , \tag{7.11}$$

with γ being a scaling factor.

Probability-driven

Nair *et al.* [288] add a final unit to their neural network architecture whose activation function computes the probability P of two samples $\mathbf{X}_1$, $\mathbf{X}_2$ being from the same class:

$$P = \frac{1}{1 + \exp(-(w.\cos(\mathbf{O}_{X_1}, \mathbf{O}_{X_2}) + b))} \ , \tag{7.12}$$

with w and b being scalar parameters.

Statistical

Chen *et al.* [112] compute the first and second-order statistics, $\mu^{(i)}$ and $\Sigma^{(i)}$, over sliding windows on the SNN outputs of a speech sample i, and define the objective function as:

$$E_W(\mathbf{X}_1, \mathbf{X}_2, Y) = (1 - t(Y))(D_m + D_S) + t(Y).(\exp(\frac{-D_m}{\lambda_m}) + \exp(\frac{-D_S}{\lambda_S})) \ , \tag{7.13}$$

where

$$D_m = \left\|\mu^{(i)} - \mu^{(j)}\right\|_2^2 \quad , \quad D_S = \left\|\Sigma^{(i)} - \Sigma^{(j)}\right\|_F^2 \tag{7.14}$$

are incompatibility measures of these statistics between two samples i and j, λ_m and λ_s are tolerance bounds on these measures, and $\|.\|_F$ is the Frobenius norm.

We have made several contributions proposing novel training strategies and objective functions to more effectively train SNNs for different applications related to pairwise face verification and gesture classification. For example, the Triangular Similarity Metric and the Polar-Sine Metric, described in the following sections.

7.3 Triangular Similarity Metric Learning for pairwise verification

7.3.1 Introduction

In this section, I will describe our work [4, 8, 25–27] that has been performed in the context of the PhD thesis of Lilei Zheng [441], co-supervised with Christophe Garcia, Khalid Idrissi and Atilla Baskurt. This research focused on novel approaches for similarity metric learning with SNN applied to the problem of face verification.

Compared with the traditional identification task in which a decision of acceptance or rejection is made by comparing a sample to models (or templates) of each class [274, 320], pairwise verification is more challenging because of the difficulty of building robust models with enough training data for each class [196]. Often, only one training sample per class is available. Moreover, in current benchmark datasets, usually, the class identities in the training and test sets are mutually exclusive,*i.e.* there are no examples in the *training* set from a class figuring in the *test* set. The problem of pairwise face verification is to analyse two face images and decide whether they represent the same person or not. As mentioned above, it is usually assumed that neither of the face images shows a person from the training set.

In our work, we focused on Similarity Metric Learning approaches based on SNNs, which are particularly suitable for this type of problem. The models that we have studied are different linear and non-linear MLP architectures, trained with pairs of examples. Our main contributions have been: a new objective function for effective similarity metric learning with SNNs and a new training approach that only uses similar pairs of examples.

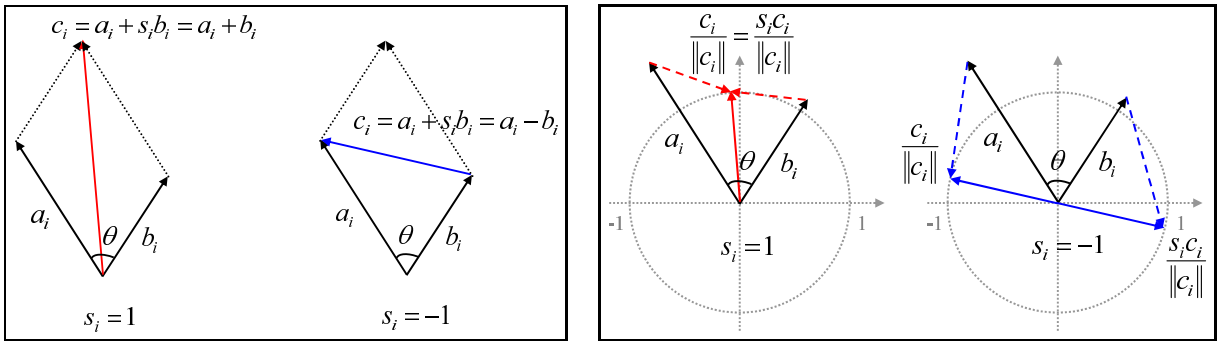
7.3.2 State of the art

One of the most used benchmark for face verification in the literature is "Labeled Faces in the Wild" (LFW) [196][8] that includes faces with varying poses and lighting conditions as well as facial expressions (see Fig. 7.3). It defines two evaluation protocols, one called "restricted" where only provided data can be used, and one called "unrestricted" where additional external data may be used for training.

In particular, for this last setting, recent deep learning techniques have approached the accuracy of 100% under the LFW evaluation standard. Almost all the published methods employed deep Convolutional Neural Networks (CNN) to process face images and to learn a robust face representation on additional large labelled training datasets, such as DeepFace from Facebook using the non-public SFC dataset [365]; DeepID [359, 360] using the CelebFaces dataset [258] [9] and the WDRef dataset [111]; FaceNet from Google using a 260-million image dataset [332] and the Tencent-BestImage commercial system using their BestImage Celebrities Face dataset[10].

However, with limited training data, these deep methods are more prone to overfitting and thus usually result in inferior performance for unseen test data. Under the restriction of no outside labelled training data, tremendous efforts have been put on developing robust face descriptors [62, 81, 111, 130, 197, 209, 243, 301, 328, 335, 342, 370, 402] and metric learning methods [81, 106, 131, 172, 186, 194, 398, 424]. Popular face descriptors include eigenfaces [370], Gabor wavelets [130], SIFT [209, 260], Local Binary Patterns (LBP) [62], *etc.* Especially, LBP and its variants, such as center-symmetric LBP (CSLBP) [180], multi-block LBP (MBLBP) [432],

[8] http://vis-www.cs.umass.edu/lfw/index.html
[9] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
[10] http://bestimage.qq.com/



(a) (b)

Figure 7.2: Geometrical interpretation of the TSML cost and gradient. (a) Minimising the cost means to make similar vectors parallel and make dissimilar vectors opposite. (b) The gradient function suggests unit vectors on the diagonals as targets for a_i and b_i: the same target vector for a similar pair ($s_i = 1$); or the opposite target vectors for a dissimilar pair ($s_i = -1$).

three patch LBP (TPLBP) [402] and over-complete LBP (OCLBP) [81], have proven to be effective for describing facial texture in images. Since face verification needs an appropriate way to measure the difference or similarity between two images, many researchers have been studying *metric learning* which aims at automatically specifying a metric from data pairs. For instance, Guillaumin *et al.* [171] proposed Logistic Discriminative Metric Learning (LDML) to model the probability of two vectors being similar by using a parametric *sigmoid function* with learnt parameters. Cao *et al.* [106] proposed to *simultaneously* learn a Mahalanobis distance-like metric and a bilinear similarity metric. They call their method Similarity Metric Learning over the Intra-personal Subspace (Sub-SML). The method introduced by Nguyen *et al.* [186] is called Cosine Similarity Metric Learning (CSML) and iteratively optimises a cosine-based objective function using similar and dissimilar pairs (*c.f.* Eq. 7.2). Finally, Chopra *et al.* [121] were the first to apply a SNN model to face verification.

Besides, other efforts have been made on face frontalisation (*i.e.* pose alignment) [89, 241, 418] or multiple descriptor fusion [66, 125, 301], in order to further improve the face verification performance.

7.3.3 Learning a more effective similarity metric

In the context of Lilei Zheng's PhD [441], we proposed a new approach for pairwise similarity learning called Triangular Similarity Metric Learning (TSML). We used a SNN with shared weights W that, at an iteration i, takes two inputs $\mathbf{X}_1$ and $\mathbf{X}_2$ and produces two outputs a_i and b_i ($\mathbf{O}_{X_1}$ and $\mathbf{O}_{X_2}$ in the previous section). The cost function of TSML is defined as:

$$E_W = \frac{1}{2}\|a_i\|^2 + \frac{1}{2}\|b_i\|^2 - \|c_i\| + 1, \tag{7.15}$$

where $c_i = a_i + s_i b_i$: c_i can be regarded as one of the two diagonals of the parallelogram formed by a_i and b_i (see Fig. 7.2(a)). Moreover, this cost function can be rewritten as:

$$E_W = \boxed{\frac{1}{2}(\|a_i\| - 1)^2 + \frac{1}{2}(\|b_i\| - 1)^2} + \boxed{\|a_i\| + \|b_i\| - \|c_i\|}. \tag{7.16}$$

We can see that minimising the first part aims at making the vectors a_i and b_i having unit length; the second part concerns the well-known *triangle inequality theorem*: the sum of the

Figure 7.3: Some challenging image pairs of same persons from the LFW dataset.

lengths of two sides of a triangle must always be greater than the length of the third side, *i.e.* $\|a_i\| + \|b_i\| - \|c_i\| > 0$. More interestingly, with the length constraints induced by the first part, minimising the second part is equivalent to minimising the angle θ between the vectors of a similar pair ($s_i = 1$) or maximising the angle θ between a dissimilar pair ($s_i = -1$), in other words, *minimising the Cosine Similarity* between a_i and $s_i b_i$ (*c.f.* Eq. 7.2).

In practice, this cost function can be minimised iteratively using Stochastic Gradient Descent (SGD) and the well-known gradient Backpropagation algorithm for multi-layer neural architectures. The gradient of the cost function (Eq. (7.15)) with respect to the parameters W is:

$$\frac{\partial E_W}{\partial W} = (a_i - \frac{c_i}{\|c_i\|})^T \frac{\partial a_i}{\partial W} + (b_i - \frac{s_i c_i}{\|c_i\|})^T \frac{\partial b_i}{\partial W}. \tag{7.17}$$

We can obtain the optimal cost at the zero gradient: $a_i - \frac{c_i}{\|c_i\|} = 0$ and $b_i - \frac{s_i c_i}{\|c_i\|} = 0$. That is, the gradient function has $\frac{c_i}{\|c_i\|}$ and $\frac{s_i c_i}{\|c_i\|}$ as targets for a_i and b_i, respectively. Figure 7.2(b) illustrates this: for a similar pair, a_i and b_i are mapped to the same target vector along the diagonal (the red solid line); for a dissimilar pair, a_i and b_i are mapped to opposite unit vectors along the other diagonal (the blue solid line). This perfectly reveals the objective of attracting similar pairs and separating dissimilar pairs. See [4] for more details on the optimisation procedure.

7.3.4 Experiments

7.3.4.1 Pairwise face verification results

As mentioned above, we used the LFW benchmark ('funnelled' version, *i.e.* aligned and cropped face images) to evaluate our proposed TSML approach. Both settings, the restricted and unrestricted protocol are evaluated. The restricted dataset contains fixed 300 positive and 300 negative pairs, whereas in the unrestricted case we generate more pairs from the same images using their identity labels. For the SNN, we considered three types of neural network architectures, "Linear" corresponding to a single-layer network without bias term, "Non-linear" with an additional tanh activation function and bias and "MLP" corresponding to a two-layer network (*i.e.* one hidden layer) with bias and tanh activation functions. We further experimented with a training strategy that uses *only similar pairs* in the optimisation, and no dissimilar pairs at all. The intuition behind this is that learning *dissimilarity* is challenging and not well conditioned as opposed to similarities. Thus, depending on the data, the negative term in the objective function related to dissimilarity may be partly contradictory to the positive one and might inhibit the learning of similarities during the training process. As training input, we used standard

Approaches	Restricted Training	Unrestricted Training
Baseline	84.83±0.38	
WCCN	91.10±0.45	91.17±0.36
TSML-Linear	87.95±0.40	92.03±0.38
TSML-Nonlinear	86.23±0.39	91.43±0.52
TSML-MLP	84.10±0.45	89.30±0.73
TSML-Linear-Sim	**91.90±0.52**	**92.40±0.48**
TSML-Nonlinear-Sim	90.58±0.52	91.47±0.37
TSML-MLP-Sim	88.98±0.64	89.03±0.58

Table 7.1: Mean maxDA scores (±standard error of the mean) of pairwise face verification by the TSML-based methods on the LFW-funneled image dataset. '-Sim' means training with similar pairs only.

state-of-the-art Fisher Vector (FV) features [307, 342] to describe the face images and reduce them to 500 dimensions using Whitened Principal Component Analysis (WPCA).

Like the minimal Decision Cost Function (minDCF) in [170], we defined a Decision Accuracy (DA) function to measure the overall verification performance on a set of data pairs:

$$DA(\gamma) = \frac{number\ of\ right\ decisions\ (\gamma)}{total\ number\ of\ pairs}, \tag{7.18}$$

where the threshold γ is used to make a decision on the final distance or similarity values: for the TSML system, $cos(a, b) > \gamma$ means (a, b) is a similar pair, otherwise it is dissimilar. The maximal DA (maxDA) over *all possible threshold values* is the final score recorded. We report the mean maxDA scores (±standard error of the mean) of the 10 experiments.

Table 7.1 shows the results for a baseline method, *i.e.* the maxDA scores directly computed on the FV data, the state-of-the-art method "Within Class Covariance Normalization" (WCCN) [81], and different variants of the proposed TSML approach. It can be noted that the unrestricted training gives better results, in general, as more training pairs are available. Surprisingly, *linear* models perform better than non-linear ones. This has been confirmed by several other experiments and is probably due to overfitting and the lack of training data. And finally, the methods that train the models on only similar pairs give superior results, and also outperform the other baseline approaches. Note that also WCCN does not model "dissimilarity" as it minimises only the *intra-class* variance and not the *inter-class* variance.

We further compared our approach to the state of the art. For a fair comparison, all tested methods use a *single* type of feature as input data. Better results could be obtained with a fusion of several types of features but, here, we wanted to focus on the different algorithms for similarity metric learning. Table 7.2 summarises the results on the LFW face verification benchmark (restricted setting). The proposed TSML approach outperforms all existing methods. Note that DDML (Discriminative Distance Metric Learning) is another method that we developed in the context of Lilei Zheng's thesis [4]. It uses a norm-based objective function and also gives excellent results, only slightly inferior to TSML.

Method	**Feature**	**Accuracy**
MRF-MLBP [65]	multi-scale LBP	79.08±0.14
APEM [241]	SIFT	81.88±0.94
APEM [241]	LBP	81.97±1.90
Eigen-PEP [243]	PEP	88.47±0.91
Hierarchical-PEP [240]	PEP	90.40±1.35
SVM [342]	Fisher Vector faces	87.47±1.49
DDML-Linear-Sim	Fisher Vector faces	91.03±0.61
WCCN [81]	Fisher Vector faces	91.10±0.45
TSML-Linear-Sim	Fisher Vector faces	**91.90±0.52**

Table 7.2: Comparison of TSML-Linear-Sim with other methods using a single type of face descriptor under the restricted configuration with no outside data on LFW-funneled.

Approaches	Restricted Training	Unrestricted Training
Baseline	87.78±0.39 / 0.1335	
WCCN	91.69±0.29 / 0.0900	91.97±0.33 / 0.0853
TSML-Linear	89.78±0.25 / 0.1108	93.97±0.20 / **0.0648**
TSML-Nonlinear	87.43±0.31 / 0.1340	93.11±0.20 / 0.0733
TSML-MLP	84.88±0.24 / 0.1592	90.21±0.36 / 0.1023
TSML-Linear-Sim	**92.94±0.15** / **0.0785**	**93.99±0.24** / 0.0662
TSML-Nonlinear-Sim	91.29±0.25 / 0.0918	93.43±0.23 / 0.0690
TSML-MLP-Sim	89.59±0.45 / 0.1093	90.83±0.30 / 0.0967

Table 7.3: Mean maxDA scores (±standard error of the mean) and mean EER of pairwise speaker verification by the TSML methods on the NIST i-vector speaker dataset. '-Sim' means training with similar pairs only.

7.3.4.2 Results on other applications

Speaker verification

We further evaluated our TSML approach on other pairwise verification problems – for instance, speaker verification using audio data. To this end, we used the data of the NIST 2014 Speaker i-Vector Challenge [170], which consist of i-vectors [132, 158, 232] derived from conversational telephone speech data in the NIST speaker recognition evaluations from 2004 to 2012. Each i-vector, the identity vector, is a vector of 600 components, designed to be characteristic for the voice of a given person. This dataset consists of a development set for building models and a test set for evaluation. Table 7.3 shows the average maxDA scores of the different variants of TSML with respect to the baseline and the state-of-the-art method WCCM. These results confirm the observations made for face verification, *i.e.* the linear models perform better, and training with only similar pairs improves the performance.

Kinship verification

Another similar problem is kinship verification from images, where the objective is to determine whether there is a kin relation between a pair of given face images. Lu *et al.* [261]

Team	F-S	F-D	M-S	M-D	Mean
Polito	84.00	82.20	84.80	81.20	83.10
LIRIS	89.40	83.60	86.20	85.00	**86.05**
ULPGC	85.40	75.80	75.60	81.60	80.00
NUAA	84.40	81.60	82.80	81.60	82.50
BIU	87.51	80.82	79.78	75.63	80.94
SILD (LBP)	78.20	70.00	71.20	67.80	71.80
SILD (HOG)	79.60	71.60	73.20	69.60	73.50

Table 7.4: Kinship verification accuracy (%) on KinFaceW-II under the restricted configuration in the FG 2015 Kinship Verification Evaluation.

constructed the Kinship Face in the Wild (KinFaceW) dataset for studying the problem of kinship verification from unconstrained face images mostly collected from the Internet, defining four types of relationships: Father-Son (F-S), Father-Daughter (F-D), Mother-Son (MS) and Mother-Daughter (M-D). Similar to LFW, the KinFaceW dataset holds the setting of limited training data for some classes and the setting of mutually exclusive training and test sets. There are two sub-sets: KinFaceW-I and KinFaceW-II, and besides their size, the major difference is that any two relative faces were acquired from different photos in KinFaceW-I but most relative faces in KinFaceW-II were captured from the same photo. In other words, environment conditions such as lighting differ more significantly between face pairs in KinFaceW-I than those in KinFaceW-II.

Using FV feature vectors as input, we evaluated TSML on this dataset, and, more importantly, we participated in an international competition on kinship verification that was organised by Lu *et al.* [25] in conjunction with FG 2015 and that used the KinFaceW dataset to benchmark the submitted methods. Table 7.4 shows the verification accuracy of the different submitted methods for KinFaceW-II in the restricted setting (ours is called LIRIS here). Our approach achieved the first place in this part of the competition. In the first part, we achieved the third place with a mean accuracy of 82.74% behind the method from "Politecnico di Torino" (Italy) (86.30%) and the one from "Nanjing University of Aeronautics and Astronautics" (China) (82.96), both proposing a feature selection approach using different types of features and an SVM classifier. The Side-information based linear discriminant analysis (SILD) [208] has been used as a baseline.

7.3.5 Conclusion

The proposed TSML method represents a generic SNN-based similarity metric learning framework that can operate with different models – linear and non-linear and with varying complexity, making it a powerful approach for weakly supervised learning for various applications and giving state-of-the-art results on different pairwise verification problems.

An interesting result of our studies is the fact that training with only similar pairs may improve the overall verification performance. Our further research work, presented in the following section, tries to better incorporate the negative pairs in the training process to build a more powerful model that also exploits these dissimilarities. This will especially help in the context of *classification*, which is the objective of the following work.

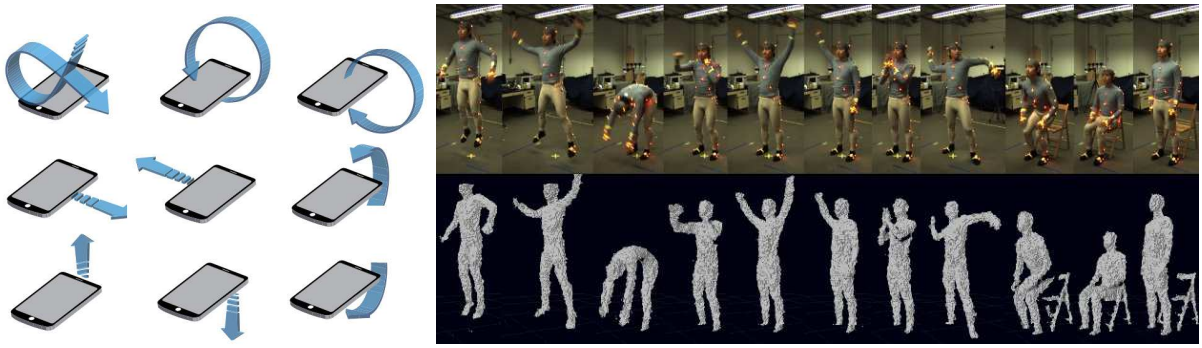


Figure 7.4: *Left:* illustrations of some symbolic gestures that are to be recognised from inertial sensor data from mobile devices. *Right:* examples from the Multimodal Human Action Dataset (MHAD).

7.4 Class-balanced Siamese Neural Networks for gesture and action recognition

7.4.1 Introduction

Our experiments showed that SNNs are able to model complex similarity metrics learnt from data. In the research work described in this section, that has been conducted as part of the PhD thesis of Samuel Berlemont [90] and that I have co-supervised with Christophe Garcia (LIRIS) and Grégoire Lefebvre (Orange Labs), we studied new approaches to incorporate negative, dissimilar examples in the training algorithm of SNNs, to improve the convergence and the resulting overall embedding in the similarity space. To this end, we proposed training algorithms that go beyond *pairwise* similarities and dissimilarities and operate on *tuples*. In addition, in this work, we were not concerned with *verification* but we concentrated on *classification* problems, *i.e.* we have data samples with labelled classes, and the classes are the same at training and test time. Thus, in principle, supervised learning approaches could be used in this context. However, there are advantages in modelling more explicitly the similarities and relationships between the different classes using a weakly supervised metric learning approach with SNNs. That is, not only the overall classification performance is improved, but also new samples from *unknown* classes can be more effectively *rejected*, as we will show in this section.

The developed algorithms have been evaluated on the problems of action recognition and on 3D symbolic gesture recognition using data from inertial sensors (accelerometers, gyroscopes *etc.*) commonly present in current mobile phones. Some of the gestures (*e.g.* flick east/west/north/south, pick, throw, heart, circle *etc.*) and activities (*e.g.* climbing, running, jumping *etc.*) are depicted in Fig. 7.4. The solution to this problem bears a certain number of challenges. On the one hand, inertial MicroElectroMechanicals Systems (MEMS) present inherent flaws that have to be taken into account, since they can be deceived by physical phenomena (*e.g.* electromagnetic interferences). On the other hand, in a real open-world application, inertial based gesture and action recognition also has to cope with high variations between users (*i.e.* right/left-handed users, dynamic/slow movements, *etc.*). Finally, to offer more functionality to final users, such a recognition system should propose a large vocabulary of possible interactions and reject all uncertain decisions and parasite or irrelevant motion.

In our work, we considered a gesture or action data example as a vector of fixed size, *i.e.* we worked with temporally segmented samples, and we performed a pre-processing step with a

certain number of traditional filtering, resampling and normalisation techniques to reduce the noise and to normalise the duration. More details on this can be found in [2, 33].

7.4.2 State of the art

Inertial gesture and action recognition

In the recent literature, three main strategies exist to deal with gesture and action recognition based on inertial data: probabilistic temporal signal modelling, temporal warping or statistical machine learning.

The probabilistic approach has mainly been studied with discrete [189, 207, 211] and continuous HMMs [314]. For instance, Kela *et al.* [211] used discrete HMMs (dHMM) from gesture velocity profiles. The first step is the input data space clustering in order to build a feature vector codebook. The second one consists in creating a discrete HMM using the sequences of vector codebook indexes. A correct recognition rate of 96.1% is obtained with 5 HMM states and a codebook size of 8 from 8 gestures realised by 37 users. In order to use gesture data correlation in time, Pylvänäinen [314] proposed a system based on a continuous HMM (cHMM) achieving a recognition rate of 96.76% on a dataset with 20 samples for 10 gestures realised by 7 persons.

The second approach is based on temporal warping from a set of reference gestures [64, 256, 401]. Liu *et al.* [256] presented a method using Dynamic Time Warping (DTW) from pre-processed signal data that gives gesture recognition and user identification rates of respectively 93.5% and 88%, outperforming in this study the HMM-based approach.

The third strategy is based on a specific classifier [188, 230, 404]. Hoffman *et al.* [188] propose a linear classifier and Adaboost, resulting in a recognition rate of 98% for 13 gestures performed by 17 participants. The study of Wu *et al.* [404] proposes to construct fixed-length feature vectors from the temporal input signal to be classified with Support Vector Machines (SVM). Each gesture is then segmented in time and statistical measures (mean, energy, entropy, standard deviation and correlation) are computed for each segment to form the final feature vectors. The resulting recognition rate is 95.21% for 12 gestures made by 10 individuals, outperforming in this study the DTW results. Finally, the recent study by Lefebvre *et al.* [230] proposes a method based on Bidirectional Long-Short-Term Memory Recurrent Neural Networks (BLSTM-RNN see [168]), which classifies sequences of raw MEM data with very good accuracy, outperforming classical HMM and DTW methods.

We also developed a inertial gesture classification method based on supervised learning and a specific Convolutional Neural Network (CNN) model. For our dataset of 14 symbolic gestures, we obtained a classification accuracy of up to 95.8% depending on the used test protocol outperforming other state-of-the-art methods. However, for the reasons mentioned above, *i.e.* better rejection of unknown classes and improved similarity metric space, we focused our following work on similarity metric learning approaches with SNN models.

Rejection approaches

The notion of rejection in classification has been studied in other areas and applications. Two kinds of rejection criteria have been proposed in the literature, with the first criterion based on the actual input to the classifier, and the second based on decision boundaries for the output space. Following the first strategy, Vasconcelos *et al.* [376], tackling handwritten digit recognition with a neural network-based classifier, suggested to use "guard units" for each class. These units are defined by their weight vector, which is composed by the means of the features for every training pattern belonging to the class. After activation of the network by a new sample,

the guard units check a similarity score between the input sample and each class, issuing a "0" output for neurons corresponding to the classes that do not meet the rejection criterion. For an input sample I and a weight vector W corresponding to the class of the sample, the scalar product $I \cdot W$ should be closer to the norm of W than the scalar product for a sample belonging to a different class. The rejection criterion is then defined by a threshold ρ, where the input is accepted by a class i only if $I \cdot W_i \geqq (W_i \cdot W_i - \rho)$.

The second strategy is more common, and can be subdivided into threshold-based and custom boundary determination methods. Fels *et al.* [145] applied a neural network model relying on 5 MLPs to a glove-based hand-gesture-to-speech system, Here, a thresholding strategy on the value of the highest softmax (neuron) output is adopted for the rejection of uncertain gestures. In [343], Singh *et al.* proposed an additional step to improve this rejection method. Applied to object recognition using a sequence of still images from the Minerva benchmark, their rejection criterion relies on synthetically generated patterns. For each feature, random numbers are sampled between $\mu - 2.5\sigma$ and $\mu + 2.5\sigma$ and removed if comprised between a minimum and maximum value. The generated patterns represent thus the outside boundaries of each class, and are trained to produce outputs close to zero for every class. Test samples are then classically rejected if all of their outputs are under a 0.5 threshold. A thresholding on the maximum output corresponds to a spheric reliability zone.

In order to define more flexible boundaries, Gasca *et al.* [159] proposed to estimate hyperplanes emulating the decision boundaries in the MLP output space in order to identify "overlap" regions, where the samples are more likely to be misclassified. The MLP is combined with a k-NN classification, based on the outputs of the training samples correctly classified after training. When recognizing a pattern, from the two nearest classes, the label is accepted only if the class given by the network matches the one selected by the k-NN, provided that the sample is not in the overlap area between hyperplanes.

7.4.3 Learning with tuples

As with the approaches presented in the previous section for pairwise verification, we proposed to use a SNN for similarity metric learning. The underlying model, that is non-linear here, was trained on gesture sample vectors, and the architecture of the neural network is an MLP with one hidden layer and sigmoid activation functions.

Cosine-based objective function

While Bromley *et al.* in [99] originally defined an objective function on separate positive and negative pairs (Eq. 7.1), whose number was arbitrary, Weinberger *et al.* [397] and Lefebvre *et al.* [231] proposed an error criterion based on triplets, with one reference example, one negative and one positive examples (Eq. 7.7 and 7.8). In order to keep symmetric roles for every class, we proposed a novel objective function defined over *sets* of training examples $\mathbf{T} = \{\mathbf{R}; \mathbf{P}_+; \mathbf{N}_1; ..; \mathbf{N}_K\}$ involving one reference example $\mathbf{R}$, one positive examples $\mathbf{P}_+$ and K negatives $\mathbf{N}_j$, one for each class (see Fig. 7.5). Thus, for a given set $\mathbf{T}_s$ in one training iteration, we define:

$$E_W(\mathbf{T}_s) = (1 - cos(\mathbf{O_R}, \mathbf{O_P}))^2 + \sum_{j,j \neq i} (0 - cos(\mathbf{O_R}, \mathbf{O_{N_j}}))^2 \ . \tag{7.19}$$

And the overall objective is:

$$E_W = \sum_s E_W(\mathbf{T}_s) \ , \tag{7.20}$$

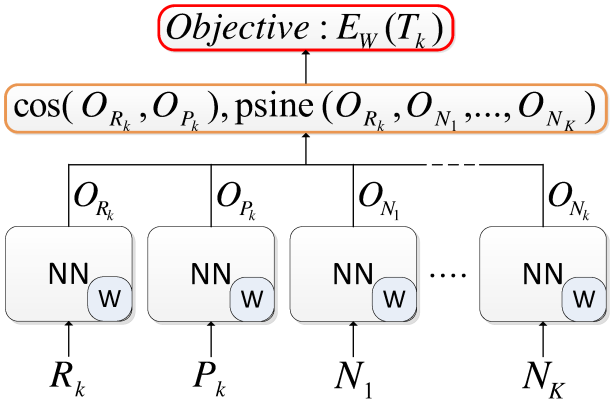


Figure 7.5: Proposed SNN training architecture with tuples.

for all possible tuples $\mathbf{T}_s$, which can be minimised iteratively, *e.g.* on randomly selected sets $\mathbf{T}_s$, using the Stochastic Gradient Descent and the gradient Backpropagation algorithm.

Norm regularisation

As with our TSML method, we wanted to further improve our objective function and convergence by introducing a normalisation and regularisation term. Cosine similarity-based objective functions are defined on the *angle* between the output vectors disregarding their lengths. Thus, introducing additional constraints on the vector norms helps to improve the convergence and to form a more stable embedding.

To this end, we first studied the behaviour of a weight update over the samples projected by the SNN. In the following, we will analyse the cosine metric as a function of two vectors of dimension n,

$$cos_{X_1,X_2} : \mathbb{R}^{2n} \rightarrow \mathbb{R} \,|\, (\mathbf{X}_1, \mathbf{X}_2) \rightarrow \frac{1}{2}(1 - \cos(\mathbf{X}_1, \mathbf{X}_2))^2. \tag{7.21}$$

Let $(\mathbf{O}_1, \mathbf{O}_2)$ be a pair of outputs used for update. Given the functions

$$\begin{aligned} cos_{O_1} &: \mathbb{R}^n \rightarrow \mathbb{R} \,|\, \mathbf{X} \rightarrow \frac{1}{2}(1 - \cos(\mathbf{O}_1, \mathbf{X}))^2 \\ cos_{O_2} &: \mathbb{R}^n \rightarrow \mathbb{R} \,|\, \mathbf{X} \rightarrow \frac{1}{2}(1 - \cos(\mathbf{X}, \mathbf{O}_2))^2 \end{aligned} \tag{7.22}$$

respectively evaluated at the points $\mathbf{O}_2$ and $\mathbf{O}_1$, the cos_{X_1,X_2} directional derivative at $(\mathbf{O}_1, \mathbf{O}_2)$ can be expressed as the concatenation of the two directional derivatives $\nabla_{cos_{O_1}}(\mathbf{O}_2)$ and $\nabla_{cos_{O_2}}(\mathbf{O}_1)$.

We showed that every gradient descent step will increase the norms for both samples. Considering the function cos_{O_1}, the update of $\mathbf{O}_2$ is

$$\mathbf{O}_2^{t+1} = \mathbf{O}_2^t - \lambda . \nabla_{cos_{O_1}}(\mathbf{O}_2), \lambda \in \mathbb{R}. \tag{7.23}$$

Figure 7.6 gives a graphical illustration in three dimensions. The line directed by the vector $\frac{\mathbf{O}_2}{\|\mathbf{O}_2\|}$ belongs to the equipotential for the $\cos_{O_1}$ function. By definition, we can conclude that the directional derivative $\nabla_{cos_{O_1}}(\mathbf{O}_2)$ is orthogonal to $\mathbf{O}_2$. According to Pythagoras' theorem, we can conclude:

$$\left\|\mathbf{O}_2^{t+1}\right\|^2 = \left\|\mathbf{O}_2^t\right\|^2 + \lambda^2 \left\|\nabla_{cos_{O_1}}(\mathbf{O}_2)\right\|^2 \Rightarrow \left\|\mathbf{O}_2^{t+1}\right\| > \left\|\mathbf{O}_2^t\right\|. \tag{7.24}$$

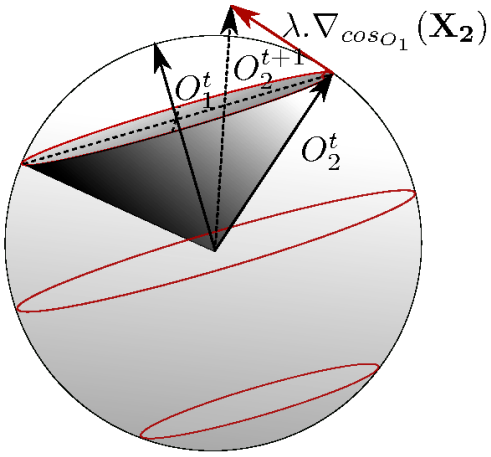


Figure 7.6: An update step on the projection norm for a pair $(\mathbf{O}_1^t, \mathbf{O}_2^t)$. The sphere centre corresponds to the origin. The grey cone represents the equipotential surface for the function cos_{O_1}.

Increasing the norms of the output vectors may incur progressive divergence. Moreover, with hyperbolic tangent activation functions, the output space is a hyper-cube of dimension n, which restricts the norms to a maximum of $\sqrt{n}$. Therefore, we proposed to add constraints on the norms of every output by forcing them to 1 and thus avoid any undesired saturation effects.

We modify our objective function E_{W_1} (see Eq.7.19) for a training subset $\mathbf{T}_s$:

$$E_{W_2}(\mathbf{T}_s) = E_{W_1}(\mathbf{T}_s) + \sum_{\mathbf{X}_p \in \mathbf{T}_s} (1 - \|\mathbf{O}_{X_p}\|)^2. \tag{7.25}$$

Given $\forall(\mathbf{O}_1, \mathbf{O}_2) \in (\mathbb{R}^n, \mathbb{R}^n)$, $\cos(\mathbf{O}_1, \mathbf{O}_2) = \frac{\mathbf{O}_1 \cdot \mathbf{O}_2}{\|\mathbf{O}_1\| . \|\mathbf{O}_2\|}$, we also propose to replace the cosine distance for each pair by the scalar product of the pair outputs, since the norms of the two outputs are set to one during training.

Thus, the final objective function for one training subset $\mathbf{T}_s$ is defined as:

$$E_W(\mathbf{T}_s) = (1 - \mathbf{O}_{R_k} \cdot \mathbf{O}_{P_k})^2 + \sum_{l \in \mathcal{N}_k} (0 - \mathbf{O}_{R_k} \cdot \mathbf{O}_{N_l})^2 + \sum_{\mathbf{X}_p \in \mathbf{T}_s} (1 - \|\mathbf{O}_{X_p}\|)^2. \tag{7.26}$$

This loss function can be minimised iteratively using Stochastic Gradient Descent and the gradient Backpropagation algorithm that takes into account the weight sharing and several activations by the different samples in each set $\mathbf{T}_s$.

Since the SNN is trained to evaluate similarities between multiple samples simultaneously, our assumption was that *unknown* samples are projected in a feature space in a coherent manner with *known* classes. To validate this hypothesis experimentally, we proposed a new SNN rejection strategy explained in the following.

Proposed rejection strategies

Once the SNN is trained, the output layer gives us a feature vector representing a similarity measure of a set of samples. For classification of samples, any standard classifier can be applied to these feature vectors. We choose a k-NN classification based on the cosine similarity metric in order to prove the validity and reliability of the learned projection. Finally, our rejection criterion consists in a single threshold, common to all classes, on the distance to the closest known sample. This same thresholding criterion is applied to another model based on DTW in order to get the

closest comparison possible and to a standard MLP, trained in a supervised way, with one output neuron per class and the well-known softmax activation function. Two types of rejection strategies are then studied. The first kind encompasses all incorrect classifications, and tests the ability of a system to identify samples whose classification is too uncertain to be accepted. The main challenge for the model is to isolate the misclassified samples first. The second kind of rejection concerns "unknown" classes, and aims at evaluating a model performance in isolating elements it was not trained for from the rest of the known classes. This type rejection is only rarely taken into account by existing methods, or is taken care of by another model specifically trained for this task. The main experimental results are summarised in section 7.4.5.

7.4.4 Polar sine-based objective function

While the regularised tuple-based formulation of the objective function of Eq. 7.26 is more suited to handle angular updates, it turns out impractical when the number of classes increases. Moreover, in the derivative of the mean squared error objective, the cosine error is weighted by the difference between the target and the cosine value which tends to zero and slows down the convergence. Thus, we proposed a new error function that preserves the targets, while addressing both of these problems. More specifically, we proposed a reformulation of the objective function based on a higher-dimensional dissimilarity measure, the polar sine metric.

Inspired by the 2D sine function, Lerman *et al.* [236] define the polar sine for a set $V_m = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of m-dimensional linearly independent vectors ($m > n$) as a normalized hyper-volume. Given $\mathbf{A} = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m\end{bmatrix}$ and its transpose $\mathbf{A}^\top$:

$$PolarSine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \frac{\sqrt{\det\left(\mathbf{A}^\top \mathbf{A}\right)}}{\prod_{i=1}^{n} \|\mathbf{v}_i\|}. \tag{7.27}$$

In the special case where $m = n$, the matrix product in the determinant is replaced by the square matrix $\mathbf{A}$.

Given the matrix $\mathbf{S}$ such that $\forall (i,j) \in [1,..,n]^2, \mathbf{S}_{i,j} = \cos(\mathbf{v}_i, \mathbf{v}_j)$, this measure can be rewritten as $PolarSine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \sqrt{\det(\mathbf{S})}$. For numerical stability reasons during the derivation process and to make this value independent from the number of classes, we introduced the polar sine metric:

$$psine(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \sqrt[n]{\det(\mathbf{S})}. \tag{7.28}$$

The polar sine metric only depends on the angles between every vector of the set. It reaches its maximum value when all the vectors are orthogonal, and thus can be used as a measure for dissimilarity.

With two comparable similarity estimators whose values are between 0 and 1, one for similar and one for dissimilar samples, it is now possible to redefine the objective function for our training sets $\mathbf{T}_s$:

$$\begin{aligned} E_{W_3}(\mathbf{T}_s) &= Esim_W(\mathbf{T}_s) + E\overline{sim}_W(\mathbf{T}_s)\ , \\ Esim_W(\mathbf{T}_s) &= (1 - \cos(\mathbf{O}_{R_k}, \mathbf{O}_{P_k}))^2, \\ E\overline{sim}_W(\mathbf{T}_s) &= (1 - psine(\mathbf{O}_{R_k}, \mathbf{O}_{N_1}, \ldots, \mathbf{O}_{N_K}))^2\ . \end{aligned} \tag{7.29}$$

Optimizing the polar sine corresponds to assigning a target of 0 to the cosine value of every pair of outputs from different vectors drawn in $\mathbf{T}_s \setminus \{\mathbf{R}_k\}$, i.e. we assign a target for every pair of dissimilar samples. This actually holds more information than our original cosine or regularised objective functions, which would only define a target for pairs including the reference sample. As a consequence, the *psine* function allows for a complete representation, in every training

set, of every available relationship present in the dataset. Given a fixed number of sources, the initial inputs are transformed into maximally independent, multi-dimensional components. Thus, our Siamese network, combined with this new objective function presents all the properties of a supervised, stochastic non-linear Independent Component Analysis, with an additional advantage: the number of components is adjustable by modifying the network output layer structure.

7.4.5 Experiments

We evaluated the proposed SNN approaches on several different datasets related to gesture and action recognition with inertial sensor data, and we compared it to a standard supervised MLP [126], a DTW-based model [126] and a SVM-based approach [109]. I will present here a summary of our results for two of the tested datasets.

Internal Orange Labs gesture dataset (DB1): this is one of the private dataset that we collected using an Android Samsung Nexus S mobile phone. It comprises 40 repetitions of 18 different classes (see Fig. 7.4) performed by a single individual, for a total of 720 records. After low-pass filtering, resampling and normalisation of the 3D accelerometer and gyroscope signals in 45 time steps, we obtain data vectors of fixed size $45 \times 6 = 270$ that we used for training and evaluating the different classification methods.

Multimodal Human Action Dataset (MHAD): this public dataset [294] comprises the recordings from 12 participants performing 11 different actions (with 5 repetitions) (see Fig. 7.4). Although there are multiple types of recorded sensors, we focus our study on two main sensors, namely the right wrist inertial sensor (A_1) and motion capture (M_{20}), as they gave the best classification results for all of the tested methods. We applied a similar pre-processing step as for the internal dataset DB1 and obtain data vectors of size $45 \times 3 = 145$.

The same network architecture is selected for every SNN variant: a 2-layer neural network, with an input size adapted to the data dimensionality (135 neurons for MHAD), a 45-neuron hidden layer, and a 90-neuron output layer. The hyperbolic tangent is chosen as the activation function for every neuron, and the learning rate is set to 0.001. During training, the network is independently and successively activated with every sample of a training set. Each activation state is stored, and reused to compute the weight updates. For more details refer to [2].

First, we studied the ability of a model to reject *unknown* gesture classes with the DB1 dataset. For training the model, 14 classes are used, with 5 repetitions per class. The test data comprises 16 repetitions from these 14 classes, as well as every record available from the 4 last unused classes, for a total of 224 records from known classes and 160 records from unknown classes (41.6%). The SNN is trained on tuples using the norm-regularised objective function of Eq. 7.26. Figure 7.7 depicts the curves showing the classification rates for DTW, MLP and the proposed SNN varying the rejection threshold. The SNN model presents a superior capacity to isolate unknown samples. Around the 41.6% landmark, where every unknown sample can be rejected, the SNN presents a correct classification rate of 94%, while the DTW and the MLP get lower scores of 92% and 88%, respectively. Furthermore, in its best configurations, depicted by the means of the deviation, the SNN is the closest to the perfect rate as the rejection rate increases.

Finally, we evaluated the performances of each SNN objective function variant with a tuple-based training selection strategy, denominated *cos-tuples* (*c.f.* Eq. 7.19), *norm-regularised* (*c.f.* Eq. 7.26) and *psine* (*c.f.* Eq. 7.29). The results for MHAD are shown in Table 7.5. For our analysis, we focus on the classification rates obtained from isolated sensor data with the ac-

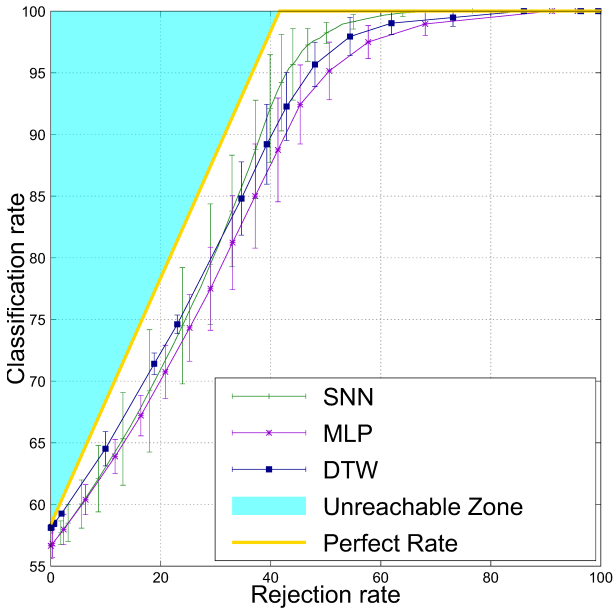


Figure 7.7: DTW, MLP and SNN comparison with the DB1 gesture dataset.

	A_1	M_{20}
DTW [126]	0.790 ± 0.107	0.888 ± 0.076
MLP [126]	0.818 ± 0.099	0.913 ± 0.067
SVM [109]	0.867	-
SNN cos-tuples (Eq. 7.19)	0.915 ± 0.102	0.906 ± 0.080
SNN norm-regularised (Eq. 7.26)	0.886 ± 0.079	0.910 ± 0.065
SNN psine (Eq. 7.29)	$\mathbf{0.918 \pm 0.091}$	$\mathbf{0.924 \pm 0.068}$

Table 7.5: Classification rates of the different SNN models and the state of the art on MHAD.

celerometer A_1 and the motion capture sensor M_{20}, and we report the results for the DTW and MLP methods proposed in [126] and the SVM-based approach [109] on these same sensors. The SNN-based approaches globally show superior results on the inertial sensor A_1, with a lowest score of 88.6% for the norm-regularised SNN, compared to a best score in the literature of 86.7% for the SVM approach. This shows that our SNN-based approach is very competitive. This conclusion is verified for the M_{20} sensor. Our SNN-psine approach, implementing the polar sine metric and tuple-based set selection strategies contributions, gives the best result of 92.4%.

7.4.6 Conclusion

In this section, we introduced new SNN similarity metric learning methods with objective functions operating on *tuples* that are well adapted for classification problems. Our contributions on the norm regularisation term and the polar sine improved the convergence of the models, the handling of rejection as well as the overall classification performance. We experimentally showed on gesture and action recognition applications that this approach can better cope with unknown classes and that the polar sine-based SNN outperforms state-of-the-art methods in terms of the classification rate.

7.5 Conclusion

This concludes the chapter on similarity metric learning with SNN for face and gesture verification and classification. The work described here mainly focused on improving the convergence and learning strategies for SNN based on MLP models in the context of pair-wise verification and classification. The neural network models that we employed were rather *shallow*, although we successfully conducted some experiments on TSML with deep CNNs for face verification [4]. This is mainly due to the relatively small amount of annotated training data that is available for the studied applications. Deeper, more complex models would have easily overfit to the training data. Even with a strong regularisation, deep neural networks only show their full potential with large amounts of training data, and, in that case, mostly CNN-based models are used for learning deep visual feature hierarchies which is not appropriate for other types of data like inertial signals. Finally, these relatively simple models allowed us to better understand the behaviour of convergence and of the learnt linear and non-linear projections with respect to different hyper-parameters.

In the following chapter, I will present our work on person re-identification from images. For this application, more training data is available, and we used deeper and more complex models and neural network architectures in order to perform similarity metric learning and ranking of pedestrian images.

8 Deep similarity metric learning and ranking for person re-identification

8.1 Introduction

In this chapter, I will present the work conducted in the context of the PhD thesis of Yiqiang Chen [116] co-supervised with Atilla Baskurt (LIRIS) and Jean-Yves Dufour (Thales Services, ThereSIS). We proposed several contributions on similarity metric learning with deep neural network models applied to the problem of person re-identification in images. The application

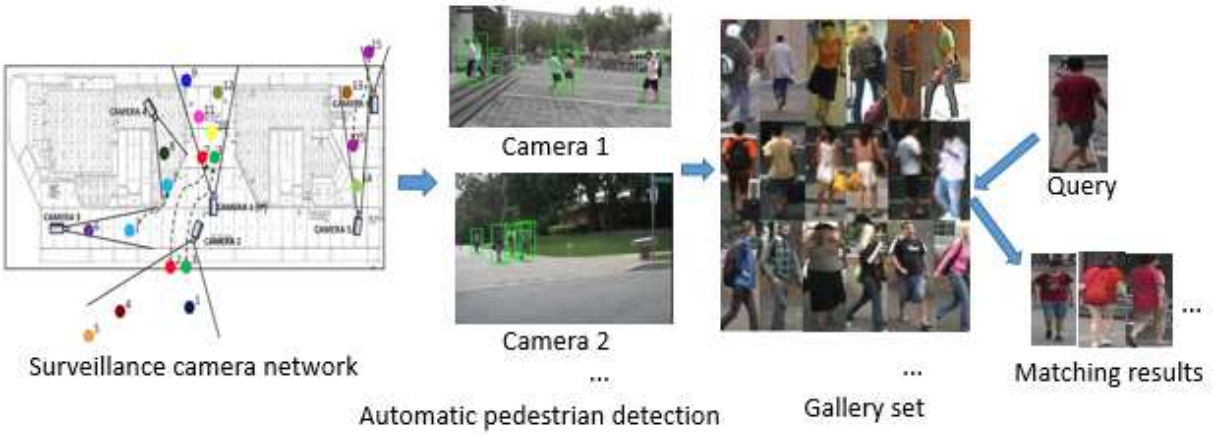


Figure 8.1: Illustration of person re-identification in video streams with non-overlapping views.

and problem is illustrated in Fig. 8.1. A network of cameras with non-overlapping views provides continuous video streams where the appearing persons should be recognised within the same or across different views and at different points in time. That is, given a query image of a person from one view (*i.e.* the probe), the goal is to re-identify this person in images from the same or from other views (*i.e.* called the gallery images).

Here, we did not work on the person detection and tracking (MOT) problem, and we considered that efficient algorithms for this are available providing us with pedestrian images that are cropped but whose identities are not matched (as illustrated in the right of Fig. 8.1).

This task bears a certain number of scientific challenges in the fields of computer vision and machine learning some of which are illustrated in Fig. 8.2. For example, the intra-class variations can be very large due to different lighting conditions, different view points and body

Figure 8.2: Examples of some person re-identification challenges. Each pair of images shows the same person except (g) and (h).

poses. Also, the images maybe of very low resolution and partial occlusions may occur frequently. Although, we assume that clothing does not change too much in a limited time frame, this may still happen when people take off a coat or a hat or carry a bag in a different way. Also, the inter-class variation may be very small since people tend to wear similar clothes. From a machine learning point of view, an effective model is required that is robust to the above variations and that generalises well to persons (*i.e.* classes) and possibly view points that it has not been trained for. Further, in realistic settings a query image needs to be matched to images in a very large gallery set. To build such models using statistical machine learning approaches, often the training data is very limited, and their annotation is difficult and laborious.

We employed deep neural network models for this task, because of their capacity of automatically and jointly learning robust visual features and classifiers. To prevent them from overfitting in this difficult setting, we used different techniques to combine supervised and weakly supervised learning and to include external data related to more semantic information for the given application, like semantic pedestrian attributes, body orientation or group context. I will describe these different contributions in the following.

8.2 State of the art

Approaches for person re-identification are generally composed of an appearance descriptor to represent the person and a matching function to compare those appearance descriptors. Over the years, numerous contributions have been made to improve both the representation as well as the matching algorithm in order to increase robustness to the variations in pose, lighting, and background inherent to the problem. In the literature, person re-identification is mostly performed using the person appearance in a single colour image. However, some approaches also use temporal information, depth images, gait, camera topology *etc.*

8.2.1 Feature extraction approaches

The appearance of pedestrians from static images can be characterised according to three aspects: colour, shape, and texture. Colour histograms are widely used to characterise colour distributions. Also, some photometric transformation or normalisation methods are proposed: for example, Porikli *et al.* [311] learned a brightness transfer function, Bak *et al.* [77] applied a histogram equalisation technique and Liao *et al.* [250] applied the Retinex algorithm to pre-

Figure 8.3: Illustrations of representative approaches of three feature extraction strategies. (a) Patch-based descriptors extracted from a dense grid [337]. (b) Descriptors extracted from segmented body parts [118]. (c) Stripe-based descriptors extracted from horizontal bands [416].

process person images. Therefore, colour features are often combined with shape and texture features, such as Gabor filter banks [148], Scale-invariant feature transform (SIFT) [259], Local Binary Pattern (LBP) [296], and region covariance [371].

In order to get a feature descriptor which is discriminant, and at same time, robust to the different variations, various extraction strategies have been proposed in the literature. Here we divide the approaches into three classes: patch-based descriptors and body part-based descriptors and stripe-based descriptors, as shown in Fig. 8.3.

Patch-based descriptors

The holistic representation of the above-mentioned global features shows a high robustness but a relatively low discriminative power, because of losing local detail information. A typical solution is to apply the colour histograms and texture filters on a dense grid. For example, the approach of Zhao et al. [439] computes LAB colour histograms and SIFT features on a grid of 10×10 overlapping patches at two different scales. Similarly, Liu *et al.* [257] extract the HSV histogram, gradient histogram and the LBP histogram for each local patch. Then they applied a technique called "local coordinate coding" which is a high-dimensional non-linear learning method projecting the data on lower-dimensional manifolds. The Bag-of-Words (BoW) model is used in [440] with 11-dimensional colour name descriptors [416] extracted for each local patch and aggregated into a global vector. After generating the codebook on training data, the feature responses of each patch are then quantified into visual words and a visual word histogram is used for the matching. Also, Shen *et al.* [337] extract Dense SIFT and a Dense Colour Histogram from each patch and proposed a specific patch matching process with global spatial constraints.

Body part-based approaches

A inherent problem with patch-based methods operating on a fixed grid is that they are sensitive to misalignment due to pose and viewpoint variations. In order to resolve this issue and increase the discriminative power, several approaches exploit the prior knowledge of the person geometry or body structure and try to partition the image appropriately to obtain a pose-invariant representation: For example, Wang *et al.* [394] segment different (roughly uniform) image regions using local HOG descriptors and model the context of appearance and shape by a co-occurrence matrix over these image regions. Some approaches segment images into meaningful parts like torso, legs, which are semantic and more robust to the viewpoint variation. One well-know method is Symmetry-Driven Accumulation of Local Features (SDALF) proposed by Farenzena *et al.* [144] which exploits symmetry and asymmetry principles for segmentation and uses statistical measures on the body part regions to describe them and perform the matching.

Bak *et al.* [77] proposed an approach based on spatial covariance regions which are segmented using 5 HOG-based body part detectors. To better exploit the prior knowledge of plausible body part configurations, Cheng *et al.* [118] used Pictorial Structures to localise the body parts and match their descriptors based on HSV histograms and Maximally Stable Colour Regions (MSCR).

Stripe-based approaches

In most settings, pedestrians are seen from an arbitrary horizontal viewpoint. Thus, some methods extract features on horizontal stripes and are thus invariant to large horizontal shifts in the image. Gray *et al.* [169] first proposed to divide the pedestrian image into 6 equally-sized horizontal stripes. The approximately correspond to the image regions of the head, upper and lower torso, upper and lower legs and fee. In each stripe, 8 colour channels (RGB, HS, YCbCr) and 19 texture channels (Gabor and Schmid filter banks) are represented. Similarly, Mignon *et al.* [278] build the feature vector from RGB, YUV and HSV channels and the LBP texture histograms in horizontal stripes. Yang *et al.* [416] also proposed to use 6 stripes and introduced the salient colour name-based colour descriptor (SCNCD) for pedestrian colour descriptions. The approach proposed by Ma et al. [264] computes covariance matrices on Gabor filter responses at different scales and on different image bands, and the difference of these matrices on consecutive bands is used to build the model. Finally, Liao *et al.* [250] proposed LOcal Maximal Occurrence (LOMO) features that we also used in our work. Scale-Invariant Local Ternary Patterns (SILTP) and HSV histograms are extracted on each line of the image at different scales, and then only the maximum value is retained for each line.

8.2.2 Matching approaches

Based on the extracted features, we can distinguish two types of matching methods. The first consists of learning a matching function in a supervised manner, and the other learns a distance metric in feature space.

Matching function learning

Given feature based representations of a pair of images, an intuitive approach is to compute the geodesic distance between the descriptors, for instance, using the Bhattacharyya distance between the histogram-based descriptors or the L2-norm between descriptors in a Euclidean space. However, some features may be more relevant for appearance matching than others. Therefore, several approaches have been proposed to learn a matching function in a supervised manner from a dataset of image pairs.

For instance, the method of Schwartz and Davis [334] transforms the high-dimensional features into low-dimensional discriminant vectors using Partial Least Squares (PLS) in a one-against-all scheme. Lin *et al.* [251] proposed an approach based on the Kullback-Leibler divergence of feature distributions of two images and pairwise dissimilarity profiles learnt from training data. Gray *et al.* [169] use boosting to find the best ensemble of localised features for matching. And Prosser *et al.* [313] proposed an ensemble of RankSVMs to solve person re-identification as a ranking problem.

Metric learning

Compared to standard generic distance measures, *e.g.* the Euclidean or Bhattacharyya distance, a metric that is learnt specifically for person images is more discriminative for the given task of re-identification and more robust to large variations of person images across views.

Most distance metrics learning approaches learn a Mahalanobis-like distance: $D_2(x, y) = (x-y)^T M(x-y)$ where M is a positive semi-definite (PSD) matrix of which the elements are to

be learnt. Several works factorise M as $M = w^T w$, ensuring the PSD constraint and implicitly defining a (potentially low-dimensional) projection into an Euclidean space which reflects the distance constraints. We distinguish two types of methods explained in the following.

The first class of methods generally defines an objective function based on distance constraints. The global idea of constraints is to keep all the vectors of the same class closer while pushing vectors of different classes further apart. M is solved by a constrained convex optimisation method. For example, Mignon *et al.* [278] presented a method called Pairwise Constrained Component Analysis (PCCA), and Zheng et al. [443] proposed the Probabilistic Relative Distance Comparison (PRDC) model, both defining pair-wise constraints. In contrast, the approach of Dikmen et al. [134] called Large Margin Nearest Neighbour classification with Rejection (LMNN-R) operates on constraints that are defined on the neighbourhood of examples. To avoid the tedious iterative optimisation procedure, Koestinger *et al.* [216], in their "Keep It Simple and Straightforward Method" (KISSME), propose a formulation that allows for a closed-form solution of the matrix M.

The methods of the second class are generally variants of the Linear Discriminative Analysis (LDA). The approaches are based on the difference of the feature vectors of two classes. The positive class consists in pairs of images of the same person acquired by different cameras, and the negative class consists in pairs of images of different person acquired by different cameras. They learn directly the projection $\mathbf{w}$ to a discriminative low-dimensional subspace where the between-class variance is maximised and the within-class variance is minimised. In LDA, the objective function is formulated as: $J(w) = \frac{w^T S_b w}{w^T S_w w}$, where S_b and S_w are the between-class and within-class scatter matrices, respectively. Methods belonging to this class are LFDA proposed by Pedagadi *et al.* [303] combining LDA and Locality-Preserving Projection for dimensionality reduction, the Cross Quadratic Discriminitive Analysis (XQDA) metric by Liao et al. [250] combining Quadratic Discriminative Analysis (QDA) and KISSME and the Null Foley-Sammon Transform (NFST) by Zhang et al. [434].

8.2.3 Deep learning approaches

Methods based on deep neural networks jointly learn discriminative features and a matching function or a similarity metric. Several approaches have been presented in the literature.

8.2.3.1 Architectures

Some deep learning architectures are conceived to operate on stripes over the input image. For example, Yi *et al.* [420] first applied a CNN on re-identification. Given two person images, they are first separated into three over-lapped horizontal stripes respectively, and the image pairs are matched by three Siamese Convolutional Neural Networks (SCNN). Varior *et al.* [374] proposed to integrate a gating layer in a SCNN to compare the extracted local patterns for an image pair at the medium-level and propagate more relevant features to the higher layers of the network. Spatial dependency between stripes are exploited in [375] by a Long Short Term Memory (LSTM) network operating on LOMO [250] and SCNCD [416] features extracted on horizontal stripes. Cheng *et al.* [117] proposed to combine the global and stripe feature extraction using a multi-channel CNN model operating on different parts of the image.

Some methods directly integrate some kind of patch matching into a CNN architecture to handle misalignment and geometric transformations. For instance, Li *et al.* [247] proposed an architecture called Filter pairing neural network directly performing the patch matching using integrated displacement matrices. However, this architecture is computationally complex and

has thus been simplified by Ahmed et al. [61] introducing a cross-input neighbourhood difference layer.

Other methods integrate a body part-based feature extraction within a deep neural network architecture. Zhao *et al.* [437], for example, proposed an architecture composed of three components: a body region proposal network (locating human body joints), a feature extraction network and a feature fusion network. However, this requires training data with annotated body parts. In contrast, Li *et al.* [239] proposed to localise latent pedestrian parts through Spatial Transform Networks (STN) [199]. Zhao *et al.* [438] proposed to jointly model the human body regions that are discriminative for person matching with neither prior knowledge nor labelled data and compute a compact representation. An image feature map is first extracted by a deep Fully Convolutional Neural network (FCN), and discriminative features are automatically detected and extracted in several network branches to perform the matching, which has been trained using a siamese structure and triplets of examples.

8.2.3.2 Objective functions

As we have seen in the previous chapter, by appropriately defining objective functions, one can learn a non-linear projection into a feature space in which the similarity of pedestrian is well represented. In the following, we will rather use the term "loss function" as it is common in the recent literature for deep neural networks since there is often no regularisation term. Several loss functions for person re-identification exist in the literature. Yi *et al.* [420] used the deviance loss in a Siamese network, as follows:

$$E_{deviance} = ln(e^{-2sl} + 1), \tag{8.1}$$

where $-1 \leqslant s \leqslant 1$ is the similarity score and $l = 1$ or -1 is the label (*c.f.* Eq. 7.9). And in [61, 247], the re-identification task is considered as an image pair classification problem deciding whether an image pair is from the same person or not. Ding *et al.* [135] first applied the triplet loss (*c.f.* Eq. 7.7) to train a CNN for person re-identification. Some methods combine different objective functions to improve the performance. For example, Cheng *et al.* [117] proposed an improved variant of the triplet loss function combining it with the contrastive loss. Zheng *et al.* [444] combined an image pair classification loss and the contrastive loss (*c.f.* Eq. 7.4,7.6). And Chen *et al.* [113] applied a quadruplet loss which samples four images from three identities and minimises the difference between a positive pair from one identity and a negative pair from two different identities and they combine this quadruplet loss with the triplet loss.

8.2.4 Evaluation measures

The Cumulated Matching Characteristics (CMC) curve and the mean Average Precision (mAP) are the two most widely used evaluation measures for person re-identification.

CMC evaluates the top n nearest images in the gallery set with respect to one probe image. If a correct match of a query image is at the k^{th} position ($k <= n$), then this query is considered as success of rank n. A curve can thus be drawn for varying n, with n in the x-axis and the proportion of correct matches in the y-axis. Also, we denote the CMC curve value at a given rank n the "*Rank n* score".

mAP is the mean value of the average precision of all queries. The average precision is defined as the area under the Precision-Recall curve.

8.3 Leveraging additional semantic information – attributes

8.3.1 Introduction

As in previous work in the literature [117, 135, 419], we proposed to tackle the person re-identification problem with a similarity metric learning approach based on SNNs since the classes are not known *a priori* and since we would like to directly learn from data a generic (non-linear) metric between image pairs that robustly expresses the similarities (and dissimilarities) of persons. Re-identification of a given query image is then performed by computing this metric on all images in the gallery set and returning the one(s) with the smallest distance.

Our first contribution consisted in building a SNN-based model that is trained on triplets and that incorporates semantic attributes in order to assist the re-identification improving its performance [3, 15]. The idea is that such mid-level attributes, like gender, accessories and clothing, represent characteristic features that are highly invariant to view point, body pose and other image variations. Moreover, they could be used for so-called "zero-shot" identification, *i.e.* querying by an attribute-based description instead of an image. Since biometric features like faces are often not visible or of too low resolution to be helpful in surveillance, pedestrian attributes could be considered as soft-biometrics and provide additional discriminant information.

Our proposed approach first trains a multi-class CNN in a supervised way to recognise a set of pre-defined person attributes from images and then combines this neural network with a SNN trained in a weakly supervised way on triplets to model a similarity metric between person images.

8.3.2 State of the art

8.3.2.1 Attribute recognition

In the pioneering work of Vaquero *et al.* [373], the person image is segmented into regions, and each region is associated with a classifier based on Haar-like features and dominant colours. Layne *et al.* [227] annotated 15 attributes on the VIPeR dataset and proposed an approach to extract a 2784-dimensional low-level colour and texture feature vector for each image and to train an SVM for each attribute. The method of Zhu *et al.* [448] determines the upper and lower body regions according to the average image and extracts colour and texture features (HSV, MB-LBP, HOG) in these two regions. Then, an Adaboost classifier is trained on these features to recognise attributes. The drawback of these approaches is that all attributes are treated independently. That is, the relation between different attributes is not taken into account. Zhu *et al.* [449] tried to overcome this limitation by introducing an interaction model based on their Adaboost approach [448] learning an attribute interaction regressor. The approach from Deng *et al.* [133] used a Markov Random Field (MRF) for this purpose.

Some CNN models have also been proposed for pedestrian attribute recognition, *e.g.* the one by Li *et al.* [238] based on CaffeNet and Sudowe *et al.* [358] called Attribute Convolutional Net (ACN). Zhu *et al.* [450, 451] proposed to divide the pedestrian images into 15 overlapping parts where each part connects to several CNN pipelines with several convolution and pooling layers.

8.3.2.2 Re-identification with attributes

Some works have used pedestrian attributes to assist with the re-identification task, *e.g.* the SVM-based approach of Layne *et al.* [227]. In their approach, the final distance between two pedestrian images is computed as a weighted sum of low-level feature distance and attribute

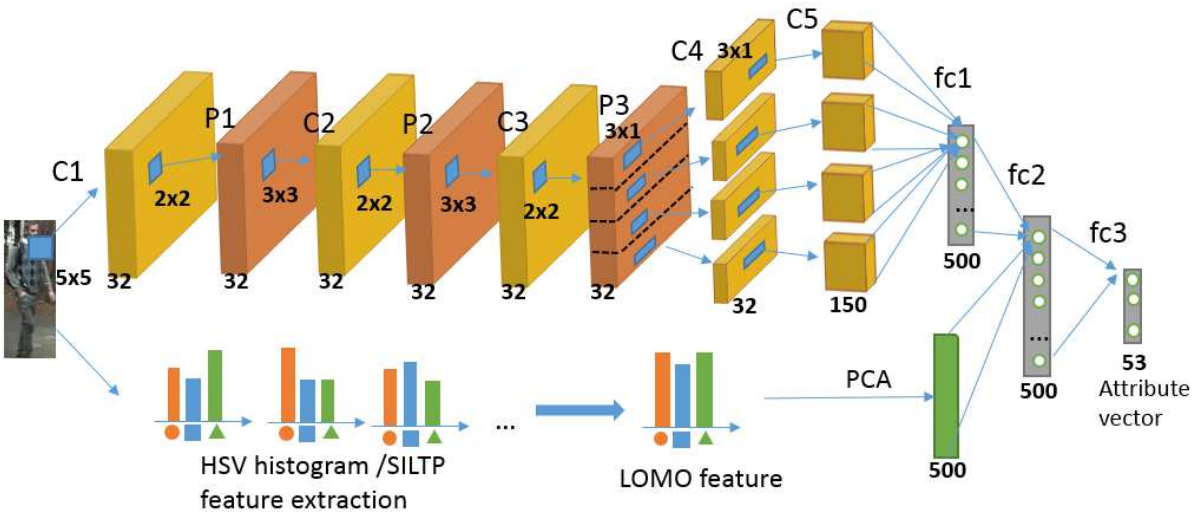


Figure 8.4: Overview of our pedestrian attribute recognition approach.

distance. Li *et al.* [237] embedded middle-level clothing attributes via a latent SVM framework for more robust person re-identification. The approach introduced by Khamis *et al.* [213] learns a discriminative projection into a joint appearance-attribute subspace in order to leverage the interaction between attributes and appearance for matching.

The approach of Zhu *et al.* [450] recognises attributes with deep neural networks then calculates a pedestrian distance by weighting the attribute distance and a low-level feature-based person appearance distance. McLaughlin *et al.* [276] proposed to perform person re-identification and attribute recognition in a multi-task learning. Their loss function is a weighted sum of the attribute and identification classification loss as well as a Siamese loss. They show that this multi-task joint learning improves the re-identification performance. Matsukawa *et al.* [275] proposed to fine-tune the well-known Alexnet with attribute combination labels to increase the discriminative power. Further they concatenated the CNN embedding directly with LOMO features and used the metric learning method XQDA [250] to learn a feature space. Su *et al.* [356] presented a three-stage procedure that pre-trains a CNN with attribute labels of an independent dataset, then fine-tunes the network with identity labels and finally fine-tunes the network with the learned attribute feature embedding on the combined dataset. The main difference of these approaches to ours is the way of making use of attributes to assist in the re-identification task.

8.3.3 Attribute recognition approach

8.3.3.1 Overall procedure and architecture

The architecture of the proposed attribute recognition approach is shown in Fig. 8.4. The framework consists of two branches. One branch is a CNN extracting higher-level discriminative features by several succeeding convolution and pooling operations that become specific to different body parts at a given stage (P3) in order to account for the possible displacements of pedestrians due to pose variations. Another branch extracts the viewpoint-invariant Local Maximal Occurrence (LOMO) features, a robust visual feature representation that has been specifically designed for viewpoint-invariant pedestrian attribute recognition and achieving state-of-the-art results [250]. The extracted LOMO features are then projected into a linear subspace using Principal Component Analysis (PCA) and then fused with the CNN output using

a fully-connected layer and a final (fully-connected) output layer with one neuron per attribute to recognise.

The architecture of the CNN is a succession of convolution and max pooling layers that is relatively standard in the literature, up to the pooling layer P3. At this stage (P3), we propose to divide the resulting feature maps vertically into 4 equal parts roughly corresponding to the regions of head, upper body, upper legs and lower legs. For each part, similar to [375], we use two layers (C4, C5) with 1D horizontal convolutions of size 3×1 without zero-padding reducing the feature maps to single column vectors. These 1D convolutions allow to extract high-level discriminative patterns for different horizontal stripes of the input image. In the last convolution layer, the number of channels is increased to 150, and these feature maps are given to a fully-connected layer (fc1) to generate an output vector of dimension 500. All the convolution layers in our model are followed by batch normalisation and ReLU activation functions.

We experimentally showed that this specific CNN architecture as well as the fusion of these discriminant deep features with the more "invariant" LOMO features provides a powerful model for pedestrian attribute recognition giving state-of-the-art results on public benchmarks.

8.3.3.2 Training

The proposed CNN is trained in a supervised way using a dataset of pedestrian images with annotated attributes. The weights are initialised at random and updated using Stochastic Gradient Descent minimising the global loss function (Eq. 8.2) on the given training set. Since most attributes are not mutually exclusive, *i.e.* pedestrians can have several properties at the same time, the attribute recognition is a multi-label classification problem. Thus, the multi-label version of the sigmoid cross entropy is used as the overall loss function:

$$E = -\frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}[w_l y_{il}\log(\sigma(x_{il})) + (1-y_{il})\log(1-\sigma(x_{il})]\ , \tag{8.2}$$

$$\text{with}\quad \sigma(x) = \frac{1}{1+exp(-x)}\ ,$$

where L is the number of labels (attributes), N is the number of training examples, and y_{il}, x_{il} are respectively the l^{th} label and classifier output for the i^{th} image. Usually, in the training set, the two classes for an attribute are highly unbalanced. That is, for most attributes, the positive label appears generally less frequently than the negative one. To handle this issue, we added a weight w to the loss function: $w = -log_2(p_l)$, where p_l is the positive proportion of attribute l in the dataset.

8.3.4 Attribute-assisted person re-identification

8.3.4.1 Overall approach

We proposed a new CNN-based similarity metric learning approach for pedestrian re-identification [3, 15] that effectively combines automatically learned visual features with semantic attributes, extracted by our method described in the previous section. The overall framework is shown in Fig. 8.5. The framework is composed of two neural networks that are pre-trained. The first is a deep CNN that is trained in a supervised way to classify identities on a separate training set using the softmax cross-entropy loss:

$$E_{identification} = -\sum_{k=1}^{N} y_k log(P(y_k = 1|x)) \quad \text{with} \tag{8.3}$$

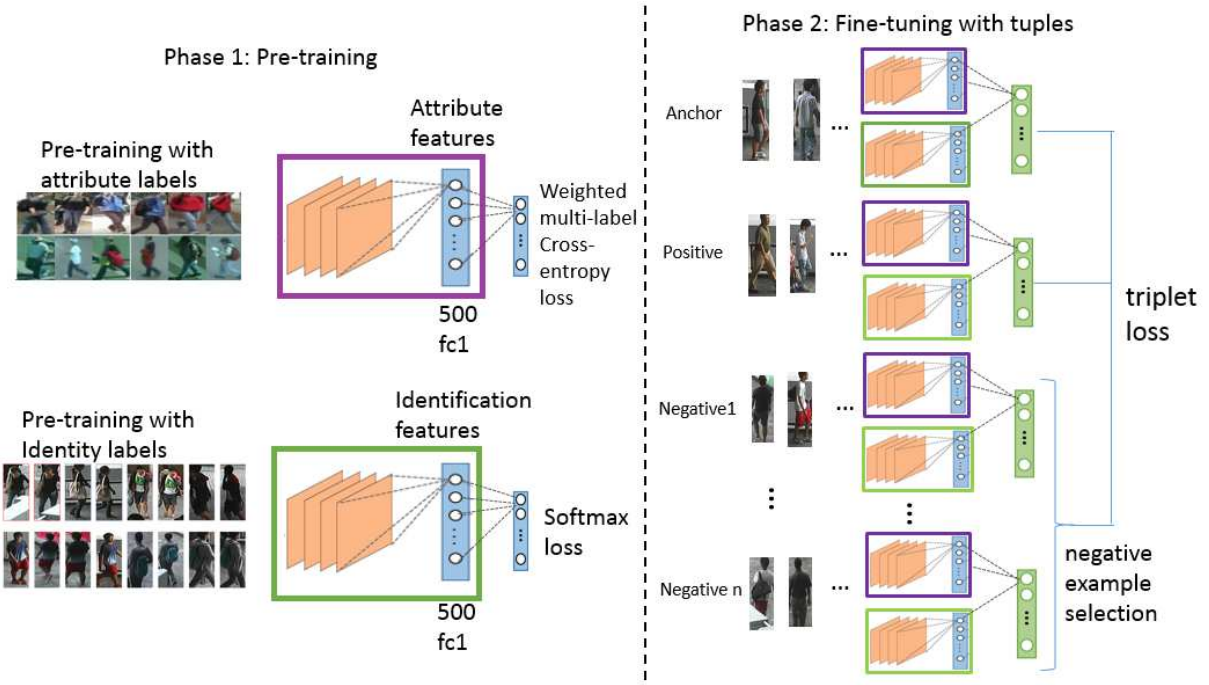


Figure 8.5: Overall architecture of our attribute-assisted re-identification approach.

$$P(y_j = 1|x) = \frac{e^{W_j^T x + b_j}}{\sum_{k=1}^{N} e^{W_k^T x + b_k}} \tag{8.4}$$

(For more details refer to [20]). Then we remove the output classification layers of the network and keep the other parts of the network which are related to feature selection. The second part is our attribute recognition network that is trained as described in the previous section (and without the LOMO features). After training, we also remove the output layers and keep all the other layers up to the first fully-connected layer (fc1). The output vectors from the hidden layers of the two CNNs represent high-level features related to attributes and pedestrian identities respectively. In order to combine the extracted features effectively, we propose to integrate both output vectors in a new neural network that automatically learns these fusion parameters on the re-identification task in a triplet architecture (described in the next section). This leads to a fully neural architecture that can be trained and fine-tuned as a whole to maximise the re-identification performance.

8.3.4.2 Fusion by triplet architecture

The pre-trained attribute CNN and identification CNN are combined (using their intermediate fully-connected layers fc1 and adding a new fully-connected layer) and trained in a triplet architecture (see Fig. 8.5). Here, we proposed to use an improved triplet loss with hard example selection to learn the optimal fusion of the two types of features. Unlike the classical triplet loss, a $(K+2)$-tuple of images instead of a triplet is projected into the feature space. The tuple includes one reference image $\mathbf{R}$, one positive image $\mathbf{P}_+$ and K negative images $\mathbf{N}_j$. With their

Accuracy Rate (%)				Recall@FPR=0.1		AUC	
MRFr2[133]	DeepMar[238]	mlcnn[451]	ours	mlcnn[451]	ours	mlcnn[451]	ours
80.8	85.4	86.6	**90.0**	65.6	**79.9**	87.0	**93.0**

Table 8.1: Attribute recognition results on PETA (in %).

respective CNN output vectors $\mathbf{O}_R$, $\mathbf{O}_P$ and $\mathbf{O}_{N_j}$, this constraint is defined as:

$$min(\|\mathbf{O}_R - \mathbf{O}_{N_j}\|_2^2) - \|\mathbf{O}_R - \mathbf{O}_P\|_2^2 > m \ , \tag{8.5}$$

with m being a margin (set to 1 in our experiments). Similarly to the distance learning approach "Top-push" proposed by [425] , hard example mining in [61, 332] or moderate positive example mining in [338], the idea is to find the most appropriate example(s) to update the model. The negative example that is closest to the reference is considered the hardest example and having the highest potential for improvement. The network is thus updated efficiently by pushing the hardest example further away from the reference. The intuition is that if the positive example is ranked in front of the hardest negative example then the positive example is ranked first, which is our goal. In classic triplet loss, a large part of triplets does not violate the triplet constraint (*c.f.* Eq. 8.5). Thus, these triplets are useless for learning. The selection among K negative examples reduces the number of unused training data and can make the training more efficient.

To further enhance the loss function, as an extension of [117], we added a term including the distance between the reference example and the positive example. The loss function (called "min-triplet" loss) of one iteration is defined as:

$$E_{min-triplet} = -\max(\|\mathbf{O}_R - \mathbf{O}_P\|_2^2 - \min_j(\|\mathbf{O}_R - \mathbf{O}_{N_j}\|_2^2) + m, 0) + \alpha\|\mathbf{O}_R - \mathbf{O}_P\|_2 \ . \tag{8.6}$$

The first part of the loss is a comparison of two distances which defines a relative relationship in the feature space. The second part corresponds to an absolute distance in feature space weighted by a factor α (set to 0.02 in our experiments). Combining these two constraints leads to a more efficient learning of the resulting manifold that better represents the semantic similarities.

Using this loss function, we trained the additional fully-connected layer for the fusion, and, at the same time, we fine-tune the other parts of the network, *i.e.* the weights are updated at a lower rate. Unlike other approaches[213, 227, 237, 276], the advantage of our method is that the attributes do not need to be annotated on the re-identification dataset. We can make use of a separate dataset with annotated attributes and transfer this information to a re-identification dataset by fine-tuning.

8.3.5 Experiments

We first evaluated the accuracy of our attribute recognition approach on three different datasets: PETA (19 000 images, 65 attributes), APiS (3661 images, 11 attributes) and VIPeR (632 images, 21 attributes) (see Fig. 8.6) using the evaluation protocols proposed in the literature. The evaluation measures are the accuracy, the average recall at a False Positive Rate (FPR) of 0.1 and the Area Under Curve (AUC) of the average Receiver Operating Characteristic (ROC) curve. The results in Tables 8.1-8.3 show that our approach outperforms all state-of-the-art methods: MRFr2 [133], DeepMar [238], mlcnn [451], fusion [448], interact [449] and svm[227].

Figure 8.6: Some example images from pedestrian attribute datasets.

Accuracy	Recall@FPR=0.1			AUC			
ours	fusion[448]	interact[449]	ours	fusion[448]	interact[449]	DeepMar[238]	ours
89.3	62.1	64.7	**72.7**	86.7	87.2	**90.0**	89.5

Table 8.2: Attribute recognition results on APiS (in %).

Then we evaluated our attribute-assisted person re-identification framework on the public CUHK03 dataset [247] containing 13 164 images of 1 360 pedestrians (automatically cropped by a person detection algorithm). We pre-trained the attribute CNN branch on the Peta dataset, and the identity CNN branch on CUHK03 (the identities of the test set are different from the training set). Our evaluation criterion for re-identification is the proportion of test (query) images for which rank=n, *i.e.* in the ranking according to the Euclidean distance in the learnt projection space, a correct match is within the first n images. Table 8.4 shows the results for rank=1,5,10 compared to the state of the art on the CUHK03 ("detected") dataset. Our approach outperformed all other methods showing the effectiveness of our SNN similarity metric learning framework with improved triplet loss. One can also see the significant contribution of the attributes to the overall performance.

8.3.6 Conclusion

In this section, I presented our recent work on person re-identification showing that the use of additional semantic information (the attributes) can considerably improve the performance of similarity metric learning. Our first contribution was a powerful framework for attribute recognition fusing discriminant deep CNN features trained on pedestrian images with highly invariant LOMO features and achieving state-of-the-art classification accuracy on three public datasets. Then, by using both identity and attribute information and also an improved triplet objective function with hard negative examples and finally by employing effective pre-training and fusion strategies on appropriate CNN architectures, we were able to obtain state-of-the-art

Accuracy			Recall@FPR=0.2			AUC
svm[227]	mlcnn[450]	ours	svm[227]	mlcnn[450]	ours	ours
68.9	74.1	**83.9**	56.1	65.5	**69.6**	80.9

Table 8.3: Attribute recognition results on VIPeR (in %).

Method	rank=1	rank =5	rank =10
FPNN [247]	19.9	49.3	64.7
Convnet [61]	45.0	75.3	83.4
LOMO+XQDA [250]	46.3	78.9	88.6
SS-SVM [436]	51.2	80.8	89.6
SI-CI [386]	52.2	84.3	92.3
DNS[434]	57.3	80.1	88.3
S-lSTM [375]	57.3	80.1	88.3
S-CNN SQ [374]	61.8	80.9	88.3
CAN[255]	63.1	82.9	88.2
ours Identity only	59.7	86.1	93.3
ours fusion Id&Attr	**65.0**	**90.3**	**95.1**

Table 8.4: Re-identification result on CUHK03 ("detected").

results on challenging public benchmarks.

8.4 A gated SNN approach

8.4.1 Introduction

In the previous section, we saw one way of introducing prior knowledge into the SNN model, *i.e.* by pre-training a neural network on labelled data and fusing and fine-tuning the model into a larger neural network that learns the final similarity metric. In this section, I will describe another original approach that we proposed [17], which also uses a supervised pre-trained model on external semantic information but is based on a different neural architecture. In the preceding architecture, two neural networks are pre-trained in a supervised manner on two different labelled datasets (one for identification, one for attributes). In the model that I will present here, there are two neural networks (*i.e.* two branches) with some *common layers* in the beginning, and the training is done simultaneously with a loss function integrating the two different objectives; a so-called multi-task learning. Moreover, one branch acts as a gate to "steer" which part of the other branch should be activated.

In our re-identification application, one branch is a CNN pre-trained for person identification, and the other branch, the gating branch, is pre-trained for person body orientation estimation. The whole combined neural network model is then fine-tuned for person re-identification. In the following, we will first described the related work and then explain our approach, called Orientation-Specific CNN (OSCNN), in more detail.

8.4.2 State of the art

Most existing methods for person re-identification focus on developing a robust representation to handle the variations of view. Some methods take into account the view as extra information. For example, Ma *et al.* [266] divide the data according to the additional camera position information and learn a specific distance metric for each camera pair. Lisanti *et al.* [252] proposed to apply Kernel Canonical Correlation Analysis which finds a common subspace between the feature space from disjoint cameras. Yi *et al.* [420] proposed to apply a Siamese CNN to person re-identification. Similar to [252], the weights of two subnetworks are not shared to learn a camera view projection to a common feature space. In these approaches, camera information is used but the body orientation which is only partly due to different camera views is not modelled. That is, in the same camera view, pedestrians can exhibit different orientations and thus largely different appearances in the resulting images.

In order to solve this issue, Bak *et al.* [78] perform an orientation-driven feature weighting and the body orientation is calculated according to the walking trajectory. some other approaches [330, 378] deal with the orientation variations of pedestrian images by using Mixture of Experts. The expert neural networks map the input to the output, while a gating network produces a probability distribution over all experts' final predictions. Verma *et al.* [378] applied an orientation-based mixture of experts to the pedestrian detection problem. Sarfraz *et al.* [330] proposed to learn the orientation sensitive units in a deep neural network to perform attribute recognition. Garcia *et al.* [157] used orientations estimated by a Kalman filter and then trained two SVM classifiers for pedestrian images matching with respectively similar orientations and dissimilar orientations. And the approach of Li *et al.* [245] learns a mixture of experts, where samples were softly distributed into different experts via a gating function according to the viewpoint similarity.

Sharing the idea of mixture of experts, we proposed to build a multi-domain representation in different orientations with deep CNNs. Intuitively, an orientation-specific model should have a better generalisation ability than a camera view-specific model, since we cannot incorporate all possible surveillance camera views. Further, instead of using discrete orientations for the gating activation function, in our method, we use a *regression model* to estimate an accurate and continuous body orientation. This allows to continuously weight different expert models for re-identification and also avoids combining contradictory orientations at the same time.

8.4.3 Body orientation-assisted person re-identification

8.4.3.1 Model architecture

The overall procedure of our re-identification approach OSCNN is shown in Fig. 8.7. The network contains an orientation gating branch and a re-identification branch consisting of 4 feature embeddings regarding the 4 main orientations: left, right, frontal and back. The final output feature representation is a linear combination of the four expert outputs and is steered by an orientation gate unit which is a function of the estimated orientation.

The proposed neural network architecture consists of two convolution layers shared between the two branches. In the re-identification branch, there are 3 further convolution layers followed by 4 separate, parallel fully-connected layers (left,right,front,back) of 512 dimensions, each one corresponding to a local expert. Thus, our network learns different projections from different orientation domains to a common feature space.

In the orientation regression branch, 2 convolution layers and 2 fully connected layers are connected to the common convolutional layers. The estimated orientation output by the ori-

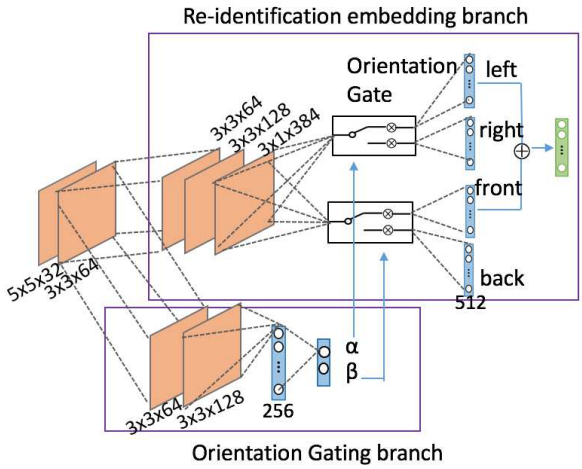


Figure 8.7: Overview of the OSCNN architecture.

entation gating branch is represented by a two-dimensional Cartesian vector $[\alpha, \beta]$ constructed by projecting the orientation angle on the left-right axis (x) and on the front-back (y) axis and then normalising it to a unit vector. Based on this vector, the orientation gate selects and weights either the left or the right component and either the front or the back component of the re-identification branch. Let $f_{\{left,right,front,back\}}$ be the output feature vectors of the 4 different orientation branches. The final re-identification output vector is the sum of the left-right component and the front-back component:

$$f_{ouput} = max(\alpha, 0)f_{left} + max(-\alpha, 0)f_{right} + max(\beta, 0)f_{front} + max(-\beta, 0)f_{back} \tag{8.7}$$

Different from the classic mixture of experts approach, our orientation gate is set before the local experts, and we perform a regression instead of a classification. The advantage of our orientation gate is that it avoids combining contradictory orientations like front and back. Computationally, only two among four orientations are used and combined according to the sign of α and β. This further allows saving computation.

8.4.3.2 Training

There are two stages to train the model, explained in the following.

Multi-task network pre-training

In the first stage, the orientation regressor and a general re-identification feature embedding are both trained in parallel with two separate objective functions.

We start training the network with pedestrian identity labels and orientation labels respectively. The identification branch is trained as in our attribute-assisted model in section 8.3.4, *i.e.* using the cross-entropy loss $E_{identification}$ (Eq. 8.3) on the softmax output and identity-labelled person images.

For the body orientation, we use the Euclidean loss to train the orientation regression of α and β. For a given training example, we have:

$$E_{orientation} = \frac{(\alpha - \widehat{\alpha})^2 + (\beta - \widehat{\beta})^2}{2} \tag{8.8}$$

Methods	R1	R5	R10	R20
Baseline (CUHK01)	76.6	93.8	97.0	98.8
OSCNN (CUHK01)	78.2	94.1	97.3	99.1
OSCNN (CUHK01+03)	**83.5**	96.4	**99.0**	**99.5**
LOMO+XQDA [250]	63.2	83.9	90.1	94.2
ImporvedDL [61]	65.0	88.7	93.1	97.2
Deep Embedding [338]	69.4	-	-	-
Norm X-Corr [357]	81.2	-	97.3	98.6
Multi-task [114]	78.5	**96.5**	97.5	-

Table 8.5: Experimental evaluation of OSCNN on the CUHK01 dataset.

where $\widehat{\alpha}, \widehat{\beta}$ are predicted orientation labels of the example. Orientation has been annotated with 8 discrete labels.

For datasets that have both identity and orientation labels, we train the network with a combined loss $E_{multi-task} = E_{identification} + \lambda E_{orientation}$. Then, orientation and identification are learned jointly. Otherwise, the two branches are trained separately.

Orientation-specific fine-tuning with triplets

In the second training stage, we fine-tune the network parameters using similarity metric learning in order to specialise the 4 different local experts. For the re-identification branch, we remove the last fully-connected layer and duplicate four times the first fully-connected layer. Different choices and weightings are performed according to the orientation of the person in the input image estimated by the orientation branch. Thus, the four orientation-specific layers are updated in different ways, whereas the other layers keep their pre-trained weights. For the similarity metric learning, we use the improved triplet loss with hard examples (*c.f.* Eq. 8.6) as for our attribute-based SNN.

8.4.4 Experiments

For the evaluation of our approach, we used the datasets Market-1501 [440] (32 668 images of 1 501 different persons), Market-1203 [267] (a subset of Market-1501 with annotated body orientations) and CUHK01 [246] (3884 images, 971 persons). The evaluation measures are the rank 1 accuracy (R1), the mean average precision (mAP) and the Cumulative Match Curve (CMC) (*c.f.* section 8.2.4).

We compared our OSCNN to the state-of-the-art approaches on Market-1501 and CUHK01. Following the test protocol in [61, 114], we added also the CUHK03 images to the training for the test on the CUHK01 and we compared to the methods only using these two datasets for training. As Table 8.5 shows, our method is superior to most results of the state of the art. Even without much extra CUHK03 training data, our method shows a competitive performance. The *baseline* model is not using body orientation and shows inferior performance, clearly demonstrating the benefit of the orientation regression and local expert training via the gated SNN architecture.

On the Market-1501 dataset, our OSCNN outperforms most state-of-the-art methods. The advantage of our approach is that the model does not need a pre-training step with a much larger pre-training dataset composed of ImageNet as [239, 361, 445, 447]. And our model has a much lower complexity (1.15×10^8 FLOPs of our model compared to 1.45×10^9 FLOPs of JLML and to 3.8×10^9 FLOPs of SVDNet). Recently some state-of-the-art approaches

Methods	R1	mAP
Baseline	77.3	53.9
OSCNN	78.9	55.2
OSCNN+re-rank	83.9	**73.5**
LOMO+XQDA [250]	43.8	22.2
Gated SCNN [117]	65.9	39.6
Divide fues re-rank [427]	82.3	72.4
LSRO [445]	78.1	56.2
DeepContext [239]	80.3	57.5
K-reciprocal re-rank [447]	77.1	63.6
SVDnet [361]	82.3	62.1
JLML [248]	**85.1**	65.5

Table 8.6: Experimental evaluation of OSCNN on the Market-1501 dataset.

employ re-ranking [427, 447] which uses information from nearest neighbours in the gallery and significantly improves the performance. As Table 8.6 shows, our approach can largely benefit from this technique and achieves a state-of-the-art result on Market-1501.

8.4.5 Conclusion

In this section, we presented a novel approach to tightly integrate labelled information (body orientation and person identity) via supervised learning into a multi-task CNN framework, and a method to dynamically activate certain parts of the structure via a learnt gating mechanism. By fine-tuning this pre-trained model in a weakly supervised similarity metric learning approach, *i.e.* a triplet SNN, we were able to achieve excellent results on the task of person re-identification on several challenging public benchmarks.

In the next section, I will present a final approach for including additional semantic information in the similarity learning for person re-identification: the appearance of the group of persons surrounding the one to re-identify.

8.5 Using group context

8.5.1 Introduction

The two previous contributions consisted in integrating more semantic external information (prior knowledge) into the similarity metric learning: pedestrian attributes and body orientation, and this significantly improved the re-identification performance. Thus, we can assume that the learnt metric better reflects real semantic similarities. However, in a realistic setting, there are still frequent cases, where the images of two different persons are extremely similar and where even humans have difficulties to tell if they belong to the same person or not. Attributes and body orientation might not help always, as in public spaces there are often common paths that people take; thus there are only a few different walking directions and thus body orientations. Also attributes may not discriminate sufficiently, as people tend to wear similar clothes (*e.g.* a suit) and may have similar hair styles (*e.g.* short brown hair).

To address this problem, our idea was to use context information about the surrounding group of persons [16] (see Fig. 8.8). In realistic settings, people often walk in groups rather than

Figure 8.8: (a) Single person images. (b) Corresponding group images of (a). Even for a human, it may be difficult to tell if the three top images belong to the same person or not. Using the context of the surrounding group, it is easier to see that the middle and right images belong to the same person and the left image belongs to another person.

alone. Thus, the appearance of these groups can serve as visual context and help to determine whether two images of persons with similar clothing belong to the same individual. However, matching the surrounding people in a group in different views is also challenging. On the one hand, it undergoes the same variations as for a single person's appearance. On the other hand, the number of persons and their relative position within the group can vary over time and across cameras. Further, partial occlusions among individuals are very likely in groups.

Our approach [16] was to train a deep CNN for single-person re-identification and use it to additionally extract group features via a specific pooling operation. Finally, we defined a combined similarity measure using group and individual representations.

8.5.2 State of the art

In the literature, there are several group association (or group re-identification) approaches. Zheng *et al.* [442] extracted visual words which are the clusters of SIFT+RGB features in a group image. Then they built two descriptors that describe the ratio information of visual words between local regions to represent group information. Cai *et al.* [103] used covariance descriptors to encode group context information. And Lisanti *et al.* [253] proposed to learn a dictionary of sparse atoms using patches extracted from single person images. Then the learned dictionary is exploited to obtain a sparsity-driven residual group representation. These approaches can be severely affected by background clutter, and thus a preprocessing stage is necessary, *e.g.* a background subtraction.

Some other approaches use trajectory features to describe group information. Wei *et al.* [395], for example, presented a group extraction approach by clustering the persons' trajectories observed in a camera view. They introduced person-group features composed of two parts: SADALF features [144], extracted after background subtraction and representing the visual appearance of the accompanying persons of a given individual, and a signature encoding the position of the subject within the group. Similarly, Ukita *et al.* [372] determined for each pair of

pedestrians whether they form a group or not, using spatio-temporal features of their trajectories like relative position, speed and direction. Then, the group features composed of the trajectory features (position, speed, direction) of individuals in each group, the number of persons as well as the mean colour histograms of the individual person images. However, when people walk in a group, the position and speed are not always uniform. Thus, the trajectory-based features may not be precise and change significantly over time.

Unlike these methods, the advantage of our approach is that there is no need for a pre-processing stage of person detection or background subtraction. Our model is pre-trained on single-person re-identification data to learn the discriminative features that distinguish identities in images. Using a global max-pooling operation, the proposed model is, by design, invariant to displacements of individuals within a group. Moreover, the deep neural network that we employed can provide a richer feature representation to describe groups than the colour and texture features used by existing methods.

8.5.3 Proposed group context approach

In the first step, we trained a CNN for classification of persons on a given training dataset using a supervised approach by minimising the softmax cross-entropy loss as in sections 8.3.4 and 8.4.3 (*c.f.* Eq. 8.3). We tested a deep CNN architecture of 9 alternating convolution and max-pooling layers followed by 2 fully-connected layers, as well as a pre-trained Resnet-50 [178] that is fine-tuned. After training the model, in order to compute a distance between a query image and a gallery image, a Siamese architecture is build, where each branch (query and gallery) contains two parts: one for a single-person image and the other for an image with the surrounding context (showing a group of persons). Figure 8.9 illustrates this. The region corresponding to the queried individual in the group image is covered (with the mean colour) in order to remove the redundancy.

For the single-person branch, we discarded the last fully-connected layer and used the intermediate fully-connected layer as an embedding to compute a single-person distance. Note that here we did not perform an additional similarity metric fine-tuning as in the previous approaches. This might further improve the results, but would have been considerably longer to train and was not necessary to demonstrate the contribution of group context. For the group branch, the last two fully-connected layers are discarded and larger group images are given as input. A Global Max-Pooling (GMP) operation on all spatial locations of the resulting feature map from the last convolution layer is applied leading to a k-dimensional vector, with K being the number of feature maps. As illustrated in Fig. 8.9, for a given set of query and candidate person and group images, P_i, G_i and P_j, G_j respectively, we obtain 4 feature vectors: $F(C(P_i)), GMP(C(G_i)), F(C(P_j))$, and $GMP(C(G_j))$, where $C(I)$ is the output of the last convolution layer for image I, $F(\cdot)$ represents the operation of the first fully-connected layer, and $GMP(\cdot)$ the Global Max-Pooling operation. Then, the single-person and group distances between two images are defined as:

$$D_{id}(P_i, P_j) = 1 - cos(F(C(P_i)), F(C(P_j))) \text{ and} \tag{8.9}$$

$$D_{gr}(G_i, G_j) = 1 - cos(GMP(C(G_i)), GMP(C(G_j))) \, . \tag{8.10}$$

The final distance measure is simply the sum of these two distances:

$$D(I_i, I_j) = D_{id}(P_i, P_j) + D_{gr}(G_i, G_j) \, , \tag{8.11}$$

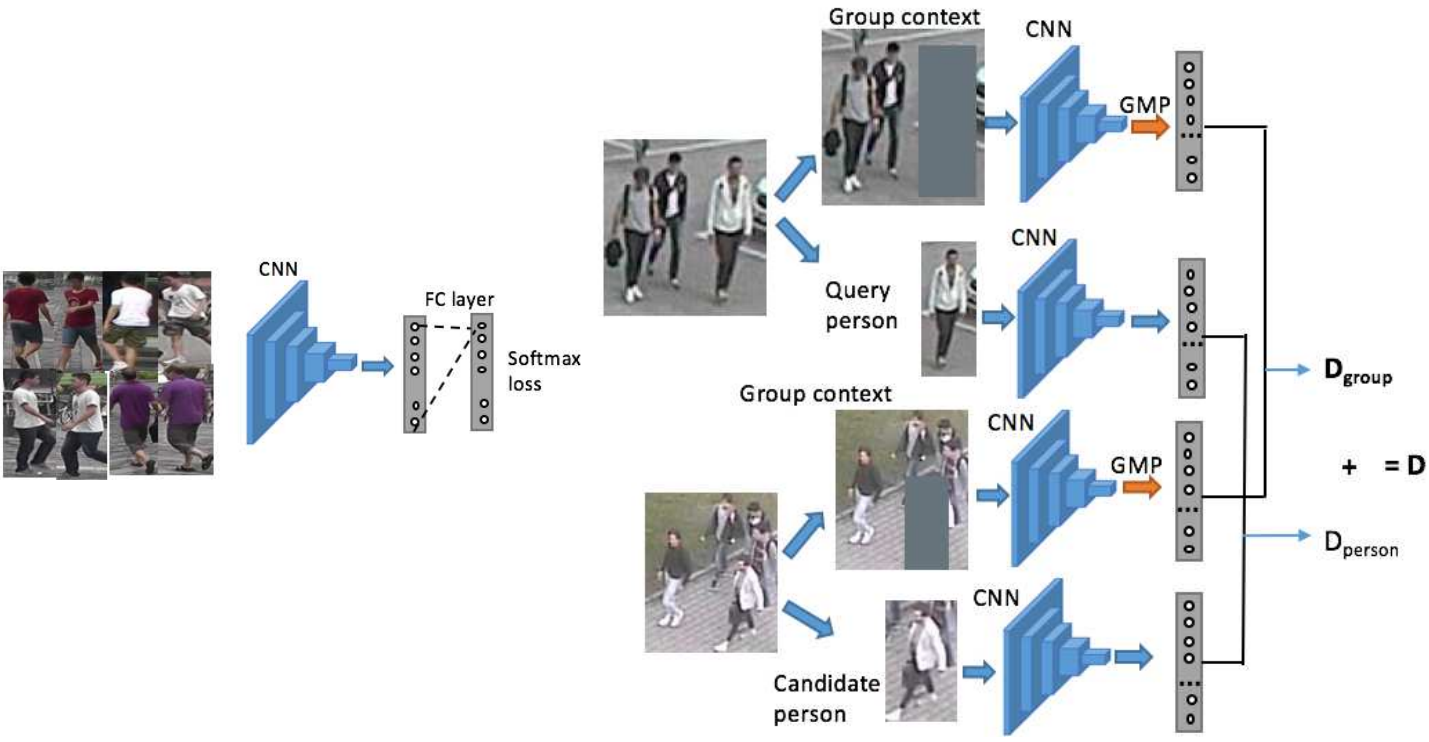


Figure 8.9: Overview of our group-assisted approach. *Left:* A CNN is first trained with single-person images. *Right:* For re-identification, the fully-connected layer(s) are removed and the same CNN model is used for query and candidate images to compute pairwise distances used for the final ranking.

which can be used for ranking and for re-identification based on the first ranked result with respect to a given query image.

8.5.4 Experiments

For evaluation our approach, we used the OGRE dataset [253], which contains 1 279 images of 39 groups acquired by three disjoint cameras pointing at a parking lot. This is a challenging dataset with many different viewpoints and self-occlusions. We manually annotated a subset of this dataset with 450 bounding boxes and 75 identities.

To show that our approach can be applied with different CNN architectures, we used Resnet-50, pre-trained on ImageNet, and also our own CNN architecture (similar to the ones of the previous sections and denoted as Convnet-5 here) composed of 9 alternating convolution and max-pooling layers and 2 fully-connected layers.

The CMC scores (*c.f.* section 8.2.4) for person re-identification are shown in Table 8.7. We compared the person re-identification results with some variants of our method. "*Sum feature*" and "*Concatenate feature*" represent variants that first sum or concatenate the single-person feature representation and the group feature representation and then compute the distance measure on these vectors. We tested also a variant that retains the query or candidate person image in the group image without covering the corresponding region with the mean colour.

The results in Table 8.7 show that the proposed method (*i.e.* covering the person in the group image and summing the person and group distance) achieved the best re-identification

Variant	Resnet-50			Convnet-5		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
Single person only	47.2	69.3	78.8	51.6	70.4	77.9
Group context only	26.2	57.2	66.3	12.7	42.3	53.0
Sum features	41.1	69.9	77.7	16.3	50.3	61.7
Concatenate features	51.9	**75.1**	81.1	16.4	52.6	61.9
Dist sum w/o img cover	54.1	73.7	80.8	52.2	**70.6**	78.1
Dist sum w/ img cover	**56.8**	73.7	**81.7**	**53.7**	70.4	**78.6**

Table 8.7: Person re-identification accuracy (CMC scores in %) on the OGRE dataset.

results with both tested CNN models. Overall, our proposed method based on Resnet-50 and Convnet-5 increased the result by 9.6% points and 2.1% points respectively compared to the approach only using single-person images. These results clearly demonstrate that group context has the ability to considerably reduce the appearance ambiguity and improve the person re-identification performance. An advantage of our method is that it can be easily applied to any CNN-based single person re-identification model without any further training.

8.5.5 Conclusion

In this section, we presented our last approach for integrating prior semantic information into the similarity metric. We used the context in the image to gather additional descriptions on the surrounding objects, *i.e.* accompanying persons in this case. The proposed method has the advantage that no additional training is required to create a model for the context. However, it assumes that the context is composed of the same category of objects (*i.e.* persons). It would be interesting to see how a model specifically trained for other types of "accompanying" objects, *e.g.* suitcases, trolleys, animals, could improve on the re-identification performance, assuming that this context does not change significantly over time and across different cameras.

8.6 Listwise similarity metric learning and ranking

8.6.1 Introduction

In this final section, we come back to a more fundamental study on the underlying similarity metric learning algorithm for SNNs applied to person re-identification [18]. In particular, we conceived an new objective function and metric learning approach that indirectly optimises the final re-identification evaluation measures, based on the ranking of examples with respect to a query.

In contrast to our previous contributions on tuple-based and polar sine-based functions, where dissimilar pairs a simultaneously "pushed away" in the projection space, here, we explicitly consider the *ranking*, *i.e.* the order, of a set of candidate samples (positive and negative) w.r.t. to a given query. This is similar to our min-triplet loss (Eq. 8.6) in section 8.3, but incorporates more information about mis-ranked examples and their relation to correct matches.

8.6.2 State of the art

We already discussed the state of the art in similarity metric learning and SNN in section 7.2, and we will concentrate on approaches for ranking. In particular, we are interested here in one category of methods which is called *learning-to-rank*. This approach, widely applied in information retrieval and natural language processing, consists in learning a model that can compute the optimal ordering of a list of items. Many learning-to-rank methods have been proposed in the literature, like the pairwise approaches RankSVM [183], RankNet [100] and listwise approaches ListNet [107], ListMLE [406] and LambdaRank [101], which take an entire ranked list of objects as the learning instance.

Person re-identification could be considered as a retrieval problem based on ranking, where the matching images, *i.e.* the ones showing the same person as in the query image, should be ranked before all the others. Some methods proposed in the literature follow this approach. For example, in the work of Prosser *et al.* [313] a set of weak RankSVMs is learnt, each computed on a small set of data, and then combined to build a stronger ranker using ensemble learning. Wang *et al.* [389] applied the ListMLE method to person re-identification: their approach maps a list of similarity scores to a probability distribution, then utilizes the negative log likelihood of ground truth permutations as the loss function.

8.6.3 Learning to rank with SNNs

The principal problem with rank learning with statistical machine learning approaches, particularly neural networks, is that the rank is discrete, and, thus, rank errors are not differentiable. In order to apply an iterative gradient descent optimisation, the RankNet approach from Burges *et al.* [100] uses a sigmoid function (which is convex and derivable) and pairwise relationships to model the ranking. Later, they proposed the LambdaRank method [101], where the gradient is scaled by the improvement of a global ranking measure (the Normalized Discounted Cumulative Gain (NDCG)) incurred by swapping two examples in the ranking.

Our proposed approach is based on this approach. However, we do not use the pairwise cross-entropy loss function but the triplet loss (*c.f.* Eq. 7.7 in section 7.2.3). Also, we combine two ranking measures, the rank-1 (R1) score and the Average Precision (AP), *i.e.* the area under the precision-recall curve, for a given query and ranking. Finally, we propose to minimise our modified triplet loss function, called *Rank-Triplet*, on batches of examples (*i.e.* lists), where, in each batch, only the most relevant triplets are selected for updating the model parameters, *i.e.* we use only mis-ranked examples.

More precisely, a training batch is formed by M images of N identities. We take one example in the batch as query and perform a ranking among the rest of images in the batch. (For the sake of a robust metric, we add a margin m to the distance of ranking positions between the true correspondences and the probe before ranking.) The AP and R1 scores are computed for each of those query rankings. Then, with respect to one given query, we form all possible mis-ranked pairs (false correspondences ranked before the true correspondence), and we re-calculate the new AP and R1 scores by swapping positions of the pair in the ranking and thus obtain the gains ΔAP and $\Delta R1$, respectively. The loss of each triplet is weighted by the sum of these gains. The

Loss function	R1	mAP
Classification loss	74.3	51.0
Siamese loss	62.9	46.6
Triplet loss	74.3	56.5
Quadruplet loss	74.9	58.1
Hardbatch	81.0	63.9
Baseline	82.1	66.5
Ours (Rank-triplet)	**83.6**	**67.3**

Table 8.8: Re-identification results (in %) on Market-1501 with different loss functions.

final Rank-triplet loss is calculated as follows:

$$E_{rank-triplet} = \frac{1}{MN}\sum_{i=1}^{MN}\frac{1}{K_i}\sum_{j\in TC_i}\sum_{\substack{k\in FC_i \\ r_k^i<r_j^i}}[\|f(x_i)-f(x_j)\|_2^2 - \|f(x_i)-f(x_k)\|_2^2+m]\cdot(\Delta AP_{jk}^i + \Delta R1_{jk}^i), \quad (8.12)$$

where x_i is the i^{th} training example in a training batch, K_i is the number of mis-ranked pairs w.r.t. the i^{th} example as query, and r_j^i is the rank of the j^{th} example w.r.t. the i^{th} image as query. TC_i/FC_i is the true/false correspondence set of the i^{th} example. ΔAP_{jk}^i is the gain of AP by swapping the j^{th} and k^{th} examples w.r.t. the i^{th} example as query and analogously for R1.

With our evaluation based weighting, we make a trade-off between the moderate hard examples and hardest examples, *i.e.* more weight is given to the hardest examples to make the learning efficient, and, at the same time, the less hard examples are used to stabilise the training.

8.6.4 Experiments

I will present here a summary of the results of our experiments on the person re-identification problem with a Resnet-50 neural network architecture pre-trained on Imagenet and with different loss functions. More detailed experiments with different neural architectures and on an image retrieval task can be found in [116].

First, we evaluated the re-identification performance of our Rank-Triplet approach on the Market-1501 dataset and compared it with several other common loss functions. The results are shown in Table 8.8. For the supervised classification with identity labels, the softmax cross entropy loss is used. The margin in the Siamese loss and triplet loss is fixed to the default value $m = 1$. For the pairwise Siamese learning the contrastive loss is used (*c.f.* Eq. 7.6), and we generated all possible pairs of images within a batch. The triplet loss is calculated according to Eq. 7.7. And the hard batch triplet loss takes only the hardest positive image and negative image, the hard batch triplet loss is calculated as follows.

$$L_{hard-batch} = \frac{1}{N}\sum_{i=1}^{N} max(\max_{j\in TC_i}\|f(x_i)-f(x_j)\|_2^2 - \min_{k\in FC_i}\|f(x_i)-f(x_k)\|_2^2+m, 0)\ , \quad (8.13)$$

where N is the number of triplets, TC_i/FC_i is the true/false correspondence set of the i^{th} example.

Methods	Market-1501		DukeMTMC-Reid		CUHK03-NP	
	R1	mAP	R1	mAP	R1	mAP
Hardbatch triplet loss [184]	81.0	63.9	62.8	42.7	46.4	50.6
Our baseline	82.1	66.5	72.4	52.0	45.3	48.9
Our Rank-Triplet loss	83.6	67.3	74.3	55,6	47.8	52.4
Rank-Triplet+re-rank [447]	86.2	**79.8**	78.6	**71.4**	**60.4**	**60.8**
LOMO+XQDA [250]	43.8	22.2	30.8	17.0	12.8	11.5
LSRO [445]	78.1	56.2	67.7	47.1	-	-
Divide and fuse [427]	82.3	72.4	-	-	30.0	26.4
K-reciprocal re-rank [447]	77.1	63.6	-	-	34.7	37.4
ACRN[333]	83.6	62.6	72.6	52.0	-	-
SVDNet [361]	82.3	62.1	76.7	56.8	41.5	37.3
JLML [248]	85.1	65.5	-	-	-	-
DPFL [115]	**88.6**	72.6	**79.2**	60.6	40.7	37.0

Table 8.9: Comparison of our Rank-Triplet approach with state-of-the-art methods for person re-identification.

The quadruplet loss in [113], based on triplets, pushes away also negative pairs from positive pairs w.r.t. different probe images. The loss is formulated as:

$$E_{quadruplet} = -\frac{1}{N}\sum_{i=1}^{N}[max(\|f(x_i) - f(x_j)\|_2^2 - \|f(x_i) - f(x_k)\|_2^2 + m_1, 0) + max(\|f(x_i) - f(x_j)\|_2^2 - \|f(x_k) - f(x_l)\|_2^2 + m_2, 0)], \quad (8.14)$$

where x_j is the feature embeddings of an image from the same identity as x_i and x_k,x_l are from different identities. As [113], we set the $m_1 = 1, m_2 = 0.5$.

Finally, we implemented a baseline which is based on the Rank-triplet loss function without the term for evaluation gain weighting. But the triplet selection is still based on the batch ranking orders.

Our Rank-triplet achieved the best performance among these loss functions. This comparison shows the effectiveness of the listwise evaluation measure-based weighting. Rank-triplet gives also better results than hardbatch. This shows that using moderate difficult examples and weighting them helps the metric learning. Using the quadruplet loss also slightly improves the performance with respect to triplets. This could eventually be combined with our loss.

We also compared our method with state-of-the-art methods on three benchmark datasets: Market-1501, DukeMTMC-Reid and CUHK03-NP. The results are shown in Table 8.9. Our method Rank-triplet achieves better results than most of the other methods. Only on Market-1501, DPFL obtains a slightly better result, and SVDNet and DPFL on DukeMTMCReid. On the CUHK03 benchmark, our methods achieves the best results. DPFL and SVDNet are based on a classification loss, and the CUHK03 dataset contains probably too few images per person to train a good classifier. However, the triplet loss is not much affected because a large number of triplets can still be formed. Since the main contribution of these two state-or-the-art methods focuses on the network architecture, these methods can potentially be combined with our loss function.

8.6.5 Conclusion

In this section, we presented a final contribution on improving similarity metric learning with SNN-based models. This approach not only uses lists (or tuples) to improve the convergence and final similarity metric, and thus the overall re-identification performance, but also directly incorporates in the optimisation the *order of the ranking* of lists of examples (operating iteratively on random batches). Our experimental results showed that this is a very powerful method for similarity metric learning on image-based recognition problems, like person re-identification and image retrieval with challenging and realistic state-of-the-art datasets.

8.7 Conclusion

In summary, we presented four different approaches to improve similarity metric learning with SNNs illustrated on the (single-shot) person re-identification problem. We showed how semantic prior knowledge can be effectively incorporated in these neural network-based architectures, for example semantic pedestrian attributes and body orientation. Further, we presented an elegant approach to make use of scene context, *i.e.* surrounding persons, for re-identifying a given person, which can be applied to any CNN-based person re-identification model without any further retraining and fine-tuning. Finally, we introduced an original objective function for a new learning-to-rank approach with SNNs giving state-of-the-art results.

Most of these contributions could be combined to probably further improve the results on person re-identification. Also, applying some of these methods to the face and gesture verification and classification problems of the previous chapter would be very interesting and give insights into the general performance of the different approaches. Despite the large progress that has been recently made on the person re-identification problem, there still remain some open issues that make their application to real-world settings difficult. For example, the overall generalisation capacity of deep neural network models has been little studied in this context, because evaluation is usually conducted on a given test dataset that is similar to the training dataset in terms of acquisition conditions. In real-world applications, methods that are performing well on any type of realistic data are required, *i.e.* different viewing angles, resolution, lighting conditions, possibly by new mechanisms that automatically adapt to new previously unseen environments (domain adaptation, unsupervised and continuous learning *etc.*). Further, the automatic optimal fusion of different cues would allow for more robust systems, *e.g.* the context of carried objects and accompanying persons, the person's gait, walking speed, characteristic motion as well as scene information from several cameras, for example.

We will elaborate more on future research directions in the following section.

9 Conclusion and perspectives

In the preceding chapters, I described a large part of our research work that I co-supervised and that I was involved in during the last 10 years. I had the chance the work in several European countries with extremely competent persons and on a variety of related topics, which, on the one hand, gave me a solid technical and scientific background, and on the other hand, broadened and enriched my culture, vision and thinking in various aspects.

Of course, this is not an exhaustive report of all the work that I have been involved in. Only the most representative results are presented, and I have not mentioned some of the more recent studies. For example, our work on weakly supervised and unsupervised learning techniques for object recognition and tracking, performed in a lab-internal project across several teams and with two Master students. And also, our work on neural network compression and approximation conducted within the context of an international collaboration with a colleague from the Federal Pernambuco University, Recife, Brazil. I will briefly describe these in the perspectives, as both of the studies are being pursued, or will be, with PhD theses starting in 2018 and 2019.

But before outlining future research directions and projects, I will first summarise the research work that I presented in this manuscript and draw some general conclusions.

9.1 Summary of research work and general conclusion

Although there is some overlap, I categorised the description of our past research into two parts, mainly due to the context of the two different research institutes and environments I have been working in: Idiap and LIRIS. The first part described our machine learning and computer vision approaches for visual object tracking in challenging dynamic environments, and the second outlined our contributions on weakly-supervised metric learning methods using Siamese Neural Networks.

From a computer vision point of view, we first focused on on-line multiple object tracking in videos from a single RGB camera, and in particular, multiple face tracking applications in dynamic contexts, and we addressed a variety of common problems, such as data association, the tracking of a variable number of objects, an efficient inference for real-time applications, the integration and adaptation of robust appearance models.

In particular, in chapter 5, we proposed an effective on-line algorithm for multi-face tracking that is able to cope with *missing face detections* over longer periods of time without losing the object, as well as *frequent false detections* without falsely initialising new tracks. Then, we extended this work by an additional algorithm that estimates the Visual Focus of Attention of a tracked person. The proposed unsupervised on-line learning approach learns a robust model from very few examples of face images acquired during the tracking.

In chapter 6, we limited ourselves on tracking a single object in a video and focused on creating more robust appearance models and on-line learning approaches. This included three

major contributions: one method particularly suited to tracking deformable objects in challenging videos without any prior knowledge, another method including motion context in the on-line learning of a robust discriminative appearance model and the last contribution resulting from the PhD work with Salma Moujtahid [**?**] proposing a method that effectively and dynamically incorporates the scene context in the tracking process by selecting tracking algorithms specialised for a given environment and context.

In the second part, in chapter 7, I presented our work with Lilei Zheng [441] and Samuel Berlemont [90] on Siamese Neural Networks for similarity metric learning, applied to pairwise face verification as well as gesture and action classification. Through more balanced and better conditioned objective functions and learning algorithms, our proposed approaches were able to address several problems, such as the better rejection of unknown classes and a higher classification and verification performance.

In the final chapter, in the context of the PhD thesis of Yiqiang Chen [116], we studied non-linear similarity metric learning approaches employing more complex SNN models based on deeper CNN architectures. Our work focused here on the application of person re-identification in images coming from several cameras with non-overlapping views. We showed that the use of pedestrian attributes in the learnt similarity metric, and different models that are specific to different body orientations as well as person group context largely improves the re-identification performance.

From a machine learning point of view, in the presented work in the first part, one can notice a trend from *off-line* learnt models (*e.g.* for track creation and removal) to more *on-line* learning (*e.g.* for the objects' appearance). There is a natural need for such algorithms in on-line visual tracking approaches, and on-line data processing methods in general. However, there is a considerable risk of model drift because very few training data is usually available, and the model needs to be adapted continuously to changes of the object or environment without "forgetting" previously acquired relevant information (the "stability-plasticity" dilemma). To address this problem, we first proposed an effective on-line learnt model for pixel-wise detection and segmentation and a *co-training* framework, where the estimation of one model (detector) is used to update the model of the other one (segmentation) (and vice versa). And in the work on scene context, this idea is extended to several models, where only the selected model updates all the other ones at each point in time. Nevertheless, more principled methods for on-line learning (with neural networks) and the better integration and understanding of contextual information are still needed.

Another trend of the presented work is from supervised learning used for MOT, to algorithms that are only *weakly supervised*, which we have employed for the similarity metric learning with SNNs. We have presented several new objective functions (tuple-based, polar sine-based, Rank-triplet *etc.*) and example selection strategies that improved the training and the performance of SNNs for various different applications. We further introduced original neural architectures and training strategies that allowed us to introduce semantic external knowledge. In this way, we were able to combine different models that have been trained in a supervised and weakly supervised way, according to the quantity of data as well as the quantity and type of annotation that is available.

To summarise, we made several significant contributions in visual object tracking to improve the long-term performance and to improve the robustness of the visual appearance models and on-line learning. Nevertheless, many challenges still remain. Several recent tracking methods are based on deep CNNs and showed excellent performance on public benchmarks. However, there are two major issues to be considered. First, these models cannot be created from very few training data as in short video sequences. Thus, they need to be pre-trained on large datasets

such as ImageNet and "fine-tuned" on-line on a given test video. This pre-training introduces a large bias on the type of videos and the type of objects to be tracked. Also, the on-line learning or fine-tuning of neural networks with such non-stationary data is a topic that has not been studied much in machine learning. We will come back to this issue in the perspectives. The second issue is the computational complexity of deep CNNs. Apparently, such models can operate in real-time with recent GPUs. But, the memory requirements and especially the power consumption make their usage mostly impossible or at least impractical for smaller-scale hardware, such as embedded systems. We will also come back to this in the perspectives.

Concerning similarity metric learning, we made several important contributions to improve their convergence and performance of SNNs for verification, classification and ranking, both on the theoretical and practical side. With the current revival and exaltation for neural networks, there has been some recent work and advances on metric learning with deep neural networks and deep CNN (as ours in chapter 8). However, as some of these models are very complex and require a significant amount of training data, some fundamental questions arise. For example, how generic is the learnt similarity metric, *i.e.* how does it cope with unseen data? According to some of our experiments with large amounts of training data, the difference in classification performance between a supervised and a weakly supervised model seems to decrease. This may suggest that the learnt metric just "compares" the given test data to the learnt examples by computing a type of non-linear distance, which might give very poor results when the test data distribution is too far from the training data. We will further investigate these phenomena in our future work. In any case, the inclusion of prior knowledge into the metric learning showed to be a very effective approach. We showed that external semantic information improves the overall performance, but as labels may not be available abundantly it would also be interesting, in the future, to investigate how *unsupervised* learning may help the metric learning. This may also be used to better define or learn the somewhat vague notions of "similarity" and "dissimilarity" imposed by traditional SNN approaches.

9.2 Perspectives

There are numerous perspectives and possible future directions for our research, and, in the following, I will present the most important ones. My colleagues and I have the chance to have obtained the funding for several PhD students that will start soon or have started recently and that will give us the opportunity to do research on these topics. I will co-supervise them with my colleagues from LIRIS and other laboratories, which will further allow me to strengthen existing collaborations within and between different teams ("Imagine", "Data Mining and Machine Learning" (DM2L) and "Multi-Agent Systems" (SMA)) and start new collaborations with others (Laboratoire Hubert Curien (LHC), Saint Étienne, ENS Lyon, UCLy). This evolution in terms of funding and new supervision activities as well as the identified new research directions outlined in the following give me a strong motivation for my habilitation and to build up my own research group.

More specifically, the following topics will be the focus of my research for the next 4–6 years.

9.2.1 On-line and sequential similarity metric learning

In 2018, Christophe Garcia and I continued our collaboration with Grégoire Lefebvre at Orange Labs, Grenoble, related to the work that has been carried out during the PhD thesis of Samuel Berlemont (*c.f.* chapter 7). Within the context of a newly funded PhD, being performed by Paul Compagnon, we will explore and develop new methods and techniques for SNN-based

similarity metric learning of temporal sequences of mobile sensor data. The goal here is to discover and recognise patterns and routines, for example of elderly persons, using inertial sensor data, *e.g.* from a mobile phone, a smart watch *etc.* To this end, we will study new approaches based on recurrent SNNs and on unsupervised and weakly supervised learning that will allow to automatically recognise similar patterns that are recurring and others that are more exceptional or abnormal in order to detect when there is a degradation in the autonomy or any other problem related to the health of the person. Very little work has been done so far on such recurrent SNN-based similarity metric learning approaches to characterise and classify sequential data.

Another work on metric learning on images has started with the PhD that I am co-supervising with Christophe Garcia and Pierre Letissier from the "Institut National de l'Audiovisuel" (INA) in Paris. In this upcoming work that will be conducted by Thomas Petit, we will study new approaches based on deep convolutional SNN for the indexation of videos and broadcasts by recognising faces in large-scale face image datasets. A part from the large training data volume, we will focus on two scientific aspects here. First, we will investigate how such a complex SNN similarity metric can generalise to unseen faces, and how it can effectively and incrementally learn from sequential streams of (face) data and user annotation. And second, we will seek to develop new compressed, robust representations of faces (particular embeddings, hash codes *etc.*) to allow for an efficient indexation and retrieval of images and videos containing faces or possibly other specific content.

9.2.2 Autonomous developmental learning for intelligent vision

During the last two years, in a lab-internal project between the teams Imagine (Christophe Garcia and myself, INSA Lyon) and SMA (Fréderic Armetta, Mathieu Lefort, University Lyon 1), we conducted exploratory research on new approaches for learning visual representation to recognise and track objects in videos in a completely *unsupervised* way and with *minimum prior knowledge*. Two Master students supported us on this ambitious project, and we were able to develop some novel ideas and approaches [19] based on developmental and constructivist learning and SNNs. Figure 9.1 illustrates the principle procedure. Given a video stream, a first algorithm detects salient image regions in each frame based on low-level visual features (colour, edges, motion *etc.*). Then, using temporal consistent pairs of detected image regions, visual representations of candidate objects are learnt on-line with a Siamese CNN architecture. The output of the neural network forms an embedding space where similar objects are close and different objects further apart. After a bootstrapping learning phase, this learnt model can in turn be used to better detect and recognise the objects in the scene. To improve the embedding and the convergence, we extended this approach with multi-task learning of image reconstruction and object segmentation and with an attention mechanism that focuses on different objects in the scene over time.

Many challenges and open issues remain in this project: for example, how to learn incrementally and update an optimal representation from a continuous stream of (visual) data without any prior knowledge, what are the perception, attention and memory mechanisms that allow for plastic and abstract representations that are efficient for recognising known objects and for transferring learnt knowledge to unknown environments, and how can higher-level information be integrated in the model and be used to help this autonomous process, *e.g.* the scene context, higher-level temporal or causal information, sensori-motor stimuli, affordances *etc.* These fundamental issues will be studied in the context of the PhD thesis of Ruiqi Dai, that I will co-supervise with my colleagues from the SMA team and a researcher in humanities (philosophy), Matthieu Guillermin from the "Institut Catholique de Lyon".

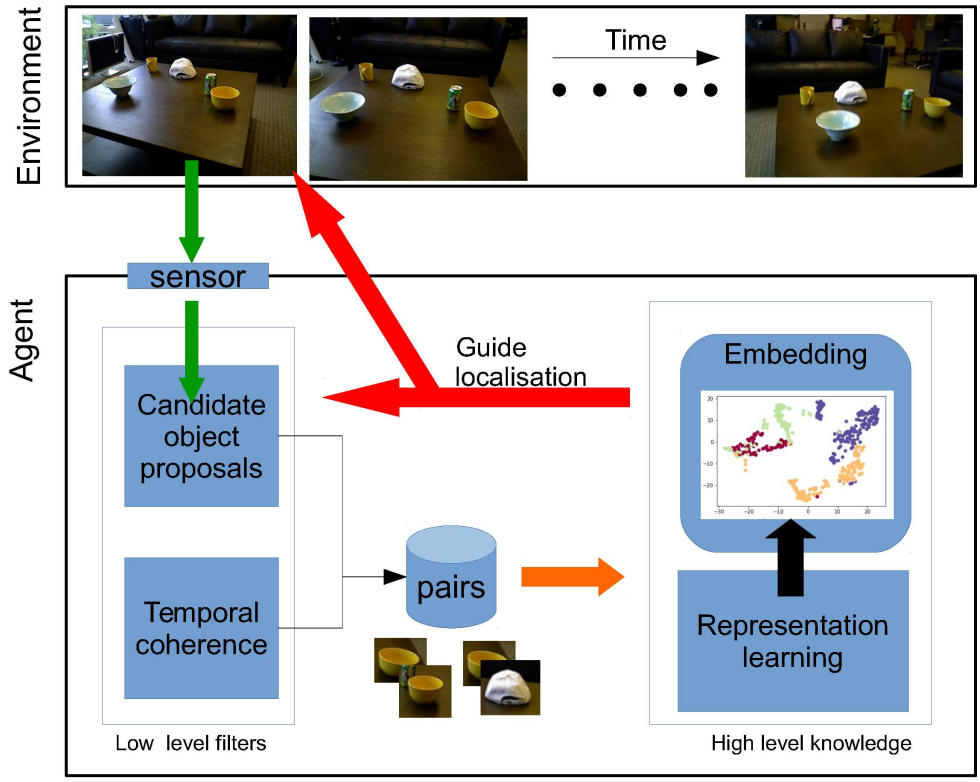


Figure 9.1: Our autonomous perceptive learning approach.

9.2.3 Neural network model compression

With the advent of General-Purpose GPU computing and large-scale High-Performance Computing (HPC) applied to CNNs, more and more complex, deep neural network models have been developed requiring large amounts of computational and memory resources for training and inference. However, it has been shown that the redundancy in such models is very large, and that they can be reduced in size by several orders of magnitude using specific quantisation, factorisation or pruning techniques without any major negative impact on the overall (classification) performance.

In our recent collaboration with Renato Cintra and André Leite from the Federal Pernambuco University, Recife, Brazil, we proposed an approach that approximates the parameters of a trained neural network with so-called dyadic rationals, *i.e.* powers of two, and Canonical Signed Digit-coded factors. In this way, we were able to obtain multiplication-free approximations of the models, *i.e.* requiring only bit shifts and additions, paving the way for very low-complexity hardware implementations of deep neural networks. We showed the effectiveness of our approach [1] on a variety of tasks and network architectures, MLPs and CNNs, including the well-known AlexNet trained on ImageNet [220].

This is an exciting research area that we will continue to explore. It raises also some fundamental questions and issues that we will tackle in the near future. For example, instead of reducing the complexity of already *trained* neural networks, can we develop an algorithm that directly learns such sparse or simplified models while maintaining the same classification performance? In a PhD thesis starting soon in the context of an ambitious research project called "Academics" between LIRIS, LHC, LIP and the Physics Laboratory of ENS Lyon and funded by

the IDEX Lyon Scientific Breakthrough program, we will investigate new approaches to learn sparse neural networks applied to large-scale problems related to complex dynamical systems (*e.g.* climate models, social networks). To this end, with my colleagues from the DM2L team, Marc Plantevit and Céline Robardet, we will work on the combination of specific data mining and feature selection methods to reduce the noise and redundancy in the training data as well as in the model. In particular, we will try to identify frequent and typical activation patterns of a trained neural network graph, layer-wise and across several layers, in order to remove (*i.e.* prune) redundant or useless neurons and to understand the responsibilities of different parts of the model that we will then try to factorise and combine. A preliminary study (Master thesis) on this topic gave promising experimental results.

We will also investigate methods for building simpler and more sparse models directly during training. Recently, approaches that construct neural network architectures that incrementally and iteratively grow (*e.g.* Neural Architecture Search, Network Morphism) may provide an interesting starting point.

In the PhD thesis of Guillaume Anoufa on far-field object detection and recognition in video streams that will in 2019 and that I will co-supervise with Christophe Garcia and Nicolas Bélanger from Airbus Helicopters, we will also tackle this problem of model simplification and compression with CNNs. We will approach this from the training data side and with semi-supervised learning, *i.e.* we will create synthetic annotated images for training a classifier. Among other issues, we will investigate how the choice and type of synthetic data influences the performance and capability to automatically build a neural model of reduced size and complexity and increased generalisation capacity.

9.2.4 From deep learning to deep understanding

Exploring and developing new learning strategies and neural network architectures, possibly in a self-organising and autonomous way, may provide numerous further perspectives. First, drastically reducing the computational complexity and memory requirements of deep neural networks will not only allow for a efficient deployment in embedded devices (the Internet of Things) giving them increased AI capabilities but also enable powerful machine learning-based modelling of complex large-scale problems such as climate and weather prediction in physics or brain signal analysis and interpretation in Neuroscience.

Another issue that gains increasing interest in the AI and Machine Learning research communities is the fact that trained deep neural network models and their inference are difficult to interpret and explain for humans, which leads to an increasing interest in "Explainable AI" approaches. Much work is still to be done to introduce this "explainability" and "interpretability" into neural network-based models, and this will be one perspective of our future work on model simplification and sparsity. That is, we will seek to learn models that favour the identification and isolation of processing paths or sub-models allowing to more easily incorporate and learn semantic knowledge, probably of symbolic nature. This may also raise and hopefully answer some questions on causality. The goal here is formally and statistically analyse input-output correlations at different levels of the trained model, and, further, to develop new training strategies that specifically create such correlations favouring explainable representations, functions and processing paths.

The automatic learning of higher-level concepts, and the effective mechanisms of storing, processing, and transforming them clearly forms the foundation of a strong AI that is able to memorise, recognise and to reason. I believe that much work has still to be done in this regard, and that this can only be achieved in a multi-disciplinary approach involving researchers from AI,

Machine Learning, Robotics, Computer Vision, Cognitive Science and Psychology, Neuroscience, Physics and Mathematics among others. I am keen to accept this challenge in the future, and, in the long term, I hope that we will not only be able to make significant progress towards a general AI but, more importantly, that these results will help us to better understand human intelligence and the human brain as well as the world that surrounds us.

Publications

Journals

[1] Renato J. Cintra, Stefan Duffner, Christophe Garcia, and André Leite. Low-complexity approximate Convolutional Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, March 2018.

[2] Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, and Christophe Garcia. Class-balanced siamese neural networks. *Neurocomputing*, 273:47–56, 2018.

[3] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Deep and low-level feature-based attribute learning for person re-identification. In *Image and Vision Computing*, 2018.

[4] Lilei Zheng, Stefan Duffner, Khalid Idrissi, Christophe Garcia, and Atilla Baskurt. Pairwise identity verification via linear concentrative metric learning. *IEEE Transactions on Cybernetics*, 48(1):324–335, 2018.

[5] Stefan Duffner and Christophe Garcia. Fast pixelwise adaptive visual tracking of non-rigid objects. *IEEE Transactions on Image Processing*, 26(5):2368–2380, 2017.

[6] S. Duffner and C. Garcia. Using discriminative motion context for on-line visual object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(12):2215–2225, 2016.

[7] S. Duffner and C. Garcia. Visual focus of attention estimation with unsupervised incremental learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(12):2264–2272, 2016.

[8] Lilei Zheng, Stefan Duffner, Khalid Idrissi, Christophe Garcia, and Atilla Baskurt. Siamese Multi-layer Perceptrons for dimensionality reduction and face identification. *Multimedia Tools and Applications*, 75(9):5055–5073, 2016.

[9] T. Nakashika, T. Yoshioka, T. Takiguchi, Y. Ariki, S. Duffner, and C. Garcia. Convolutive Bottleneck Network with Dropout for Dysarthric Speech Recognition. *Transactions on Machine Learning and Artificial Intelligence*, pages 1–15, April 2014.

[10] S. Duffner, C. Liu, and J.-M. Odobez. Leveraging colour segmentation for upper-body detection. *Pattern Recognition*, 47(6):2222–2230, 2014.

[11] P. Motlicek, S. Duffner, D. Korchagin, H. Bourlard, C. Scheffler, J.-M. Odobez, O. Thiergart, G. Del Galdo, and F. Kuech. Real-time audio-visual analysis for multiperson videoconferencing. *Advances in Multimedia*, 2013, August 2013.

[12] S. Duffner and Jean-Marc Odobez. A track creation and deletion framework for long-term online multi-face tracking. *IEEE Transactions on Image Processing*, 22(1):272–285, 2013.

[13] S. Duffner, P. Motlicek, and D. Korchagin. The TA2 database – a multi-modal database from home entertainment. *International Journal of Computer and Electrical Engineering*, 4(5):670–673, 2012.

[14] A. Herbulot, S. Jehan-Besson, S. Duffner, M. Barlaud, and G. Aubert. Segmentation of vectorial image features using shape gradients and information measures. *Journal of Mathematical Imaging and Vision*, 25(3):365–386, October 2006.

International Conferences and Workshops

[15] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Pedestrian attribute recognition with part-based CNN and combined feature representations. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, Funchal, Portugal, January 2018.

[16] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Person re-identification using group context. In *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2018.

[17] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Person re-identification with a body orientation-specific convolutional neural network. In *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2018.

[18] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Similarity learning with listwise ranking for person re-identification. In *Proceedings of the International Conference on Image Processing (ICIP)*, 2018.

[19] Nawel Medjkoune, Frédéric Armetta, Mathieu Lefort, and Stefan Duffner. Autonomous object recognition in videos using siamese neural networks. In *EUCognition Meeting (European Society for Cognitive Systems) on "Learning: Beyond Deep Neural Networks"*, Zurich, Switzerland, November 2017.

[20] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Triplet CNN and pedestrian attribute recognition for improved person re-identification. In *Proceedings of the International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Lecce, Italy, August 2017.

[21] Smrity Bhattarai, Arjuna Madanayake, Renato J. Cintra, Stefan Duffner, and Christophe Garcia. Digital architecture for real-time CNN-based face detection for video processing. In *IEEE Cognitive Communications for Aerospace Applications Workshop (CCAA)*, Cleveland, USA, June 2017.

[22] Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, and Christophe Garcia. Polar sine based siamese neural network for gesture recognition. In *Proceedings of the International Conference on International Conference on Artificial Neural Networks (ICANN)*, Barcelona, Spain, 2016.

[23] Matej Kristan, Jiri Matas, Aleš Leonardis, Michael Felsberg, Luka Čehovin, Gustavo Fernandez, Tomas Vojir, Gustav Häger, Georg Nebehay, Roman Pflugfelder, and Stefan Duffner et al. The Visual Object Tracking VOT2015 challenge results. In *ICCV (Workshop)*, December 2015.

[24] Samuel Berlemont, Gregoire Lefebvre, Stefan Duffner, and Christophe Garcia. Siamese neural network based similarity metric for inertial gesture classification and rejection. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, May 2015.

[25] Jiwen Lu, Junlin Hu, Venice Erin Liong, Xiuzhuang Zhou, Andrea Bottino, Ihtesham Ul Islam, Tiago Figueiredo Vieira, Xiaoqian Qin, Xiaoyang Tan, Songcan Chen, Yosi Keller, Shahar Mahpod, Lilei Zheng, Khalid Idrissi, Christophe Garcia, Stefan Duffner, Atilla Baskurt, Modesto Castrillon-Santana, and Javier Lorenzo-Navarro. The FG 2015 Kinship Verification in the Wild Evaluation. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, May 2015.

[26] Lilei Zheng, Khalid Idrissi, Christophe Garcia, Stefan Duffner, and Atilla Baskurt. Triangular similarity metric learning for face verification. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, May 2015.

[27] Lilei Zheng, Khalid Idrissi, Christophe Garcia, Stefan Duffner, and Atilla Baskurt. Logistic similarity metric learning for face verification. In *Proceedings of the International Conference on International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015.

[28] Salma Moujtahid, Stefan Duffner, and Atilla Baskurt. Classifying global scene context for on-line multiple tracker selection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[29] Salma Moujtahid, Stefan Duffner, and Atilla Baskurt. Coherent selection of independent trackers for real-time object tracking. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2015.

[30] Toru Nakashika, Toshiya Yoshioka, Tetsuya Takiguchi, Yasuo Ariki, Stefan Duffner, and Christophe Garcia. Dysarthric speech recognition using a convolutive bottleneck network. In *IEEE International Conference on Signal Processing (ICSP)*, pages 505–509, October 2014.

[31] S. Duffner and C. Garcia. Exploiting contextual motion cues for visual object tracking. In *Workshop on Visual Object Tracking Challenge (VOT2014) - ECCV*, pages 1–12, September 2014.

[32] Matej Kristan, Roman Pflugfelder, Ales Leonardis, Jiri Matas, Luka Cehovin, Georg Nebehay, Tomas Vojir, Gustavo Fernandez, and Stefan Duffner et al. The Visual Object Tracking VOT2014 challenge results. In *Workshop on Visual Object Tracking Challenge (VOT2014) - ECCV*, LNCS, pages 1–27, September 2014.

[33] S. Duffner, S. Berlemont, G. Lefebvre, and C. Garcia. 3D gesture classification with convolutional neural networks. In *Proceedings of the International Conference on International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014.

[34] S. Duffner and C. Garcia. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In *Proceedings of the International Conference on International Conference on Computer Vision (ICCV)*, Sidney, AUS, December 2013.

[35] S. Duffner and C. Garcia. Unsupervised online learning of visual focus of attention. In *Proceedings of the International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Kraków, PL, August 2013.

[36] D. Korchagin, S. Duffner, P. Motlicek, and C. Scheffler. Multimodal cue detection engine for orchestrated entertainment. In *Proceedings of the International Conference on MultiMedia Modeling*, Klagenfurt, Austria, January 2012.

[37] D. Korchagin, P. Motlicek, and S. Duffner. Just-in-time multimodal association and fusion from home entertainment. In *IEEE International Conference on Multimedia and Expo (ICME), Workshop on Multimodal Audio-based Multimedia Content Analysis (MAMCA)*, Barcelone, Espagne, July 2011.

[38] S. Duffner and Jean-Marc Odobez. Exploiting long-term observations for track creation and deletion in online multi-face tracking. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, Santa Barbara, USA, March 2011.

[39] S. Duffner, D. Korchagin, and P. Motlicek. The TA2 database - a multi-modal database from home entertainment. In *International Conference on Signal Acquisition and Processing (ICSAP)*, Singapour, February 2011.

[40] S. Duffner, J.-M. Odobez, and E. Ricci. Dynamic partitioned sampling for tracking with discriminative features. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009.

[41] S. Duffner and C. Garcia. Robust face alignment using convolutional neural networks. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, Funchal, Portugal, January 2008.

[42] S. Duffner and C. Garcia. Face recognition using non-linear image reconstruction. In *Proceedings of the International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 459–464, London, UK, September 2007.

[43] S. Duffner and C. Garcia. An online backpropagation algorithm with validation error-based adaptive learning rate. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, volume 1, pages 249–258, Porto, Portugal, September 2007.

[44] S. Roux, F. Mamalet, C. Garcia, and S. Duffner. An embedded robust facial feature detector. In *International Conference on Machine Learning and Signal Processing (MLSP)*, pages 170–175, Thessaloniki, Greece, August 2007.

[45] C. Garcia and S Duffner. Facial image processing with convolutional neural networks. In *International Workshop on Advances in Pattern Recognition (IWAPR)*, pages 97–108, Plymouth, United Kingdom, July 2007.

[46] S. Duffner and C. Garcia. A neural scheme for robust detection of transparent logos in TV programs. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, volume 2, pages 14–23, Athens, Greece, September 2006.

[47] S. Duffner and C. Garcia. A connexionist approach for robust and precise facial feature detection in complex scenes. In *Fourth International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 316–321, Zagreb, Croatia, September 2005.

National Conferences

[48] Yiqiang Chen, Stefan Duffner, Andrei Stoian, Jean-Yves Dufour, and Atilla Baskurt. Ré-identification de personne avec perte min-triplet. In *16e journées francophones des jeunes chercheurs en vision par ordinateur*, Colleville-sur-mer, France, June 2017.

[49] A. Lehuger, S. Duffner, and C. Garcia. A robust method for automatic player detection in sport videos. In *Compression et Représentation des Signaux Audiovisuels (CORESA)*, Montpellier, France, November 2007.

[50] S. Duffner and C. Garcia. A hierarchical approach for precise facial feature detection. In *Compression et Représentation des Signaux Audiovisuels (CORESA)*, pages 29–34, Rennes, France, November 2005.

[51] S. Jehan-Besson, S. Duffner, A. Herbulot, M. Barlaud, and Aubert G. Utilisation des gradients de forme et des contours actifs basés régions pour la segmentation des vecteurs mouvement. In *GRETSI*, Louvain la Neuve, Belgique, September 2005.

Book Chapters

[52] M. Everingham, A. Zisserman, C. K. I. Williams, L. Van Gool, A. Moray, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. D. R. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, T. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, , and J. Zhang. *The 2005 PASCAL Visual Object Classes Challenge, Selected Proceedings of the first PASCAL Challenges Workshop*. Lecture Notes in Artificial Intelligence, Springer, 2006.

Patents

[53] C. Garcia and S. Duffner. Procédé de recadrage d'images de visage", FR20060053734, WO2007FR51900. 2006.

[54] C. Garcia and S. Duffner. Procédé de reconnaissance de visages par reconstruction croisée non linéaire FR20060055929, WO2007FR52569. 2006.

[55] C. Garcia and S. Duffner. Système et procédé de localisation de points d'intérêt dans une image d'objet mettant en œuvre un réseau de neurones, FR20050003177, US20060910159, WO2006EP61110. 2005.

Other

[56] S. Duffner. *Face Image Analysis With Convolutional Neural Networks*. PhD thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, Freiburg im Breisgau, Germany, 2007.

[57] S. Duffner. Spatio-temporal segmentation of moving objects in image sequences. Master's thesis, École Nationale Supérieure d'Ingénieurs de Caen (ENSICAEN), Caen, France, 2004.

Bibliography

[58] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 798–805, June 2006.

[59] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 798–805, 2006.

[60] Chad Aeschliman, Johnny Park, and Avinash C. Kak. A probabilistic framework for joint segmentation and tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1371–1378, June 2010.

[61] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3916, 2015.

[62] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 469–481. Springer, 2004.

[63] O. Akin, E. Erdem, A. Erdem, and K. Mikolajczyk. Deformable part-based tracking by coupled global and local correlation filters. *Journal of Visual Communication and Image Representation*, 38:763–774, 2016.

[64] A. Akl and S. Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010.

[65] S. R. Arashloo and J. Kittler. Efficient processing of MRFs for unconstrained-pose face recognition. In *Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[66] Shervin Rahimzadeh Arashloo and Josef Kittler. Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features. *IEEE Transactions on Information Forensics and Security*, 9(12):2100–2109, 2014.

[67] Stylianos Asteriadis, Kostas Karpouzis, and Stefanos Kollias. Visual focus of attention in non-calibrated environments using gaze estimation. *International Journal of Computer Vision*, 107(3):293–316, 2013.

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

[69] Sileye O. Ba and Jean-Marc Odobez. A Rao-Blackwellized mixed state particle filter for head pose tracking. In *Proceedings of ICMI Workshop on Multi-modal Multi-party Meeting Processing (MMMP)*, pages 9–16, 2005.

[70] Sileye O. Ba and Jean-Marc Odobez. Evaluation of multiple cue head pose estimation algorithms in natural environements. In *Proceedings of ICME*, pages 1330–1333, 2005.

[71] Sileye O. Ba and Jean-Marc Odobez. Recognizing visual focus of attention from head pose in natural meetings. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, 39(1):16–33, February 2009.

[72] Sileye O. Ba and Jean-Marc Odobez. Multiperson visual focus of attention from head pose and meeting contextual cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):101–116, 2011.

[73] Sileye O. Ba, Hayley Hung, and Jean-Marc Odobez. Visual activity context for focus of attention estimation in dynamic meetings. In *Proceedings of ICME*, pages 1424–1427, 2009.

[74] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, December 2009.

[75] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1218–1225, 2014.

[76] Christian Bailer, Alain Pagani, and Didier Stricker. A superior tracking approach: Building a strong tracker through fusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 170–185, 2014.

[77] Slawomir Bak, Etienne Corvee, Francois Bremond, and Monique Thonnat. Person re-identification using spatial covariance regions of human body parts. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010.

[78] Slawomir Bak, Sofia Zaidenberg, Bernard Boulay, and François Bremond. Improving person re-identification by viewpoint cues. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 175–180. IEEE, 2014.

[79] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 1981.

[80] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1830–1837, June 2012.

[81] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz. Fast high dimensional vector multiplication face recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013.

[82] M. Basseville. Detecting changes in signals and systems – a survey. *Automatica*, 24: 309–326, 1988.

[83] L. Bazzani, D. Tosato, M. Cristani, M. Farenzena, G. Pagetti, G. Menegaz, and Murino V. Social interactions by visual focus of attention in a three-dimensional environment. *Expert Systems*, 30(2):115–127, 2013.

[84] Vasileios Belagiannis, Falk Schubert, Nassir Navab, and Slobodan Ilic. Segmentation based particle filtering for real-time 2d object tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 1–14, 2012.

[85] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *Computing Research Repository*, abs/1306.6709, 2013.

[86] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3457–3464, June 2011.

[87] Ben Benfold and Ian Reid. Unsupervised learning of a scene-specific coarse gaze estimator. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[88] J. Berclazi, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 (9):1806–1819, 2011.

[89] T. Berg and P. N. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 1, page 5, 2012.

[90] Samuel Berlemont. *Automatic non linear metric learning : Application to gesture recognition.* Thesis, Université de Lyon, February 2016.

[91] M. Bertalmio, G. Sapiro, and G. Randall. Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):733–737, 2000.

[92] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[93] S. M. Bhandarkar and X. Luo. Integrated detection and tracking of multiple faces using particle filtering and optical flow-based elastic matching. *Computer Vision and Image Understanding*, 113:708–725, 2009.

[94] Charles Bibby and Ian Reid. Robust real-time visual tracking using pixel-wise posteriors. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2008.

[95] S. T Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 232–237, 1998.

[96] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[97] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, 2011.

[98] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, 2011.

[99] Jane Bromley, Isabelle Guyon, Yann Lecun, Eduard Säckinger, and Roopak Shah. Signature Verification using a "Siamese" Time Delay Neural Network. In *Proceedings of NIPS*, 1994.

[100] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, pages 89–96, 2005.

[101] Christopher J Burges, Robert Ragno, and Quoc V Le. Learning to rank with nonsmooth cost functions. In *NIPS*, pages 193–200, 2007.

[102] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1846–1853, 2013.

[103] Yinghao Cai, Valtteri Takala, and Matti Pietikainen. Matching groups of people by covariance descriptor. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2744–2747. IEEE, 2010.

[104] Zhaowei Cai, Longyin Wen, Jianwei Yang, Zhen Lei, and Stan Z. Li. Structured visual tracking with dynamic graph. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 86–97, 2012.

[105] K Cannons, J Gryn, and R Wildes. Visual tracking using a pixelwise spatiotemporal oriented energy representation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 511–524, 2010.

[106] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013.

[107] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning (ICML)*, pages 129–136, 2007.

[108] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.

[109] Chen Chen, R. Jafari, and N. Kehtarnavaz. Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors. *IEEE Transactions on Human-Machine Systems*, 45(1):51–61, February 2015.

[110] Cheng Chen and Jean-Marc Odobez. We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2012.

[111] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3025–3032. IEEE, 2013.

[112] Ke Chen and Ahmad Salman. Extracting Speaker-Specific Information with a Regularized Siamese Deep Network. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 298–306, 2011.

[113] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[114] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. A multi-task deep network for person re-identification. In *AAAI*, pages 3988–3994, 2017.

[115] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep learning multi-scale representations. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2590–2600, 2017.

[116] Yiqiang Chen. *Person Re-identification in Images with Deep Learning.* Thesis, Université de Lyon, October 2018.

[117] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1335–1344, 2016.

[118] Dong Seon Cheng, Marco Cristani, Michele Stoppa, Loris Bazzani, and Vittorio Murino. Custom pictorial structures for re-identification. In *Bmvc*, volume 1, page 6. Citeseer, 2011.

[119] P Chockalingam, N Pradeep, and S Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2009.

[120] J. Choi, H.J. Chang, J. Jeong, Y. Demiris, and J.Y. Choi. Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[121] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546. IEEE, 2005.

[122] Robert T. Collins and Yanxi Liu. On-line selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[123] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 142–149, 2000.

[124] Daniel Cremers and G Funka-Lea. Dynamical statistical shape priors for level set based sequence segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, 2006.

[125] Zhen Cui, Wen Li, Dong Xu, Shiguang Shan, and Xilin Chen. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3554–3561. IEEE, 2013.

[126] Julien Cumin and Grégoire Lefebvre. A priori Data and A posteriori Decision Fusions for Human Action Recognition. In *11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, February 2016.

[127] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2014.

[128] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[129] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[130] J. G. Daugman. Complete discrete 2-D gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1169–1179, 1988.

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

[132] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):788–798, 2011.

[133] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the ACM international conference on Multimedia*, pages 789–792. ACM, 2014.

[134] Mert Dikmen, Emre Akbas, Thomas S Huang, and Narendra Ahuja. Pedestrian recognition with a learned metric. In *Asian conference on Computer vision*, pages 501–512. Springer, 2010.

[135] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10): 2993–3003, 2015.

[136] TB Dinh, Nam Vo, and G Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2011.

[137] S. L. Dockstader, N. S. Imennov, and A. M. Tekalp. Markov-based failure prediction for human motion analysis. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1283–1288, Nice, France, 2003.

[138] Ligeng Dong, Huijun Di, Linmi Tao, Guangyou Xu, and Patrick Oliver. Visual focus of attention recognition in the ambient kitchen. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 548–559, 2009.

[139] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[140] D. Du, H. Qi, W. Li, L. Wen, Q. Huang, and S. Lyu. Online deformable object tracking based on structure-aware hyper-graph. *IEEE Transactions on Image Processing*, 25(8): 3572–3584, August 2016.

[141] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of IEEE*, 90(7):1151–1163, 2002.

[142] J. Fan, W. Xu, Y. Wu, and Y. Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.

[143] Jialue Fan, Wei Xu, Ying Wu, and Yihong Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.

[144] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2360–2367, 2010.

[145] S. S. Fels and G. E. Hinton. Glove-Talk: a neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4(1), 1993.

[146] M. Felsberg. Enhanced distribution field tracking using channel representations. 2013.

[147] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010.

[148] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2):103–113, 1989.

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

[150] Daniel Freedman and Tao Zhang. Active contours for tracking distributions. *IEEE Transactions on Image Processing*, 13(4):518–526, April 2004.

[151] E. G. Freedman and D. L. Sparks. Eye-head coordination during head-unrestrained gaze shifts in rhesus monkeys. *Journal of Neurophysiology*, 77:2328—2348, 1997.

[152] K. Fukushima. Neocognitron: A self-organizing neural-network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[153] Kenneth Alberto Funes Mora and Jean-Marc Odobez. Gaze estimation from multimodal kinect data. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Gesture Recognition*, June 2012.

[154] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2188–202, November 2011.

[155] J. Gao, J. Xing, W. Hu, and Zhang X. Graph embedding based semi-supervised discriminative tracker. 2013.

[156] C. Garcia and M. Delakis. Convolutional Face Finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004.

[157] Jorge García, Niki Martinel, Gian Luca Foresti, Alfredo Gardel, and Christian Micheloni. Person orientation and feature distances boost re-identification. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 4618–4623. IEEE, 2014.

[158] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Interspeech*, pages 249–252, 2011.

[159] E. Gasca, S. Salda na, V. Velásquez, E. Rendón, and R. Cruz et al. A rejection option for the multilayer perceptron using hyperplanes. *Adaptive and Natural Computing Algorithms*, pages 51–60, 2011.

[160] N. Gengembre and P. Pérez. Probabilistic color-based multi-object tracking with application to team sports. Technical Report 6555, INRIA, 2008.

[161] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[162] Martin Godec and Peter M. Roth. Hough-based tracking of non-rigid objects. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[163] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the International Conference on*, pages 2672–2680, 2014.

[164] H. Grabner and H. Bischof. On-line boosting and vision. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 260–267, June 2006.

[165] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *Proceedings of British Machine Vision Conference (BMVC)*, 2006.

[166] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2008.

[167] Helmut Grabner, Jiri Matas, Luc Van Gool, and Philippe Cattin. Tracking the invisible: Learning where the object might be. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, volume 3, pages 1285–1292, June 2010.

[168] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *IEEE Transactions on Neural Networks*, (18):5–6, 2005.

[169] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 262–275, 2008.

[170] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds. The NIST 2014 speaker recognition i-vector machine learning challenge. In *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[171] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 498–505. IEEE, 2009.

[172] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 309–316. IEEE, 2009.

[173] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006.

[174] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109, October 2016.

[175] Sam Hare, Amir Saffari, and Philip H. S. Torr. Struck: Structured output tracking with kernels. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[176] Sam Hare, Amir Saffari, and Philip H. S. Torr. Efficient online structured output learning for keypoint-based object tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[177] M. Hayhoe and D. Ballard. Eye movements in natural behavior. *TRENDS in Cognitive Sciences*, 9(4):188–194, 2005.

[178] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[179] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *Proceedings of Automatic Face and Gesture Recognition (FG)*, 1996.

[180] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with center-symmetric local binary patterns. In *Proceedings of Indian conference on Computer Vision, Graphics and Image Processing*, pages 58–69. Springer, 2006.

[181] Alexandre Heili, Adolfo López-Méndez, and Jean-Marc Odobez. Exploiting long-term connectivity and visual motion in CRF-based multi-person tracking. *IEEE Transactions on Image Processing*, 2014.

[182] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (3):583–596, 2015.

[183] Ralf Herbrich. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers*, pages 115–132, 2000.

[184] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[185] R. Hess and A. Fern. Discriminatively trained particle filters for complex multi-object tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 240–247, 2009.

[186] N. V. Hieu and B. Li. Cosine similarity metric learning for face verification. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 709–720. Springer, 2011.

[187] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[188] M. Hoffman, P. Varcholik, and J. J. LaViola. Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices. In *Virtual Reality Conference (VR)*, pages 59–66, 2010.

[189] F. G. Hofmann, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete Hidden Markov Models. In *Gesture and Sign Language in Human-Computer Interaction*, volume 1371 of *Lecture Notes in Computer Science*, pages 81–95. 1998.

[190] Zhibin Hong, Xue Mei, and Dacheng Tao. Dual-force metric learning for robust distracter-resistant tracker. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 513–527, 2012.

[191] Zhibin Hong, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. Tracking using multilevel quantizations. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 155–171, 2014.

[192] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. MUlti-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[193] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[194] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1875–1882, 2014.

[195] Yang Hua, Karteek Alahari, and Cordelia Schmid. Occlusion and motion reasoning for long-term tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–187, 2014.

[196] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007.

[197] S. U. Hussain, T. Napoléon, and F. Jurie. Face recognition using local quantized patterns. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012.

[198] Michael Isard and Andrew Blake. CONDENSATION – conditional density propagation for visual tracking. *Proceedings of International Conference on Computer Vision (ICCV)*, 29(1):5–28, 1998.

[199] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[200] A. D. Jepson, J. D. Fleet, and T. F. El-Margaghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25: 1296–1311, 2003.

[201] A.D. Jepson, D.J. Fleet, and T.R. El-Maraghi. Robust online appearance models for visual tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume I, pages I–415–I–422, 2001.

[202] Y. Jin and F. Mokhtarian. Variational particle filter for multi-object tracking. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[203] Z. Kalal, J. Matas, and K. Mikolajczyk. Online learning of robust object detectors during unstable tracking. In *Proceedings of International Conference on Computer Vision (ICCV)(CV workshop)*, pages 1417–1424, 2009.

[204] Z Kalal, J Matas, and K Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[205] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, 2010.

[206] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, July 2012.

[207] S. Kallio, J. Kela, and J. Mantyjarvi. Online gesture recognition system for mobile interaction. In *Proceedings of Systems, Man and Cybernetics*, volume 3, pages 2070–2076 vol.3, 2003.

[208] Meina Kan, Shiguang Shan, Dong Xu, and Xilin Chen. Side-information based linear discriminant analysis for face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

[209] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–506. IEEE, 2004.

[210] D. Kedem, S. Tyree, F. Sha, G. R. Lanckriet, and K. Q. Weinberger. Non-linear metric learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2573–2581, 2012.

[211] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5):285–299, July 2006. ISSN 1617-4909.

[212] ObaidUllah Khalid, J.C. SanMiguel, and Andrea Cavallero. Multi-tracker partition fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, pages –, January 2016.

[213] Sameh Khamis, Cheng-Hao Kuo, Vivek K Singh, Vinay D Shet, and Larry S Davis. Joint learning for attribute-consistent person re-identification. In *ECCV Workshops (3)*, pages 134–146, 2014.

[214] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (11):1805–1918, November 2005.

[215] S. Kim, S. Kwak, J. Feyereisl, and B. Han. Online multi-target tracking by large margin structured learning. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 98–111, 2012.

[216] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2295, 2012.

[217] Matej Kristan, Luka Cehovin, Roman Pflugfelder, Georg Nebehay, Gustavo Fernandez, Jiri Matas, and et al. The Visual Object Tracking VOT2013 challenge results. In *Proceedings of the International Conference on Computer Vision (ICCV) (Workshops)*, 2013.

[218] Matej Kristan, Roman Pflugfelder, Ales Leonardis, Jiri Matas, Luka Cehovin, Georg Nebehay, Tomas Vojir, and Fernández Gustavo et al. The Visual Object Tracking VOT2014 challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV) (Workshops)*, 2014.

[219] Matej Kristan, Jiri Matas, Ales Leonardis, Tomas Vojir, Roman P. Pflugfelder, Gustavo Fernández, Georg Nebehay, Fatih Porikli, and Luka Cehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, November 2016.

[220] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012.

[221] Cheng-Hao Kuo and Ram Nevatia. How does person identity recognition help multi-person tracking? In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1217–1224, 2011.

[222] Junseok Kwon and K.M. Lee. Visual tracking decomposition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1269–1276, 2010.

[223] Junseok Kwon and K.M. Lee. Tracking by sampling trackers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1195–1202, 2011.

[224] Junseok Kwon and Kyoung Mu Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive Basin Hopping Monte Carlo sampling. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[225] S.R.H. Langton, R.J Watt, and I. Bruce. Do the eyes have it? cues to the direction of social attention. *Trends in cognitive sciences*, 4(2):50–59, February 2000.

[226] Oswald Lanz and Roberto Brunelli. Joint bayesian tracking of head location and pose from low-resolution video. In *Multimodal Technologies for Perception of Humans*, pages 287–296, 2007.

[227] Ryan Layne, Timothy M Hospedales, Shaogang Gong, and Q Mary. Person re-identification by attributes. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 2, page 8, 2012.

[228] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Handwritten digit recognition with a back-propagation network. In David Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan Kaufman, Denver, CO, 1990.

[229] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[230] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia. BLSTM-RNN based 3D gesture classification. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 381–388, 2013.

[231] Grégoire Lefebvre and Christophe Garcia. Learning a bag of features based nonlinear metric for facial similarity. In *Proceedings of the International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 238–243, 2013.

[232] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1695–1699. IEEE, 2014.

[233] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV worksh. on statist. learning in comp. vis.*, 2004.

[234] Ido Leichter, Michael Lindenbaum, and Ehud Rivlin. A general framework for combining visual trackers – "black boxes" approach. *International Journal of Computer Vision*, 67 (3):343–363, 2006.

[235] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–79, 2006.

[236] Gilad Lerman and J. Tyler Whitehouse. On D-dimensional D-semimetrics and simplex-type inequalities for high-dimensional sine functions. *Journal of Approximation Theory*, 156(1):52–81, January 2009.

[237] Annan Li, Luoqi Liu, and Shuicheng Yan. Person re-identification by attribute-assisted clothes appearance. In *Person Re-Identification*, pages 119–138. Springer, 2014.

[238] Dangwei Li, Xiaotang Chen, and Kaiqi Huang. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. *Proceedings of the Asian Conference on Pattern Recognition (ACPR)"*, 2015.

[239] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–393, 2017.

[240] H. Li and G. Hua. Hierarchical-PEP model for real-world face recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4055–4064. IEEE, 2015.

[241] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic matching for pose variant face verification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3499–3506. IEEE, 2013.

[242] Hanxi Li, Yi Li, and Fatih Porikli. DeepTrack : Learning Discriminative Feature Representations by Convolutional Neural Networks for Visual Tracking. In *Proceedings of British Machine Vision Conference (BMVC)*, 2014.

[243] Haoxiang Li, Gang Hua, Xiaohui Shen, Zhe Lin, and Jonathan Brandt. Eigen-pep for video face recognition. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 17–33. Springer, 2015.

[244] Quannan Li, Xinggang Wang, Wei Wang, Yuan Jiang, Zhi-Hua Zhou, and Zhuowen Tu. Disagreement-based multi-system tracking. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, volume 7729, pages 320–334, 2012.

[245] Wei Li and Xiaogang Wang. Locally aligned feature transforms across views. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3594–3601, 2013.

[246] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012.

[247] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid:deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 152–159, 2014.

[248] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. In *International Joint Conference on Artificial Intelligence*, 2017.

[249] Y. Li, C. Huang, and R. Nevatia. Learning to associate: HybridBoosted multiple object tracking. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2953–2960, June 2009.

[250] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2197–2206, 2015.

[251] Zhe Lin and Larry S Davis. Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance. In *International symposium on visual computing*, pages 23–34. Springer, 2008.

[252] Giuseppe Lisanti, Iacopo Masi, and Alberto Del Bimbo. Matching people across camera views using kernel canonical correlation analysis. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 10. ACM, 2014.

[253] Giuseppe Lisanti, Niki Martinel, Alberto Del Bimbo, and Gian Luca Foresti. Group re-identification via unsupervised transfer of sparse features encoding. In *Proceedings of the IEEE International Conference on International Conference on Computer Vision (ICCV)*, 2017.

[254] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1313–1320, 2011.

[255] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 2017.

[256] Jiayang Liu, Zhen Wang, Lin Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. In *International Conference on Pervasive Computing and Communications*, pages 1–9, 2009.

[257] Xiao Liu, Mingli Song, Dacheng Tao, Xingchen Zhou, Chun Chen, and Jiajun Bu. Semi-supervised coupled dictionary learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3557, 2014.

[258] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the International Conference on Computer Vision (ICCV)*, December 2015.

[259] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

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

[261] Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):331–345, 2014.

[262] L. Lu, Z. Zhang, H. Shum, Z. Liu, and H. Chen. Model and exemplar-based robust head pose tracking under occlusion and varying expression. In *Proceedings of the IEEE Workshop on Models versus Exemplars in Computer Vision (CVPR-MECV)*, December 2001.

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

[264] Bingpeng Ma, Yu Su, and Frédéric Jurie. Local descriptors encoded by fisher vectors for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 413–422, 2012.

[265] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[266] Lianyang Ma, Xiaokang Yang, and Dacheng Tao. Person re-identification over camera networks using multi-task distance metric learning. *IEEE Transactions on Image Processing*, 23(8):3656–3670, 2014.

[267] Liqian Ma, Hong Liu, Liang Hu, Can Wang, and Qianru Sun. Orientation driven bag of appearances for person re-identification. *arXiv preprint arXiv:1605.02464*, 2016.

[268] J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N Waber. A human-computer interface using symmetry between eyes to detect gaze direction. *IEEE Transactions on Systems, Man, and Cybernetics. Part A*, 38(6):1248–1261, 2008.

[269] E. Maggio, M. Taj, and A. Cavallaro. Efficient multi-target visual tracking using random finite sets. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1016–1027, 2008.

[270] Emilio Maggio and Andrea Cavallaro. Learning scene context for multiple object tracking. *IEEE Transactions on Image Processing*, 18(8):1873–1884, 2009.

[271] A. Mansouri. Region tracking via level set PDEs without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):947–961, 2002.

[272] M. E. Maresca and A. Petrosino. Matrioska: A multi-level approach to fast tracking by learning. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 419–428, 2013.

[273] Jonathan Masci, Michael M. Bronstein, Alexander M. Bronstein, and Jurgen Schmidhuber. Multimodal Similarity-Preserving Hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):824–830, April 2014.

[274] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, et al. Comparison of face verification results on the XM2VTFS database. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, volume 4, pages 858–863. IEEE, 2000.

[275] Tetsu Matsukawa and Einoshin Suzuki. Person re-identification using cnn features learned from combination of attributes. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2428–2433. IEEE, 2016.

[276] Niall McLaughlin, Jesus Martinez del Rincon, and Paul C Miller. Person reidentification using deep convnets with multitask learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):525–539, 2017.

[277] Xue Mei and Haibin Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11): 2259–72, November 2011.

[278] Alexis Mignon and Frédéric Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2666–2672. IEEE, 2012.

[279] D. Mikami, K. Otsuka, and J. Yamato. Memory-based particle filter for face pose tracking robust under complex dynamics. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 999–1006, 2009.

[280] D. Mikami, K. Otsuka, and J. Yamato. Memory-based particle filter for tracking objects with large variation in pose and appearance. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume 3, pages 215–228, 2010.

[281] A. Milan, S. Roth, , and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):58–72, 2014.

[282] C. Morimoto and M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98:4–24, 2005.

[283] Salma Moujtahid. *Exploiting scene context for on-line object tracking in unconstrained environments*. Thesis, Université de Lyon, November 2016.

[284] Panagiotis Moutafis, Mengjun Leng, and Ioannis A Kakadiaris. An overview and empirical comparison of distance metric learning methods. *IEEE Transactions on Cybernetics*, 2016.

[285] Stefan Munder, Christoph Schnorr, and Dariu M. Gavrila. Pedestrian detection and tracking using a mixture of view-based shape–texture models. *IEEE Transactions on Intelligent Transportation Systems*, 9(2), 2008.

[286] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[287] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4): 607–626, April 2009.

[288] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[289] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[290] S. S. Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[291] Sj Nowlan and Jc Platt. A convolutional neural network hand tracker. In *Advances in Neural Information Processing Systems*, pages 8–10, 1995.

[292] Jean-Marc Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.

[293] Jean-Marc Odobez, Daniel Gatica-Perez, and Sileye. O Ba. Embedding motion in model-based stochastic tracking. *IEEE Transactions on Image Processing*, 15(11):3514–3530, 2006.

[294] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley MHAD: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60, 2013.

[295] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Marlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, 2009.

[296] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1): 51–59, 1996.

[297] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 28–39, 2004.

[298] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[299] K Otsuka and J Yamato. Conversation scene analysis with dynamic bayesian network based on visual head tracking. In *Proceedings of the International Conference on Multimedia and Expo*, pages 949–952, 2006.

[300] Jiazhi Ou, Lui Min Oh, Susan R. Fussell, Tal Blum, and Jie Yang. Predicting visual focus of attention from intention in remote collaborative tasks. *IEEE Transactions on Multimedia*, 10(6):1034–1045, 2008.

[301] Abdelmalik Ouamane, Bengherabi Messaoud, Abderrezak Guessoum, Abdenour Hadid, and Mohamed Cheriet. Multi scale multi descriptor local binary features and exponential discriminant analysis for robust face authentication. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 313–317. IEEE, 2014.

[302] Nikos Paragios and Rachid Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[303] Sateesh Pedagadi, James Orwell, Sergio Velastin, and Boghos Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3318–3325, 2013.

[304] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume 1, pages 661–675, Copenhagen, May–June 2002.

[305] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. 92(3): 495–513, 2004.

[306] Frederico Pernici and Alberto Del Bimbo. Object tracking by oversampling local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2538–2551, 2014.

[307] F. Perronnin and C. Dance. Fisher kenrels on visual vocabularies for image categorizaton. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[308] C. Peters, S. Asteriadis, and K Karpouzis. Investigating shared attention with a virtual agent using a gaze-based interface. *Journal on Multimodal User Interfaces*, 3(1–2):119–130, 2009.

[309] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1208, 2011.

[310] C. Plagemann, D. Fox, and W. Burgard. Efficient failure detection on mobile robots using particle filters with gaussian process proposals. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

[311] Fatih Porikli. Inter-camera color calibration by correlation model function. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–133. IEEE, 2003.

[312] Horst Possegger, Thomas Mauthner, and Horst Bischof. In defense of color-based model-free tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2113–2120, 2015.

[313] Bryan James Prosser, Wei-Shi Zheng, Shaogang Gong, Tao Xiang, and Q Mary. Person re-identification by support vector ranking. In *BMVC*, volume 2, page 6, 2010.

[314] T. Pylvänäinen. Accelerometer Based Gesture Recognition Using Continuous HMMs Pattern Recognition and Image Analysis. volume 3522 of *Lecture Notes in Computer Science*, chapter 77, pages 413–430. Berlin, Heidelberg, 2005.

[315] A. M. Qamar, E. Gaussier, J. P. Chevallet, and J. H. Lim. Similarity learning for nearest neighbor classification. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 983–988. IEEE, 2008.

[316] Yogesh Rathi, Namrata Vaswani, Allen Tannenbaum, and Anthony Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1470–1475, August 2007.

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

[318] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692, 2014.

[319] Xiaofeng Ren and Jitendra Malik. Tracking as repeated figure/ground segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[320] D. A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17(1):91–108, 1995.

[321] Elisa Ricci and Jean-Marc Odobez. Learning large margin likelihoods for realtime head pose tracking. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 2593–2596, November 2009.

[322] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006. ISSN 0885-6125.

[323] D. A. Ross, Jongwoo Lim, R.S. Lin, and M.H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77:125–141, 2008.

[324] A. B. Carsten Rother and V. Kolmogorov. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004.

[325] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[326] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof. Online multi-class LPBoost. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3577, June 2010.

[327] Amir Saffari and Christian Leistner. On-line random forests. In *Proceedings of International Conference on Computer Vision (ICCV) (Worksh. on Online Comp. Vis.)*, 2009.

[328] Conrad Sanderson and Brian C Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. In *Advances in Biometrics*, pages 199–208. Springer, 2009.

[329] Jakob Santner, C Leistner, A Saffari, and T Pock. PROST: Parallel robust online simple tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 723–730, 2010.

[330] M Saquib Sarfraz, Arne Schumann, Yan Wang, and Rainer Stiefelhagen. Deep view-sensitive pedestrian attribute inference in an end-to-end model. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[331] T. Schoenemann and D. Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1153–1164, 2009.

[332] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[333] Arne Schumann and Rainer Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1435–1443. IEEE, 2017.

[334] William Robson Schwartz and Larry S Davis. Learning discriminative appearance-based models using partial least squares. In *Conference on Computer Graphics and Image Processing (SIBGRAPI)*, pages 322–329. IEEE, 2009.

[335] Hae Jong Seo and Peyman Milanfar. Face verification using the LARK representation. *IEEE Transactions on Information Forensics and Security*, 6(4):1275–1286, 2011.

[336] L. Sevilla-Lara and E. G. Learned-Miller. Distribution fields for tracking. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pages 1910–1917, 2012.

[337] Yang Shen, Weiyao Lin, Junchi Yan, Mingliang Xu, Jianxin Wu, and Jingdong Wang. Person re-identification with correspondence structure learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3200–3208, 2015.

[338] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. Embedding deep metric for person re-identification: A study against large variations. In *European Conference on Computer Vision*, pages 732–748. Springer, 2016.

[339] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[340] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, Andrew Blake, and Xbox Incubation. Real-time human pose recognition in parts from single depth images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[341] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007.

[342] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 1, page 7, 2013.

[343] S. Singh and M. Markou. An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):396–407, 2004.

[344] Michael Siracusa, Louis-Philippe Morency, Kevin Wilson, John Fisher, and Trevor Darrell. A multi-modal approach for determining speaker location and focus. In *ICMI*, pages 77–80, 2003.

[345] K. Smith, D. Gatica-Perez, J. Odobez, and Sileye Ba. Evaluating multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshop on Empirical Evaluation Methods in Computer Vision*, page 36, June 2005.

[346] Kevin Smith, Sileye O. Ba, Jean-Marc Odobez, and Daniel Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1212–1229, July 2008.

[347] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume 1, pages 605–619, 2010.

[348] Shuran Song and Jianxiong Xiao. Tracking revisited using RGBD camera: Unified benchmark and baselines. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013.

[349] Severin Stalder, Helmut Grabner, and L Van Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *Proceedings of International Conference on Computer Vision (ICCV) (Worksh. on Online Comp. Vis.)*, 2009.

[350] C. Stauffer and W. Grimson. Learning patterns of activity using real time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–767, 2000.

[351] Björn Stenger, Thomas Woodley, and Roberto Cipolla. Learning to track with multiple observers. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2647–2654, 2009.

[352] R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2002.

[353] R. Stiefelhagen, J. Yang, and A. Waibel. Modeling focus of attention for meeting indexing based on multiple cues. *IEEE Transactions on Neural Networks*, 13(4):928–938, July 2002.

[354] Rainer Stiefelhagen, Jie Yang, and Alex Waibel. Modeling focus of attention for meeting indexing. In *Proceedings of the ACM Multimedia*, 1999.

[355] Rainer Stiefelhagen, Jie Yang, and Alex Waibel. Tracking focus of attention for human-robot communication. In *IEEE-RAS International Conference on Humanoid Robots - Humanoids*, Tokyo, Japan, 2001.

[356] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Deep attributes driven multi-camera person re-identification. In *European Conference on Computer Vision*, pages 475–491. Springer, 2016.

[357] Arulkumar Subramaniam, Moitreya Chatterjee, and Anurag Mittal. Deep neural networks with inexact matching for person re-identification. In *Advances in Neural Information Processing Systems*, pages 2667–2675, 2016.

[358] Patrick Sudowe, Hannah Spitzer, and Bastian Leibe. Person attribute recognition with a jointly-trained holistic cnn model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 87–95, 2015.

[359] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1988–1996, 2014.

[360] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1891–1898. IEEE, 2014.

[361] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *Proceedings of the IEEE International Conference on International Conference on Computer Vision (ICCV)*, 2017.

[362] Zhongqian Sun, Hongxun Yao, Shengping Zhang, and Xin Sun. Robust visual tracking via context objects computing. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 509–512, September 2011.

[363] J. S. Supančič and D. Ramanan. Self-paced learning for long-term tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[364] Darrell T., G. Gordon, Harville M., and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, pages 175–185, 2000.

[365] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708. IEEE, 2014.

[366] C. Tian, X. Gao, W. Wei, and H. Zheng. Visual tracking based on the adaptive color attention tuned sparse generative object model. *IEEE Transactions on Image Processing*, 24(12):5236–5248, December 2015.

[367] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1023–1029, 2003.

[368] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *International Journal of Computer Vision*, 48(1):9–19, 2002.

[369] J. Triesch and Christoph v. d. Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001.

[370] Ma. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991.

[371] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *European conference on computer vision*, pages 589–600. Springer, 2006.

[372] Norimichi Ukita, Yusuke Moriguchi, and Norihiro Hagita. People re-identification across non-overlapping cameras using group features. *Computer Vision and Image Understanding*, 144:228–236, 2016.

[373] Daniel A Vaquero, Rogerio S Feris, Duan Tran, Lisa Brown, Arun Hampapur, and Matthew Turk. Attribute-based people search in surveillance environments. In *Proceedings of Workshop on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2009.

[374] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *Proceedings of the IEEE International Conference on European Conference on Computer Vision (ECCV)*, pages 791–808. Springer, 2016.

[375] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–153, 2016.

[376] G. C. Vasconcelos, M. C. Fairhurst, and D. L. Bisset. Investigating the recognition of false patterns in backpropagation networks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 133–137, 1993.

[377] Luka Čehovin, Matej Kristan, and Aleš Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):941–953, April 2013.

[378] Ankit Verma, Ramya Hebbalaguppe, Lovekesh Vig, Swagat Kumar, and Ehtesham Hassan. Pedestrian detection via mixture of cnn experts and thresholded aggregated channel features. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 163–171, 2015.

[379] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[380] Michael Voit and Rainer Stiefelhagen. Tracking head pose and focus of attention with multiple far-field cameras. In *Proceedings of the IEEE International Conference on Multimodal Interfaces (ICMI)*, pages 281–286, 2006.

[381] Michael Voit and Rainer Stiefelhagen. Deducing the visual focus of attention from head pose estimation in dynamic multi-view meeting scenarios. In *Proceedings of ICMI*, pages 173–180, 2008.

[382] Michael Voit and Rainer Stiefelhagen. 3D user-perspective, voxel-based estimation of visual focus of attention in dynamic meeting scenarios. In *Proceedings of the ICMI-MLMI*, 2010.

[383] Tomas Vojir, Jiri Matas, and Jana Noskova. Online adaptive hidden markov model for multi-tracker fusion. *Computer Vision and Image Understanding*, 153:109–119, 2016.

[384] Tomáš Vojíř and Jiri Matas. Robustifying the flock of trackers. In *Computer Vision Winter Workshop*, pages 91–97, 2011.

[385] D. Wang, H. Lu, and C. Bo. Fast and robust object tracking via probability continuous outlier model. *IEEE Transactions on Image Processing*, 24(12):5166–5176, December 2015.

[386] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1288–1296, 2016.

[387] J.-G. Wang and E. Sung. Study on eye gaze estimation. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, 32:332–350, 2002.

[388] Jian-Gang Wang, Eric Sung, and Ronda Venkateswarlu. Eye gaze estimation from a single image of one eye. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 136–143, 2003.

[389] Jin Wang, Zheng Wang, Changxin Gao, Nong Sang, and Rui Huang. Deeplist: Learning deep features with adaptive listwise constraint for person reidentification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):513–524, 2017.

[390] L. Wang, W. Ouyang, X. Wang, and H. Lu. STCT: Sequentially training convolutional networks for visual tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[391] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1–9, 2013.

[392] Naiyan Wang and Dit Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 32, pages 1107–1115, 2014.

[393] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[394] Xiaogang Wang, Gianfranco Doretto, Thomas Sebastian, Jens Rittscher, and Peter Tu. Shape and appearance context modeling. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[395] Li Wei and Shishir K Shah. Subject centric group feature for person re-identification. In *CVPR Workshops*, pages 28–35, 2015.

[396] U. Weidenbacher, G. Layher, P. Bayerl, and H Neumann. Detection of head pose and gaze direction for human-computer interaction. *Perception and Interactive Technologies*, 4021: 9–19, 2006.

[397] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, volume 18, page 1473. MIT; 1998, 2006.

[398] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[399] Longyin Wen, Zhaowei Cai, Zhen Lei, Dong Yi, and S Li. Robust online learned spatio-temporal context model for visual tracking. *IEEE Transactions on Image Processing*, 23 (2):785–796, 2014.

[400] Longyin Wen, Dawei Du, Zhen Lei, Stan Z Li, and Ming-hsuan Yang. JOTS : Joint Online Tracking and Segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[401] D. H. Wilson and A. Wilson. Gesture recognition using the xwand. Technical Report CMU-RI-TR-04-57, Robotics Institute, April 2004.

[402] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008.

[403] C. Wren, A. Zarbayejani, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[404] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. Gesture recognition with a 3-D accelerometer. Ubiquitous Intelligence and Computing, pages 25–38, 2009.

[405] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 789–795, 2003.

[406] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *International Conference on Machine Learning (ICML)*, pages 1192–1199, 2008.

[407] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[408] J. Xiao, R. Stolkin, and A. Leonardis. An enhanced adaptive coupled-layer lgtracker++. 2013.

[409] J. Xiao, R. Stolkin, and A. Leonardis. Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[410] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 521–528. MIT; 1998, 2003.

[411] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2013.

[412] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 212–219, 2005.

[413] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. Robust superpixel tracking. *IEEE Transactions on Image Processing*, 23(4):1639–1651, April 2014.

[414] Ming Yang, Ying Wu, and Gang Hua. Context-aware visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1195–1209, 2009.

[415] Ming-Hsuan Yang, Serge Belongie, and Boris Babenko. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1619–1632, 2010.

[416] Yang Yang, Jimei Yang, Junjie Yan, Shengcai Liao, Dong Yi, and Stan Z Li. Salient color names for person re-identification. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[417] Jian Yao and Jean-Marc Odobez. Multi-camera multi-person 3D space tracking with MCMC in surveillance scenarios. In *European Conference on Computer Vision, workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCV-M2SFA2)*, Marseille, France, October 2008.

[418] Dong Yi, Zhen Lei, and Stan Z Li. Towards pose robust face recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3545. IEEE, 2013.

[419] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Deep metric learning for person re-identification. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 34–39, 2014.

[420] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *Proceedings of the IEEE International Conference on International Conference on Pattern Recognition (ICPR)*, pages 34–39, 2014.

[421] Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256. Association for Computational Linguistics, 2011.

[422] Alper Yilmaz, Xin Li, and Mubarak Shah. Object contour tracking using level sets. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2004.

[423] Zhaozheng Yin, Fatih Porikli, and Robert T. Collins. Likelihood map fusion for visual object tracking. In *IEEE Workshop on Applications of Computer Vision*, pages 1–7, January 2008.

[424] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, 13(1):1–26, 2012.

[425] Jinjie You, Ancong Wu, Xiang Li, and Wei-Shi Zheng. Top-push video-based person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1353, 2016.

[426] Jun Yu, Xiaokang Yang, Fei Gao, and Dacheng Tao. Deep multimodal distance metric learning using click constraints for image ranking. *IEEE Transactions on Cybernetics*, 2016.

[427] Rui Yu, Zhichao Zhou, Song Bai, and Xiang Bai. Divide and fuse: A re-ranking approach for person re-identification. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[428] C. Zhang and Y. Rui. Robust visual tracking via pixel classification and integration. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 37–42, Hong Kong, September 2006.

[429] Honggang Zhang, Lorant Toth, Weihong Deng, Jun Guo, and Jie Yang. Monitoring visual focus of attention via local discriminant projection. In *Proceeding of the International Conference on Multimedia Information Retrieval*, 2008.

[430] Jianming Zhang, Shugao Ma, and Stan Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[431] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang. Fast visual tracking via dense spatio-temporal context learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 8693, pages 127–141, 2014.

[432] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block lbp representation. In *Advances in Biometrics*, pages 11–18. Springer, 2007.

[433] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[434] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1239–1248, 2016.

[435] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, November 2012.

[436] Ying Zhang, Baohua Li, Huchuan Lu, Atshushi Irie, and Xiang Ruan. Sample-specific svm learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1278–1287, 2016.

[437] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1085, 2017.

[438] Liming Zhao, Xi Li, Jingdong Wang, and Yueting Zhuang. Deeply-learned part-aligned representations for person re-identification. In *Proceedings of the IEEE International Conference on International Conference on Computer Vision (ICCV)*, 2017.

[439] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3586–3593. IEEE, 2013.

[440] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015.

[441] Lilei Zheng. *Triangular similarity metric learning : A siamese architecture approach*. Thesis, Université de Lyon, May 2016.

[442] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Associating groups of people. In *BMVC*, 2009.

[443] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 649–656. IEEE, 2011.

[444] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1):13, 2017.

[445] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on International Conference on Computer Vision (ICCV)*, 2017.

[446] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparse collaborative appearance model. *IEEE Transactions on Image Processing*, 23(5):2356–2368, 2014.

[447] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[448] Jianqing Zhu, Shengcai Liao, Zhen Lei, Dong Yi, and Stan Li. Pedestrian attribute classification in surveillance: Database and evaluation. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 331–338, 2013.

[449] Jianqing Zhu, Shengcai Liao, Zhen Lei, and Stan Z Li. Improve pedestrian attribute classification by weighted interactions from other attributes. In *Asian Conference on Computer Vision*, pages 545–557. Springer, 2014.

[450] Jianqing Zhu, Shengcai Liao, Dong Yi, Zhen Lei, and Stan Z Li. Multi-label cnn based pedestrian attribute learning for soft biometrics. In *Proceedings of the International Conference on Biometrics(ICB)*, pages 535–540. IEEE, 2015.

[451] Jianqing Zhu, Shengcai Liao, Zhen Lei, and Stan Z Li. Multi-label convolutional neural network based pedestrian attribute classification. *Image and Vision Computing*, 58:224–229, 2017.

[452] Bohan Zhuang, Huchuan Lu, Ziyang Xiao, and Dong Wang. Visual tracking via discriminative sparse similarity map. *IEEE Transactions on Image Processing*, 23(4):1872–1881, April 2014.