Thesis for the degree of Doctor of Philosophy

Riemannian Manifold-Based Modeling and Classification Methods for Video Activities with Applications to Assisted Living and Smart Home

Yixiao Yun

CHALMERS

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology

Gothenburg 2016

Riemannian Manifold-Based Modeling and Classification Methods for Video Activities with Applications to Assisted Living and Smart Home.
Yixiao Yun
ISBN 978-91-7597-421-7

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4102
ISSN 0346-718X

Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone: + 46 (0) 31 – 772 1000

This thesis has been prepared using LaTeX.

Printed by Chalmers Reproservice,
Gothenburg, Sweden 2016.

To my parents and Jielin

Abstract

This thesis mainly focuses on visual-information based daily activity classification, anomaly detection, and video tracking through using visual sensors. The main reasons for adopting visual-information based methods are due to: (i) vision plays a major role in recognition/classification of activities which is a fundamental issue in a human-centric system; (ii) visual sensor-based analysis may possibly offer high performance with minimum disturbance to individuals' daily lives.

Manifolds are employed for efficient modeling and low-dimensional representation of video activities, due to the following reasons: (a) the nonlinear nature of manifolds enables effective description of dynamic processes of human activities involving non-planar movement, which lie on a nonlinear manifold other than a vector space; (b) many video features of human activities may be effectively described by low-dimensional data points on the Riemannian manifold while still maintaining the important property such as topology and geometry; (c) the Riemannian geometry provides a way to measure the distances/dissimilarities between different activities on the nonlinear manifold, hence is a suitable tool for classification and tracking.

In this thesis, six different methods for visual analysis of human activities are introduced, including fall detection in video, activity classification in image and video, and video tracking using single camera and multiple cameras. Considering the contribution in theoretical aspects, the use of Riemannian manifolds was investigated for mathematical modeling of video activities, and new methods were developed for characterizing and distinguishing different activities. Experiments on real-world video/image datasets were conducted to evaluate the performance of each method. Results, comparisons, and evaluations showed that the methods achieved state-of-the-art performance. From the perspective of application, the methods have a wide range of potential applications such as assisted living, smart homes, eHealthcare, smart vehicles, office automation, safety systems and services, security systems, situation-aware human-computer interfaces, robot learning, etc.
Keywords: Activity classification, fall detection, video tracking, activities of daily living (ADL), assisted living, smart homes, Riemannian manifold

Acknowledgement

First and foremost, I would like to thank my main supervisor Prof. Irene Yu-Hua Gu, whose support and guidance made my thesis work possible. She has been actively involved in my work and has always been available to advice me. I am very grateful for her patience, motivation, enthusiasm, and immense knowledge that helped me grow as a better researcher. Besides, I would like to thank my co-supervisor Prof. Hamid Aghajan, for his encouragement and inspirations that widen my research from various perspectives.

I would also like to thank Prof. Mats Viberg and Prof. Tomas McKelvey for letting me join Signal Processing Group and conduct research of my interest in this fantastic work environment. I am especially grateful to all administrators and secretaries in S2 who have provided excellent support for my work in various ways. I would like to express my special thanks to Swedish Research Council (VR) for their financial support in part of this thesis work.

Sincere thanks go to my colleagues in Signal Processing Group. It has been a pleasure to work alongside with you guys. In particular, I am grateful to Keren Fu and Mohammad Alipoor for their help and the stimulating discussions. I would also like to give thanks to all my friends in Gothenburg, especially my Chinese friends, including those who had left, for making my life abroad more lovely and delightful.

Last but not least, I would like to thank my parents for all their love and encouragement. Most of all, I wish to express my deepest gratitude to my beloved wife Jielin. Your faithful support throughout my PhD study is so much appreciated. I love you.

Yixiao Yun
Gothenburg, June 2016

List of Publications

The thesis is based on the following papers:

Paper 1: Yixiao Yun, Irene Yu-Hua Gu, "Visual information-based activity recognition and fall detection for assisted living and ehealthcare," Chapter 15 in the book *Ambient Assisted Living and Enhanced Living Environments: Principles, Technologies and Control*, Elsevier, pp. 395–427, 2016.

Paper 2: Yixiao Yun, Irene Yu-Hua Gu, "Human fall detection in videos via boosting and fusing statistical features of appearance, shape and motion dynamics on Riemannian manifolds with applications to assisted living," *Computer Vision and Image Understanding (CVIU)*, vol. 148, pp. 111–122, 2016.

Paper 3: Yixiao Yun, Irene Yu-Hua Gu, "Part-based features and geodesic-induced kernel machine for human activity classification on Riemannian manifolds," Submitted to journal.

Paper 4: Yixiao Yun, Irene Yu-Hua Gu, "Exploiting tree-structured Riemannian manifolds for daily activity classification in video towards health care," *IEEE International conference on E-Health Networking, Application & Services (Healthcom)*, pp. 363–368, Munich, Germany, Sept. 14-17, 2016.

Paper 5: Yixiao Yun, Irene Yu-Hua Gu, "Time-dependent bag of words on manifolds for geodesic-based classification of video activities," Submitted to journal.

Paper 6: Yixiao Yun, Irene Yu-Hua Gu, Hamid Aghajan, "Riemannian manifold-based support vector machine for human activity classification in images," *IEEE International Conference on Image Processing (ICIP)*, pp. 3466–3469, Melbourne, Australia, Sept. 15-18, 2013.

Paper 7: Yixiao Yun, Keren Fu, Irene Yu-Hua Gu, Jie Yang, "Visual object tracking with online learning on Riemannian manifolds by one-class support vector machines," *IEEE International Conference on*

Image Processing (ICIP), pp. 1902–1906, Paris, France, Oct. 27-30, 2014.

Paper 8: Yixiao Yun, Irene Yu-Hua Gu, Hamid Aghajan, "Multi-view ML object tracking with online learning on Riemannian manifolds by combining geometric constraints," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), Special Issue on Computational and Smart Cameras*, vol. 3, no. 2, pp. 185–197, 2013.

Other publications:

9. Yixiao Yun, "Visual object tracking and classification using multiple sensor measurements," *Licentiate thesis*, Chalmers University of Technology, 2013.

10. Yixiao Yun, Irene Yu-Hua Gu, "Human fall detection in videos by fusing statistical features of shape and motion dynamics on Riemannian manifolds," *Neurocomputing*, vol. 207, pp. 726–734, 2016.

11. Yixiao Yun, Christopher Innocenti, Gustav Nero, Henrik Lindén, Irene Yu-Hua Gu, "Fall detection in RGB-D videos for elderly care," *IEEE International conference on E-Health Networking, Application & Services (Healthcom)*, 2015.

12. Yixiao Yun, Irene Yu-Hua Gu, "Human fall detection via shape analysis on Riemannian manifolds with applications to elderly care," *IEEE International Conference on Image Processing (ICIP)*, 2015.

13. Yixiao Yun, Keren Fu, Irene Yu-Hua Gu, Hamid Aghajan, Jie Yang, "Human activity recognition in images using SVMs and geodesics on smooth manifolds," *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2014.

14. Yixiao Yun, Irene Yu-Hua Gu, Mikhail Bolbat, Zulfiqar H. Khan, "Video-based detection and analysis of driver distraction and inattention," *IEEE International Conference on Signal Processing and Integrated Networks (SPIN)*, 2014.

15. Yixiao Yun, Mohamed Hashim Changrampadi, Irene Yu-Hua Gu, "Head pose classification by multi-class AdaBoost with fusion of RGB and depth images," *IEEE International Conference on Signal Processing and Integrated Networks (SPIN)*, 2014.

16. Yixiao Yun, Irene Yu-Hua Gu, Julien Provost, Knut Åkesson, "Multi-view hand tracking using epipolar geometry-based consistent labeling for an industrial application," *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2013.

17. **Yixiao Yun**, Irene Yu-Hua Gu, "Image classification by multi-class boosting of visual and infrared fusion with applications to object pose recognition," *Swedish Symposium on Image Analysis (SSBA)*, 2013.

18. **Yixiao Yun**, Irene Yu-Hua Gu, Hamid Aghajan, "Maximum-likelihood object tracking from multi-view video by combining homography and epipolar constraints," *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2012.

19. **Yixiao Yun**, Irene Yu-Hua Gu, "Multi-view face pose classification by boosting with weak hypothesis fusion using visual and infrared images," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

20. Durga Priya Kumar, **Yixiao Yun**, Irene Yu-Hua Gu, "Fall detection in RGB-D videos by combining shape and motion features," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

21. Keren Fu, Chen Gong, **Yixiao Yun**, Yijun Li, Irene Yu-Hua Gu, Jie Yang, Jingyi Yu, "Adaptive multi-level region merging for salient object detection," *British Machine Vision Conference (BMVC)*, 2014.

22. Keren Fu, Irene Yu-Hua Gu, **Yixiao Yun**, Chen Gong, Jie Yang, "Graph construction for salient object detection in videos," *International Conference on Pattern Recognition (ICPR)*, 2014.

23. Mohamed Hashim Changrampadi, **Yixiao Yun**, Irene Yu-Hua Gu, "Multi-class ada-boost classification of object poses through visual and infrared image information fusion," *International Conference on Pattern Recognition (ICPR)*, 2012.

24. Irene Yu-Hua Gu, Durga Priya Kumar, **Yixiao Yun**, "Privacy-preserving fall detection in healthcare using shape and motion features from low-resolution RGB-D videos," *International Conference on Image Analysis and Recognition (ICIAR)*, 2016.

25. Irene Yu-Hua Gu, Grzegorz Sowulewski, **Yixiao Yun**, Anders Flisberg, Magnus Thordstein, "3D limb movement tracking and analysis for neurological dysfunctions of neonates using multi-camera videos," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.

Acronyms

ADL:	Activities of Daily Living
BoVW:	Bag of Visual Words
BoW:	Bag of Words
DPM:	Deformable Part Models
DTW:	Dynamic Time Warping
ELM:	Extreme Learning Machine
GMM:	Gaussian Mixture Model
HMM:	Hidden Markov Model
HOF:	Histogram of Optical Flow
HOG:	Histogram of Oriented Gradients
IR:	Infrared
ML:	Maximum Likelihood
NN:	Nearest Neighbors
RGB:	Red, Green, Blue
RKHS:	Reproducing Kernel Hilbert Space
SPD:	Symmetric Positive Definite
SVM:	Support Vector Machine

Contents

Part I

Introductory chapters

Chapter 1

Introduction

With the rapid technological advancement of optical electronics and data storage over the past few decades, digital imaging sensors and devices have become ubiquitous, ranging from visual cameras, thermal/infrared (IR) cameras, near-infrared (NIR) cameras, to range/depth cameras and so on. This has led to the rise of computer vision that is considered as an interdisciplinary field related to artificial intelligence (AI), machine learning, signal processing and pattern recognition. Activity classification and video tracking have been two fundamental tasks in the field of computer vision. Activity classification generally aims to determine to which of several distinct and exclusive activity classes each input image or video belongs. Video tracking is concerned with estimation of location and shape of dynamic targets in the image plane throughout all the video frames. Such analyses of video activities have attracted a great deal of research interest in recent years, largely driven by the wide range of real-world applications, for example, assisted living, smart home, eHealthcare, smart vehicle, office automation, safety systems and services, security systems, situation-aware human-computer interfaces, robot learning, etc. Designing effective and robust methods for the analysis of video activities is far from being a simple mission, due to a variety of challenges and constraints. Commonly encountered difficulties include illumination variance, scale variance, viewpoint variance, appearance change, background clutter, occlusions, camera motion, and real time constraint. Despite the efforts and success made by previous methods, achieving consistently high performance in various applications remains an open issue.

Ambient assisted living is one of the most demanding applications that aims to offer intelligent services supporting people's daily lives. An ambient assisted living system creates human-centric smart environments that are sensitive, adaptive and responsive to human needs, habits, gestures and emotions. As shown in Fig 1, on creating a human-centric smart environ-

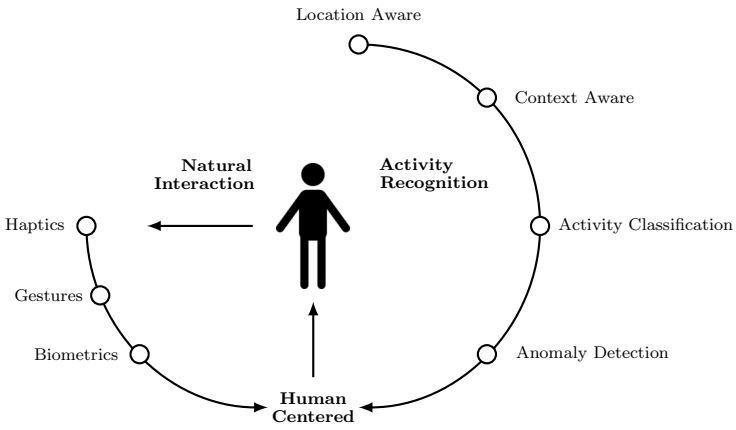


Figure 1.1: The key factors for ambient intelligence in assisted living.

ment for assisted living, three of the most fundamental issues are (a) to classify normal daily activities; (b) to detect abnormal activities; (c) to be aware of the person's location. Key functionalities of ambient assisted living include detecting anomalies like falls, robbery or fire at home, recognizing daily living patterns, and obtaining statistics of various daily activities over time. Among all activities, detecting falls is one of the basic topics attracting much attention, due to the associated risks [1], e.g. bone fracture, stroke, or even death. Triggering emergent help is desired, especially for persons who live alone.

Many different methods have been developed by exploiting different types of sensors, e.g., smartwatches [2], sound sensors, wearable motion devices (gyroscopes, speedometers, accelerometers), visual, range and infrared (IR) cameras. Sound sensors could be used for collecting sound related to sudden falls, while smartwatches and other wearable devices measure the motion and can be used for fall detection as well. However, they are not always feasible, e.g., one does not always wear the device especially during showering, and the false alarm could be high if sound detection is used alone without combining other sensor information. Devices with imaging sensors (e.g., RGB, depth, near or thermal IR) that offer real time analysis have drawn considerable attentions. In such cases, only analysis results, statistics and triggering information are stored without saving person's video information. Although some privacy concerns exist in video-based analytics, this issue can be mitigated by near real-time feature/information extraction from video followed discarding the original image data, or, by using

very low resolution on depth or IR data that does not provide personal details where only the shape of a person is visible.

This thesis mainly focuses on visual-information based daily activity recognition, anomaly detection, and video tracking through using visual sensors. The main reasons for adopting visual-information based methods are due to: (i) vision plays a major role in recognition/classification of activities which is a fundamental issue in a human-centric system; (ii) visual sensor-based analysis may possibly offer high performance with minimum disturbance to individuals' daily lives.

1.1 State of the Art: an Overview

1.1.1 Existing Work on Activity/Action Classification

For recognizing human activities from visual sensors, many different methods have been proposed, including both image-based and video-based methods. Some activity analyses only exploit *static cues* from still images or key image frames [3, 4], while others use the *dynamics* of entire video (e.g. falls) [5, 6]. Surveys of image/video-based methods can be found in [7–10].

There are many different ways to categorize the methods used for visual analysis of human activities. In this section, we follow a taxonomical breakdown based on representations. That is, we categorize the methods according to how video activities are represented or how the features are learned. We first describe global and local representations, and then review some state-of-the-art methods based on manifolds and deep learning.

Global Representations

In this category, video activities are represented by global features of human body motion, shape, and pose. Examples of global representation using motion cues include motion energy image (MEI), motion history image (MHI), and their variants. MEI and MHI were introduced by Bobick and Davis [11] with the idea to encode motion information by single images. MEI captures where the motion happens, and MHI shows how the motion proceeds. Based on the gradient of MHI, Tian *et al.* [12] proposed a method to filter out the moving and cluttered background by determining key motion regions in MHI using Harris interest point detector and detecting regions with inconsistent motions around the interest points. Moreover, effort were made to extend MEI and MHI to spatio-temporal volumes. Blank *et al.* [13] introduced the volumetric extension of MEI with the idea to represent video activities by 3-D shapes that are induced from spatio-temporal silhouettes. Weinland *et al.* [14] suggested to represent video activity using spatio-temporal volume

of MHIs. Shape is also an important feature for global representation, for example, Yilmaz and Shah [15] proposed a method to characterize actions using space-time volume (STV) which is built by temporally stacking object contours. Other types of global representation also exist. Sadanand and Corso [16] described actions by a large set of detectors acting as the bases of a high-dimensional action space. Shao *et al.* [17] used the Laplacian of 3-D Gaussian filters to construct the action space. Global representations can preserve the spatial and temporal structure of video activities to some extent, however, they may be too rigid to capture possible variations such as view point, appearance change, and occlusions. Besides, it is argued that silhouette-based representations are not capable of capturing fine details within the silhouette.

Local Representations

Methods in this category represent video activities by local features, starting from interest point detection. For example, Laptev [18] extended the 2-D Harris corner detector to 3-D Harris detector for detecting space-time interest points (STIPs) that has large spatial variations and non-constant motions. Willems *et al.* [19] also extended 2-D Hessian detector to a 3-D version that uses the second order derivatives for detecting spatio-temporal interest points. Based on detected interest points, local descriptors are formed. Some local descriptors are based on edge and motion, for example, Kläser *et al.* [20] spanned the Histogram of Oriented Gradients (HOG) descriptor [122] to the spatio-temporal domain (HoG3D). Laptev *et al.* [21] proposed to use the Histogram of Optical Flow (HOF) over local regions as a spatio-temporal descriptor. Kantorov and Laptev [22] used the motion fields of MPEG compression for obtaining HOF descriptors, which avoids computing optical flow fields for computational efficiency. Besides, several methods that describes local volumes based on Local Binary Pattern (LBP) [123] were proposed, such as [23] and [24]. It is worth mentioning that recently there is a growing trend towards extracting local features from trajectories instead of cuboids [25] [26] [27] [28]. Aggregation of extracted local descriptors also plays an important role. Commonly used aggregation methods include Bag of Visual Words (BOVW) [29], Fisher Vector (FV) [30] [31] [32], and Vector of Locally Aggregated Descriptor (VLAD) [33] [34].

Manifold-Based Methods

Using manifolds to characterize video activities has drawn increased interest in recent years. Human actions can be represented as a sequence of silhouettes, shapes, or contours on the manifold. For example, Veeraraghavan *et al.* [35] used a shape manifold to model human shapes, and extracted a sequence of shape changes from the tangent space. Shape sequences are

compared for gait and activity recognition, using a nonparametric method based on dynamic time warping (DTW), and a parametric method based on an AutoregressiveMoving-Average (ARMA) model. The space spanned by the parameters of the ARMA model was identified as an element on a Grassmann manifold. Turaga *et al.* [36] extended this ARMA representation of human activity to clustering, and conducted action recognition by treating the cascade of ARMA models as a regular expression grammar and applying grammatical inference from it. Further, Turaga *et al.* [37, 38] investigated statistical modeling Grassmann and Stiefel manifolds for human activity recognition. Abdelkader *et al.* [39] modeled contour shapes as points in a shape space of closed curves that is endowed with Riemannian geometry, and characterized trajectories on the shape space by a Markovian graphical model for action classification. Different from ARMA models that extract shapes and dynamics from the observability matrix, Lui *et al.* [40] [41] modeled the appearance, horizontal motion, and vertical motion on three factor manifolds each being a Grassmann manifold, based on a modified High Order Singular Value Decomposition (HOSVD). The geodesic distance on the product manifold formed by combining the three factor manifolds was used for action classification. Several methods conducted spatial and/or temporal alignment for video activity classification on the manifold. For example, Veeraraghavan *et al.* [42] studied the rate-invariant temporal alignment for video activities. Li and Chellappa [43] employed particle filters to optimize alignment parameters on Stiefel manifolds for spatio-temporal alignment. Another school of thought addresses the problem of activity classification using covariance descriptors that can be viewed as points on a Riemannian manifold of symmetric positive definite (SPD) matrices, e.g., [44, 45]. Last but not least, the approximate nearest neighbor search on Riemannian manifolds [46, 47] and Grassmann manifolds [47] was also applied to video activity analysis.

Deep Learning-Based Methods

Recent years have witnessed a significant advancement in various machine learning tasks using deep learning. Deep neural networks such as Convolutional Neural Networks (CNNs) [48] and Recurrent Neural Networks (RNNs) [49] have become common choices for image and video analysis, including the representation of video activities. The first category extends 2-D frame-based deep nets to 3-D domain, i.e., spatio-temporal domain. Ji *et al.* [50] introduced 3-D convolutional networks that uses 3-D kernels with filters extended along the time axis to extract features from both spatial and temporal dimensions. Other methods also investigated how to incorporate temporal information of video activities into convolutional networks, e.g., [51], [52]. Besides, some methods used recurrent networks to exploit the temporal information, such as [53] and [54]. Another school of thought

separates appearance from motion for activity recognition. For example, Simonyan and Zisserman [55] introduced the structure of two parallel deep convolutional networks, where the spatial stream network is fed by raw video frames, and the temporal stream network takes optical flow fields as input. This structure is extended by Wang *et al.* [56] through aggregating dense trajectories [28] of convolutional features from the two streams using the Fisher vector. Also, Wu *et al.* [57] extended the structure by adding a third stream using audio signal to the network. Deep generative models are another class of deep nets that can be used for video analysis in an unsupervised manner and predict the future of a video sequence. Examples of deep generative models are Dynencoder [94], Long-Short Term Memories (LSTM) autoencoder model [59], and adversarial models [60] [61]. Moreover, a deep model can be used to learn temporal coherency of video activities by feeding it with ordered and disordered sequences as positive and negative samples, e.g., [62], [63], and [64].

1.1.2 Existing Work on Video Tracking

Visual object occlusion is one of the most commonly encountered issues in visual object tracking. It occurs when other objects obstruct the line of sight between camera sensors and the object of interest (or, target). In the view of camera sensors, the object of interest is partially or fully occluded by other objects in images, and its appearance is more or less altered by the occluding objects. Tracking occluded objects becomes more difficult, which is likely to cause tracking drift. Hence, occlusion handling is required for mitigating the drift.

Single-Camera Tracking

Many existing approaches deal with occlusions in a single camera view. Wu *et al.* [65] employ a dynamic Bayesian network which accommodates an extra hidden process for occlusion to cope with occlusions. Huang and Essa [66] represent and estimate occlusion relationships between objects by using hidden variables of depth ordering of objects towards the camera. Pan and Hu [67] analyze occlusion by exploiting spatio-temporal context information and indicate occluded pixels by template matching. Amezquita *et al.* [68] detect occlusions by a probabilistic classifier and adapt motion prediction corresponding to the cases of entering occlusion, full occlusion and exiting occlusion. Papadakis and Bugeau [69] propose to track occluded objects by segmenting them into visible and occluded parts based on graph cuts. Chao *et al.* [70] recognize the start and end of occlusion frames through merging or splitting dynamic objects, and applies different template search approaches for data association between detected blobs and targets. Kwak *et al.* [71] divide target into regular grid cells and detects occlusion for each

cell using a classifier. Riemannian manifold-based trackers with a single camera are applied in [72–74], where dynamic learning is applied to mitigate the tracking drift. All these methods can handle occlusions to some extent, but become less feasible when objects undergo long-term full occlusions.

Multi-Camera Tracking

On the other hand, tracking using multiple cameras has drawn growing interest in recent years [75], largely driven by multiple view coverage that is advantageous in handling complex scenarios, including full occlusions.

Several object tracking schemes using multiple cameras with occlusion handling have been proposed recently [75]. One category of multi-view tracking methods handles the occlusion issue through using calibrated cameras, where the camera parameters (intrinsic/extrinsic) are known for projecting 3-D points into the image plane of each camera. For example, Mittal and Davis [76] detect 3-D points on an object by applying a region-based stereo algorithm, and analyze object occlusions by pixel-based classification of visible and occluded parts under Bayesian framework. Chen and Ji [77] model 3-D upper body using tree-structured probabilistic graphical model (PGM) to address self-occlusion, based on the likelihood of body part in each view. Harguess *et al.* [78] apply a 3-D cylinder head model for face tracking, where self-occlusion is handled by a weighted facial mask and full occlusion is detected by template matching. For outdoor scenarios where objects are located at large distances to cameras, it is difficult to accurately estimate 3-D point correspondences, where accurate camera calibration is non-trivial.

Another category of methods uses uncalibrated cameras, where the camera parameters are unknown. These methods exploit cross-view correspondences and transformations directly, without the attempt to compute camera parameters. For example, Kang *et al.* [79] map object trajectories across different views by registering multiple cameras via series of concatenated homography matrices (or, projective transformations). Wang *et al.* [80] and Fan *et al.* [81] each propose a spatio-temporal Bayesian filtering approach for multi-camera tracking, and use an affine transformation/homography to transform the image coordinates in difference camera views, respectively. Similarly, Zhou *et al.* [82] compute similarity transformation between different views in every previous frame for cross-view correspondence. However, the collinearity relation between points by assuming 2-D transformations may not hold for tracking objects that are not in the dominating ground plane. Instead, many methods in this category exploit underlying multi-view geometric constraints of the scene. Two constraints are often used, e.g., Chu *et al.* [83] use ground plane homography and Qu *et al.* [84] use

epipolar geometry. Sankaranarayanan and Chellappa [85] study the problem of combining estimates of ground location obtained from multiple cameras via an optimal fusion scheme based on planar homography, but the problem formulation is limited to tracking the ground point of object only. Du and Piater [86] estimate the foot position of an object in a top-view ground plane by first mapping principal axis (or, vertical axis) of the object in each view to that plane by homography, and then taking the intersection of these mapped axes. The drawback is that vertical axes can only be mapped to the common ground plane, thus the direct relation of object between different views is not established. Yue *et al.* [87] conduct two-view tracking by using a particle filter in each view, and detects occlusions by comparing pixel differences between tracked and template object. Kwolek [88] considers two-view tracking, where particle swarm optimization is used to track objects in each view, and occlusion is detected by computing the distance between the region covariance of tracked and template object. Both methods [87] [88] maintain the tracking in occluded views by mapping a transformation matrix of object bounding box from an un-occluded view using ground plane homography. However, applying homography solely is not sufficient for mapping object bounding box between views, as the bounding box is in the image plane rather than ground plane. Hence, additional geometric constraints should be added. To this end, Calderara *et al.* [157] combine the geometric constraints of planar homography, epipolar geometry and vertical vanishing point to map the vertical axis of object between views for cross-view consistent labeling.

1.2 Applications to Assisted Living, eHealthcare, and Smart Home

In this section, we show some examples of applying video activity analysis to the field of assisted living, smart home, and eHealthcare. These methods are primarily categorized into three different aspects: (a) daily living at home environments; (b) eHealthcare, hospital/nursing home monitoring, and rehabilitations; (c) falls and other abnormal activities.

Recognition of Activities of Daily Living (ADL) at Home Environments

The main purpose of such activity analysis is for life logging and the assessment of health conditions/functions. Context plays an important role for understanding ADL. One way of exploiting context is to use the location information, e.g., by dividing the living area into different functional regions and tracking the sequence of regions visited by the person for activity

analysis [90]. Alternatively, one could track the sequence of human-object interactions (e.g., enter the door, open fridge, put food into oven) as another type of contextual information to recognize an activity [91]. It is also beneficial to jointly use the location, speed, shape and motion of the person for activity recognition [92]. Further, a variety of methods can be found by applying different models, e.g. using bag-of-words (BoW) model based on HOG and HOF features [93], or extracting features from local body parts under deformable part models (DPM) [94].

Recognition of Activities for Hospital/Nursing Home Monitoring, eHealthcare, and Rehabilitations

A number of studies were conducted on visual analysis of behaviors of patients with stroke, Alzheimer's disease and other dysfunctions in hospitals or nursing homes, including medicine or food intake [95, 96], sit-to-stand motion [97] or the gait patterns [98, 99]. The activity analysis often involves recognizing human-object interactions [95], the interactions of human body parts [96], and human-to-human interactions [100]. Studies were also conducted on nursing activities from caregivers to the elderly (e.g. hygiene, feeding, giving medicine, taking vital signs), through identification of related objects or tools (e.g., paper towels, pillbox, diaper, plate, cup, sphygmomanometer) [101]. Different techniques can be applied for such analysis, e.g., hidden Markov model (HMM) [96, 100], dynamic time warping (DTW) [102], and a variety of classification methods (k-NN, LogitBoost, SVM) [103].

Detection of Falls and Other Abnormal Activities

Fall detection is a major issue in anomaly detection, due to its potential severe impact. Detection of falls is often realized through classification based on the one-against-all strategy. A variety of features related to falls were studied, including shape features based on extracted silhouettes [104, 105], curvature scale space (CSS) [164] and Riemannian manifolds [5], motion features based on optical flows [6], features based on target bounding box such as the aspect ratio [107, 108] and centroid position of the box [6, 108], and 3D modeling of human body [109–111]. Many different types of classifiers were employed for fall detection, such as SVM [5, 6, 107], AdaBoost [108], Gaussian mixture model (GMM) [104], Gustafson-Kessel (GK) clustering [105], extreme learning machine (ELM) [164] and decision trees [111]. Further, a range of camera types and settings were utilized, e.g., distributed cameras [107], IR [105], depth [6, 105, 111, 164], RGB cameras [5, 104, 105, 107–109], in day and night scenes [105].

Remark: There are many other applications beyond assisted living, smart home, and eHealthcare, for example, smart vehicle, office automation, safety systems and services, security systems, situation-aware human-computer interfaces, robot learning, etc. It is expected that the significance of activity recognition will be found even larger over time as it eventually merges into the background (ambient environment) together with powerful computational electronics.

1.3 Outline of the Thesis

This thesis consists of two parts. Part I is a general introduction to the field and puts the appended papers into context. Part II contains the appended papers.

The remainder of this introductory part (Part I) is organized as follows: Chapter 2 reviews several fundamental theories and methods upon which our methods are built. Chapter 3 summarizes the main work and contribution of this thesis, followed by Chapter 4 on conclusion and possible future work.

Chapter 2

Review of Related Theories and Methods

This chapter reviews several fundamental theories and methods upon which our methods are built.

2.1 Manifolds and Metrics

What is a manifold: Roughly speaking, a manifold can be considered as a set of low dimensional spaces embedded in a higher dimensional space. An intuitive example of manifold is the earth which is globally a sphere in 3D space but locally flat in 2D maps. Manifold-based methods are often employed for efficient low-dimensional representation of high-dimensional data meanwhile maintaining important properties of the data such as topology and geometry [112]. In case of nonlinear manifolds that are not in a single vector space, the Euclidean calculus and conventional operators do not apply. A Riemannian manifold is a smooth manifold that is differentiable [112], where a set of metrics can be defined to measure the distance, angle, and mean on the manifold. In the tangent spaces of Riemannian manifold points, linear operations can be performed. The geodesic is the shortest curve between two points on a manifold. The geodesic distance, the length of geodesic, is used to measure the distance between two manifold points.

Why we need manifold: Manifold is found useful in many vision tasks where (a) measured data naturally reside on nonlinear curved spaces, e.g., dynamic processes of video activities involving non-planar motion; (b) data representation requires low-dimensional and efficient description or dimensionality reduction, e.g., image, video, and other multidimensional signals;

(c) non-Euclidean metrics better capture the non-linear relationship between data elements. Hence, manifolds may be exploited for efficiently characterizing the dynamic process of human activities in videos.

Below, we give two examples of manifolds with the Riemannian geometry, namely the space of symmetric positive definite (SPD) matrices and the unit n-sphere.

2.1.1 Space of Symmetric Positive Definite Matrices

The space of $d \times d$ SPD matrices (Sym_+^d) is an open convex cone

$$Sym_+^d = \bigcap_{\mathbf{x} \in \mathbb{R}^d} \{\mathbf{P} \in Sym^d : \mathbf{x}^T \mathbf{P} \mathbf{x} > 0\}, \tag{2.1}$$

whose strict interior is a Riemannian manifold [112]. To compute the statistics on Sym_+^d, the affine-invariant metric [113] and the log-Euclidean metric [114] are commonly used. These two metrics are mathematically equivalent, however, numerical results can slightly differ. Log-Euclidean metric is usually computationally more efficient [114].

Two operators, the exponential map and the logarithm map, are defined over differentiable manifolds to switch between the manifold and tangent space at a given point [115].

Exponential map ($\mathcal{T}_\mathbf{P} \mapsto Sym_+^d$) is a function that maps a tangent vector $\mathbf{\Delta}$ (in the tangent space $\mathcal{T}_\mathbf{P}$ associated with a manifold point $\mathbf{P} \in Sym_+^d$) to its corresponding point $\mathbf{Q}$ on the manifold Sym_+^d (as shown in Fig. 1). Under the log-Euclidean metric [116], it is given by

$$\exp_\mathbf{P}(\mathbf{\Delta}) = \exp(\log(\mathbf{P}) + \mathbf{\Delta}) = \mathbf{Q}, \tag{2.2}$$

where $\exp(\cdot)$ is the matrix exponential [114], and $\log(\cdot)$ is the principal logarithm of a matrix which is defined as the inverse of the matrix exponential [114]. Under the affine invariant metric [113], it is given by

$$\exp_\mathbf{P}(\mathbf{\Delta}) = \mathbf{P}^{\frac{1}{2}} \exp(\mathbf{P}^{-\frac{1}{2}} \mathbf{\Delta} \mathbf{P}^{-\frac{1}{2}}) \mathbf{P}^{\frac{1}{2}} = \mathbf{Q}. \tag{2.3}$$

Logarithmic map ($Sym_+^d \mapsto \mathcal{T}_\mathbf{P}$) is a function that maps a manifold point $\mathbf{Q} \in Sym_+^d$ to its corresponding tangent vector $\mathbf{\Delta}$ in the tangent space $\mathcal{T}_\mathbf{P}$ associated with another manifold point $\mathbf{P} \in Sym_+^d$ (as shown in Fig. 1). Under the log-Euclidean metric [116], it is given by

$$\log_\mathbf{P}(\mathbf{Q}) = \log(\mathbf{Q}) - \log(\mathbf{P}) = \mathbf{\Delta}. \tag{2.4}$$

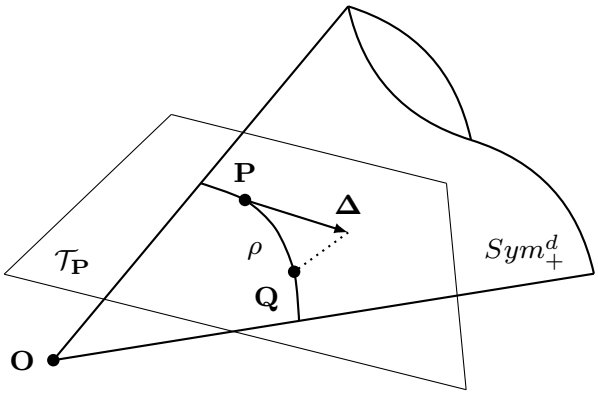


Figure 2.1: Example of Sym_+^d ($d = 2$) embedded in a 3D space $\mathbb{R}^3$. $\mathbf{O}$ is the origin. $\mathbf{P}$ and $\mathbf{Q}$ are manifold points, i.e., $\mathbf{P}, \mathbf{Q} \in Sym_+^d$. $\mathcal{T}_{\mathbf{P}}$ is the tangent space at $\mathbf{P}$. $\mathbf{\Delta} \in \mathcal{T}_{\mathbf{P}}$ is the tangent vector whose projected point on the manifold is $\mathbf{Q}$. The geodesic ρ is the shortest curve between $\mathbf{P}$ and $\mathbf{Q}$ on the manifold.

Under the affine invariant metric [113], it is given by

$$\log_{\mathbf{P}}(\mathbf{Q}) = \mathbf{P}^{\frac{1}{2}} \log(\mathbf{P}^{-\frac{1}{2}}\mathbf{Q}\mathbf{P}^{-\frac{1}{2}})\mathbf{P}^{\frac{1}{2}} = \mathbf{\Delta}. \tag{2.5}$$

Geodesic is the shortest curve ρ between two manifold points $\mathbf{P}$, $\mathbf{Q}$ on Sym_+^d. The geodesic distance is the length of ρ given by

$$d(\mathbf{P}, \mathbf{Q}) = \| \log_{\mathbf{P}}(\mathbf{Q}) \| = \| \log(\mathbf{Q}) - \log(\mathbf{P}) \|, \tag{2.6}$$

where (4) is defined under the log-Euclidean metric [116], and $\| \cdot \|$ is the Frobenius norm. Under the affine invariant metric [113], it is given by

$$d(\mathbf{P}, \mathbf{Q}) = \sqrt{\mathrm{tr}[\log^2(\mathbf{P}^{-\frac{1}{2}}\mathbf{Q}\mathbf{P}^{-\frac{1}{2}})]}. \tag{2.7}$$

Another alternative for computing the geodesic distance [117] is

$$d(\mathbf{P}, \mathbf{Q}) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(\mathbf{P}, \mathbf{Q})}, \tag{2.8}$$

where $\{\lambda_i(\mathbf{P}, \mathbf{Q})\}_{i=1}^n$ are the generalized eigenvalues of $\mathbf{P}$ and $\mathbf{Q}$, computed from $\lambda_i \mathbf{P}\mathbf{x}_i - \mathbf{Q}\mathbf{x}_i = 0$, $i = 1, \ldots, d$, and $\mathbf{x}_i \neq 0$ are the generalized eigenvectors.

Karcher mean (intrinsic mean), also known as the Fráchet or Riemannian mean, is the mean of a set of points computed directly on a Riemannian manifold. Given a set of manifold points $\{\mathbf{X}_i\}_{i=1}^N$, the Karcher mean $\mathbf{X}^*$ is defined as

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}, \mathbf{X}_i), \tag{2.9}$$

where $w_i \in \mathbb{R}$ is the weight for each point, and $d(\cdot,\cdot)$ is the geodesic distance defined in (4). The minimization problem can be solved by iteratively mapping from manifold to tangent spaces and vice versa until convergence [115, 116]:

$$\mathbf{X}_{j+1} = \exp_{\mathbf{X}_j}\left(\sum_{i=1}^{N} w_i \log_{\mathbf{X}_j}(\mathbf{X}_i) / \sum_{i=1}^{N} w_i\right), \tag{2.10}$$

where $\exp_{\cdot}(\cdot)$ and $\log_{\cdot}(\cdot)$ are the pair of exponential and logarithm mapping functions defined in (2) and (3) under log-Euclidean metric.

Extrinsic mean is computed in the tangent space. Given a set of manifold points $\{\mathbf{X}_i\}_{i=1}^N$, the extrinsic mean $\mathbf{X}^\dagger$ is defined as [115]

$$\mathbf{X}^\dagger = \exp_{\mathbf{I}}\left(\frac{1}{N}\sum_{i=1}^{N} \log_{\mathbf{I}}(\mathbf{X}_i)\right), \tag{2.11}$$

where $\mathbf{I} \in Sym_+^d$ is the identity matrix. In general, the process of estimating extrinsic means is computationally faster than its intrinsic counterpart.

The Riemannian geometry of Sym_+^d can be exploited when the extracted feature descriptors are covariance matrices, e.g., region covariance [117], due to the fact that SPD cone is exactly the set of non-singular covariance matrices. Since covariance matrices $\mathbf{C} \in Sym_+^d$, they may be viewed as points on a Riemannian manifold [118].

2.1.2 The Unit n-Sphere

The unit n-sphere, $\mathcal{S}^n$, is an n-dimensional sphere with a unit radius, centered at the origin of $(n+1)$-dimensional Euclidean space. It is mathematically defined by

$$\mathcal{S}^n = \{\mathbf{p} \in \mathbb{R}^{n+1} : \|\mathbf{p}\| = 1\}. \tag{2.12}$$

An example where $n = 2$ is illustrated in Fig. 2. It can be considered as the simplest Riemannian manifold after the Euclidean space [119]. The geodesic distance between two manifold points $\mathbf{p}$, $\mathbf{q}$ on $\mathcal{S}^n$ is the great-circle distance:

$$\rho(\mathbf{p}, \mathbf{q}) = \arccos(\mathbf{p}^T\mathbf{q}), \tag{2.13}$$

where $\arccos(\cdot) : [-1, 1] \rightarrow [0, \pi]$ is the inverse cosine function [120]. The great-circle distance between two manifold points is unique.

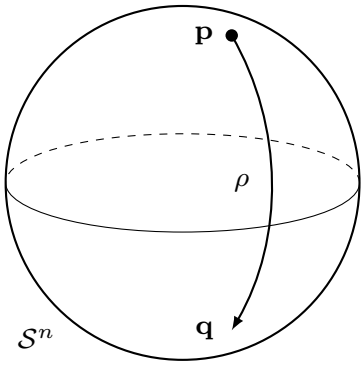


Figure 2.2: Example of an n-sphere $\mathcal{S}^n$ ($n = 2$) embedded in an $(n+1)$-D space $\mathbb{R}^{n+1}$. $\mathbf{p}$ and $\mathbf{q}$ are manifold points, i.e., $\mathbf{p}, \mathbf{q} \in \mathcal{S}^n$. The geodesic ρ is the shortest curve between $\mathbf{p}$ and $\mathbf{q}$ on the manifold.

The Riemannian geometry of unit n-sphere can be utilized when the extracted feature vectors are normalized by the ℓ_2 norm, e.g., SIFT [121], HOG [122], LBP [123]. The descriptors hence lie on a unit n-sphere $\mathcal{S}^n$, for some n.

2.2 Feature Descriptors

Several candidates of feature descriptors are reviewed in this section, including traditional hand-crafted features, and features that are automatically learned from deep learning.

In vision tasks, a feature is a measurable property (usually numeric) of an object or a scene being observed. The process of feature learning can be considered as a transformation of raw input data (usually complex, redundant, and highly variable, e.g., images, videos) to a representation that captures important properties of the original data, and that are mathematically and computationally more convenient to the task.

2.2.1 Histogram of Oriented Gradients (HOG)

Histogram of oriented gradients (HOG) is a feature descriptor for object detection and classification in images/videos [122]. The basic idea is that object shape can often be characterized by the distribution of intensity gradients through voting the dominant edge directions.

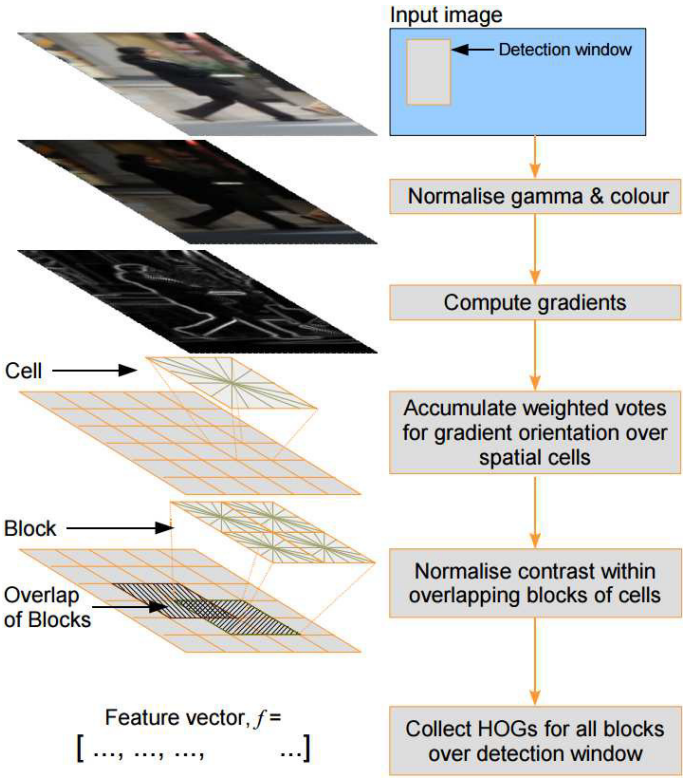


Figure 2.3: An overview of HOG feature extraction (The picture is taken from [124]).

Given an image I, gradient values in horizontal/vertical direction are computed by filtering the image with a 1-D derivative mask $[-1, 0, 1]$ along x-axis, or $[-1, 0, 1]^T$ along y-axis. For each pixel, given its gradient value I_x and I_y in both directions, the gradient magnitude and orientation can be computed by $\sqrt{I_x^2 + I_y^2}$ and $\arctan(I_y/I_x)$, respectively.

The image region is divided into non-overlapping cells of size $m \times m$. For each cell, a histogram is formed, containing l histogram bins that are evenly spread over 0 to 180 degrees if the gradient is unsigned (or, 0 to 360 degrees if signed). Every pixel in the cell casts a vote to one of the histogram bins according to its gradient orientation, weighted by its gradient magnitude. After grouping $r \times r$ adjacent cells into blocks with overlapping rate η, the grouped cell histograms are normalized block-wisely, e.g., by L_2 norm. Finally, the HOG feature descriptor is a vector concatenating the normalized histograms from all blocks. The process of extracting HOG features is illustrated in Fig. 2.3.

2.2.2 Optical Flow-Based Features

Optical flow is the pattern of apparent motion that is contained in a visual scene, as shown in Fig. 2.4. By estimating optical flows between video frames, the motion of objects can be characterized and quantified, e.g., the velocities of moving objects can be estimated. Optical flow algorithms are often based on three assumptions, namely intensity constancy, gradient constancy, and smoothness.

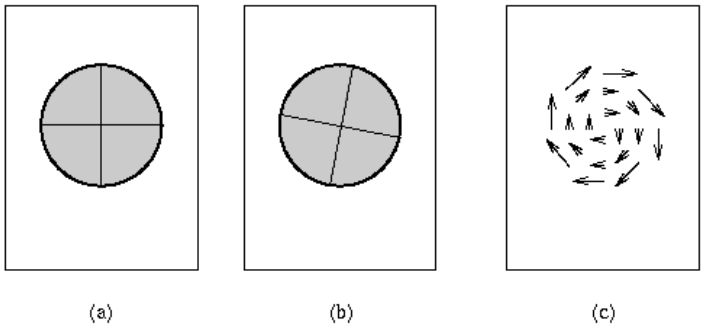


Figure 2.4: Example of optical flow (The picture is taken from [125]): (a) Time t_1; (b) Time t_2; (c) Optical flow.

Consider a pixel $I(x, y, t)$ in current frame, which moves by distance (dx, dy) in next frame taken after dt time. Due to intensity constancy, we have

$$I(x, y, t) = I(x + dx, y + dy, t + dt). \tag{2.14}$$

By taking Taylor series approximation of the right-hand side of (2.14), re-

moving common terms and dividing it by dt, we get

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0, \tag{2.15}$$

where $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ are the x and y components of the optical flow. This is an equation with two unknowns which cannot be solved directly (aperture problem). To estimate the optical flow, two techniques are commonly used, namely the Horn-Schunk method and Lucas-Kanade algorithm [126].

Given the optical flow between two consecutive video frames, histogram-based features of optical flow can be computed. The basic idea is similar to HOG that object motion can be represented by the distribution of optical flows as the votes for dominant directions of movement. Examples of such optical flow-based features are Histogram of Optical Flow (HOF) [127], Histogram of Oriented Optical Flow (HOOF) [128], and Histogram of Optical Flow Orientation and Magnitude (HOFM) [129].

2.2.3 Gabor Wavelet-Based Features

Gabor wavelets with properly selected frequencies and orientations can be assembled to form a filter bank that is similar to the model of human visual system, thus being found to be particularly appropriate for the use of pattern classification. Different representations of 2-D Gabor filters exist, and here we describe the one from [130]. In the spatial domain, a 2-D Gabor filter $g(x, y)$ is formed by modulating a Gaussian kernel function $w(x, y)$ with a sinusoidal plane wave $s(x, y)$ (shown in Figure 2.5):

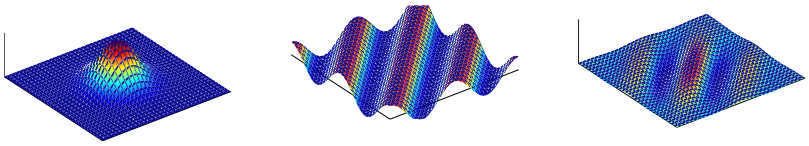
$$\begin{aligned} g(x,y) &= w(x,y) \cdot s(x,y) \\ &= \exp(-\frac{\hat{x}^2 + \hat{y}^2}{2\sigma^2}) \cdot \exp(j2\pi f\hat{x}) \end{aligned} \tag{2.16}$$

where

$$\begin{cases} \hat{x} = x\cos\theta + y\sin\theta \\ \hat{y} = -x\sin\theta + y\cos\theta \end{cases} \tag{2.17}$$

and θ is the rotation angle, the spread σ is the same for both x- and y-dimensions, and f is the spatial frequency.

If frequencies f and orientations θ are properly chosen, a bank of Gabor filters that covers the entire frequency domain can be obtained. Example of such a filter bank is shown in Figure 2.6 and 2.7. The Gabor wavelet-based representation of an image is obtained by convolving the image with each Gabor filter in the bank. Let $I(x, y)$ be the pixel intensity at coordinate

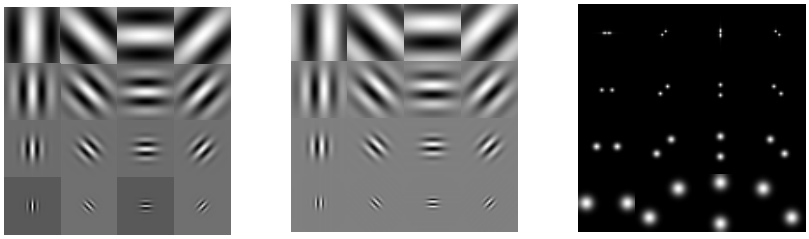


(a) A 2D Gaussian envelope.

(b) Real part of a sinusoidal plane wave.

(c) Real part of the corresponding 2-D Gabor filter.

Figure 2.5: A 2-D Gabor filter is obtained by modulating a Gaussian envelope with a sinusoidal plane wave.



(a) Real parts in spatial domain.

(b) Imaginary parts in spatial domain.

(c) Gabor wavelets in frequency domain.

Figure 2.6: A bank of Gabor filters. For each column in sub-figures from left to right: $\theta = 0$ (π), $\pi/4$ ($5\pi/4$), $\pi/2$ ($6\pi/4$), $3\pi/4$ ($7\pi/4$). For each row of from top to bottom, $\sigma = 16, 8, 4, 2$, where $f = 1/\sigma$.

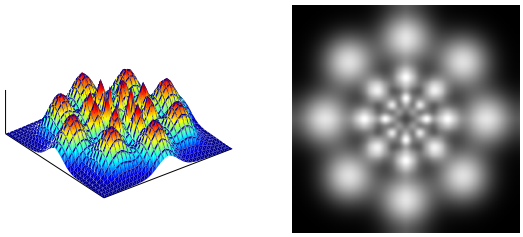


Figure 2.7: A bank of Gabor filters in frequency domain. For each layer, $\theta = 0$ (π), $\pi/4$ ($5\pi/4$), $\pi/2$ ($6\pi/4$), $3\pi/4$ ($7\pi/4$). For each circle from outer to inner layer, $\sigma = 2, 4, 8, 16$, where $f = 1/\sigma$.

(x, y) in a grayscale image, its convolution with a Gabor filter $g(x, y)$ is defined as $h(x, y) = I(x, y) * g(x, y)$. Since the filter responses are complex val-

ued, the real part $\Re\{h(x,y)\}$ or the magnitude $\sqrt{\Re^2\{h(x,y)\}+\Im^2\{h(x,y)\}}$ of each filter response is used, which is reshaped into a 1-D vector and normalized for enhanced robustness against illumination variance. The final feature vector based on Gabor wavelets is formed by concatenating all the 1-D vectors.

2.2.4 Covariance Descriptors

Covariance descriptors enable effective representation of various objects in spatial, temporal, or spatio-temporal domain. Instead of the joint representation of several different features through concatenation, one may compute the covariance matrix of these features and use it as the final feature descriptor. The covariance matrix provides a natural way of combining multiple features. The diagonal entries of the covariance matrix represent the variance of each feature, and the non-diagonal entries represent the correlations. The noise corrupting individual samples is effectively filtered out during the covariance computation.

Example: A typical application of covariance descriptor is region covariance [117]. Given a rectangular image region $\mathcal{R}$, let $\{\mathbf{f}_k\}_{k=1}^{|\mathcal{R}|}$ be l-dimensional pixel-wise feature vectors, where $|\mathcal{R}|$ is the total number of pixels in $\mathcal{R}$. The features can be, e.g., intensity, color, gradient, or filter responses. For instance, a feature vector can be formed as

$$\mathbf{f}_k = [x, y, r, g, b, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|, \sqrt{I_x^2 + I_y^2}, \arctan(\frac{I_y}{I_x})]^T \tag{2.18}$$

where (x, y) is the pixel coordinate, r, g, b are RGB color values of pixel, $|I_x|$, $|I_y|$, $|I_{xx}|$, $|I_{yy}|$ are magnitudes of the first and second derivatives along x, y directions, $\sqrt{I_x^2 + I_y^2}$ and $\arctan(\frac{I_y}{I_x})$ are the gradient magnitude and orientation, respectively. Another choice of feature vector can be Gabor wavelet-base features [72]

$$\mathbf{f}_k = [x, y, I, I_g^1, \cdots, I_g^M]^T \tag{2.19}$$

where (x, y) is the pixel coordinate, I is the image intensity, and I_g^m, $m = 1, \ldots, M$ are filtered images from 2-D Gabor filters of different orientations and frequencies [130]. The image region $\mathcal{R}$ is described by an $l \times l$ covariance matrix

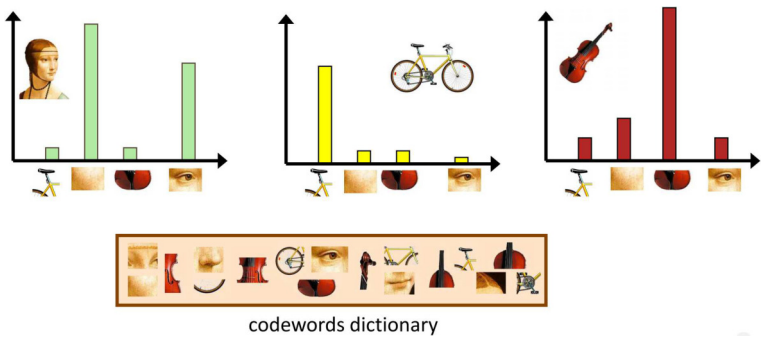
$$\mathbf{C}_\mathcal{R} = \frac{1}{|\mathcal{R}| - 1} \sum_{k=1}^{|\mathcal{R}|} (\mathbf{f}_k - \boldsymbol{\mu})(\mathbf{f}_k - \boldsymbol{\mu})^T \tag{2.20}$$

where $\boldsymbol{\mu}$ is the mean feature vector.

2.2.5 Bag of Words (BoW) Model

The bag-of-words (BoW) model is originally used in document classification, where each document is considered as a bag of words and is represented as a vector of occurrence counts of words (a histogram over the vocabularies). This model has also been applied to image classification [131], treating each image as a document (a bag of visual words). The BoW representation of an image is obtained by first clustering a set of selected local image descriptors such as SIFT (usually with k-means clustering) to generate a visual vocabulary (or, codebook), followed by extracting a histogram by assigning each descriptor to its closest visual word. The basic idea of BoW model is depicted in Fig. 2.8.

(a) Bag of visual words.



(b) Histogram over vocabularies (a vector of occurrence counts of visual words).

Figure 2.8: The basic idea of bag-of-words model in computer vision (The picture is taken from [132]).

As shown in Fig. 2.9, learning and recognition are important parts of the BoW model. Commonly used methods can be roughly divided into two categories, namely generative and discriminative models. Generative models estimate the probability of BoW features given a class, including Naïve

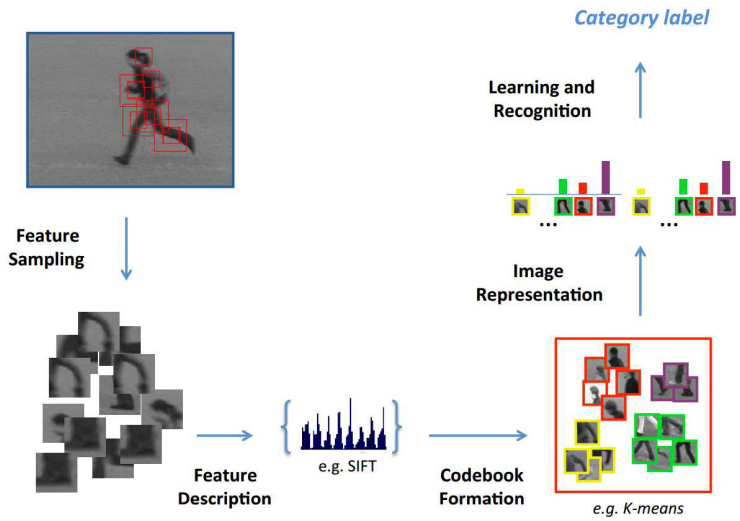


Figure 2.9: The pipeline of a typical bag-of-words model in computer vision (The picture is taken from [133]).

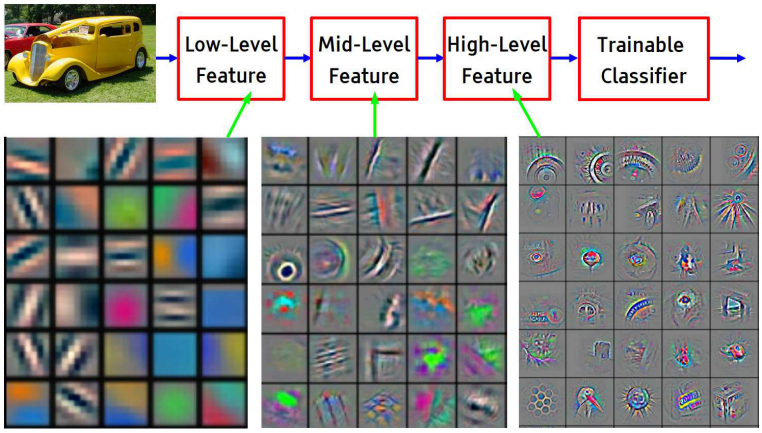
Bayes classifier, and hierarchical Bayesian models such as probabilistic latent semantic analysis (pLSA) and latent Dirichlet allocation (LDA). Discriminative models learn a decision rule (classifier) to assign BoW representation of images to different classes, including nearest-neighbor classifier, SVM, AdaBoost, and kernel methods such as pyramid match kernel. There also exist toolboxes for BoW, for example, a software package can be downloaded from [133].

Since the BoW model is an orderless representation that counts frequencies of visual words from a dictionary, efforts have been made to incorporate spatial information into the model. For example, one can compute BoW features from sub-windows of the entire image, or based on part-based models [134]. Also, spatial pyramid representation is an extension of BoW features that gives locally orderless representation at several levels of resolution [135].

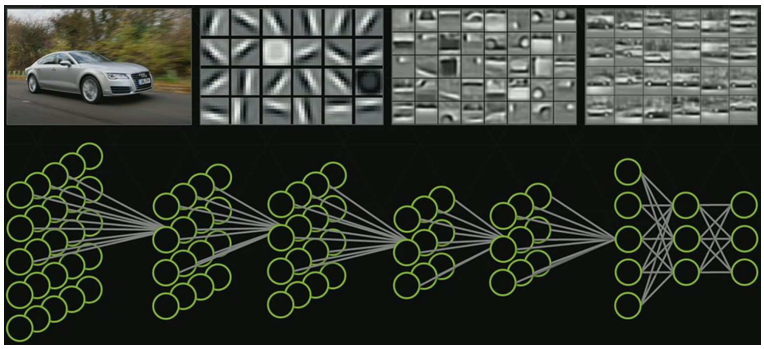
2.2.6 Automatically Learned Features from Deep Learning

Deep learning architectures (deep nets) attempt to learn multiple layers of representation of the input data with increased complexity and abstraction [136], which have recently demonstrated a remarkable success in various machine learning tasks using image, speech and video data. Different from

aforementioned feature extraction methods in previous subsections, deep nets learn multiple levels of representation, i.e., features, and completely automate the feature extraction from raw input data without requiring human intervention or expert knowledge.



(a) The picture is taken from [137].



(b) The picture is taken from [138].

Figure 2.10: The basic idea of deep learning that learns hierarchical representations.

In deep nets, the output of each intermediate layer is a different representation of the original input data, as depicted in Fig. 2.11. For each layer, the representation output from its previous layer is used as input. The raw data is fed to the input layer, and the final output layer produces the final low-dimensional feature or representation. Usually, the higher level of represen-

tations are more abstract and nonlinear, capturing structures that are not obvious from the input data [136]. Typical examples of such deep architectures for feature learning are Restricted Boltzmann machines (RBM) [139], Autoencoders [140] [141], Convolutional Neural Networks (CNNs) [142], and Recurrent Neural Networks (RNNs) [54] [143].

2.3 Classification Methods

The classification problem can be viewed as determining to which of several distinct and exclusive classes each newly observed data belongs.

In this section, two commonly used classification methods in machine learning are described, namely SVM and AdaBoost. In addition, distances and kernels for time series classification are discussed.

2.3.1 Support Vector Machines (SVMs)

Support vector machine (SVM) is a classification method, developed under the statistical learning theory, for supervised learning. A most commonly discussed form is SVM for binary classes [144]. Given a set of labeled feature vectors $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, an SVM aims to find a classifier that has the minimum generalization error on the test set. This is related to finding maximum-margin hyperplane, formulated by

$$\begin{aligned} &\min_{\mathbf{w},b} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \right), \\ \text{s.t.} \quad &y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \forall i \\ &\xi_i \geq 0, \quad \forall i \end{aligned} \tag{2.21}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, $\mathbf{w}$ is a weight vector, b is a bias, $C > 0$ is a regularization coefficient, and ξ_i is a slack variable. The Lagrange functional for the primal problem in (2.21) is

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \{y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \xi_i\} - \sum_{i=1}^{N} \mu_i \xi_i, \tag{2.22}$$

where $\alpha_i, \mu_i \geq 0$ are Lagrange multipliers. Then, the problem becomes

$$\min_{\mathbf{w},b} \max_{\alpha_i} L_P. \tag{2.23}$$

By substituting the Karush-Kuhn-Tucker (KKT) conditions [144] into (2.22), the dual Lagrangian is found to be

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \tag{2.24}$$

The dual problem is formulated as

$$\begin{aligned} &\max_{\alpha_i} L_D, \\ \text{s.t.} \quad &\begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.25}$$

which can be solved by applying quadratic programming.

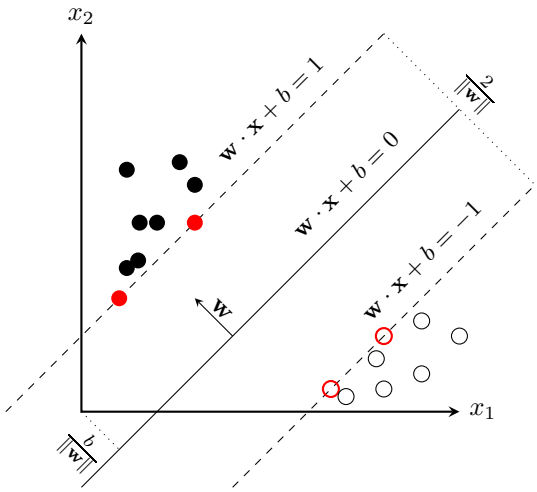


Figure 2.11: The illustration of SVM that separates the two classes with maximum-margin hyperplane. The support vectors are red colored.

For nonlinear separable classes, a mapping ($\phi : \mathbb{R}^d \mapsto \mathcal{H}$) is usually applied to map the feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ to a higher dimensional space. This produces a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with an inner product (kernel function) $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$. In this way, classes may become more close to linearly separable. For extension of a binary SVM to a multi-class SVM, *one-against-all* or *one-against-one* strategies are often adopted for simplicity [145].

2.3.2 Adaptive Boosting (AdaBoost)

Adaptive boosting (AdaBoost) is an ensemble learning method based on game theory, which is a special case of general algorithm for solving games through repeated plays [146]. It enables online sequential learning, and is solved by linear programming [147]. Comparing to SVM that is solved by quadratic programming, boosting is less computationally demanding [147]. The main idea of AdaBoost is to combine many weak learners to produce a powerful committee, where a set of weak learners are trained sequentially and weighted according to their accuracies. In each training iteration, training samples are also assigned weights, where wrongly classified samples gain weights and correctly classified ones lose weights. AdaBoost is originally intended only for binary problems [146].

AdaBoost is equivalent to forward stagewise additive modeling (FSAM). It sequentially adds new basis functions (weak learners) to the ensemble without adjusting the parameters and coefficients of those that have been already added. The optimization problem is based on the exponential loss function:

$$L(y, f(\mathbf{x})) = \exp(-yf(\mathbf{x})) \tag{2.26}$$

Given a set of labeled feature vectors $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, AdaBoost aims to solve:

$$(\alpha^{(m)}, T^{(m)}) = \arg\min_{\alpha,T} \sum_{i=1}^{N} \exp[-y_i(f^{(m-1)}(\mathbf{x}_i) + \alpha T(\mathbf{x}_i))], \tag{2.27}$$

where $T^{(m)}(\mathbf{x}) \in \{-1, 1\}$ is the weak learner to be added at m-th iteration, and $\alpha^{(m)}$ is the corresponding weight. This can be rewritten as

$$(\alpha^{(m)}, T^{(m)}) = \arg\min_{\alpha,T} \sum_{i=1}^{N} w_i^{(m)} \exp(-\alpha y_i T(\mathbf{x}_i)), \tag{2.28}$$

where

$$w_i^{(m)} = \exp(-y_i f^{(m-1)}(\mathbf{x}_i)). \tag{2.29}$$

Since $w_i^{(m)}$ is independent on α and $T(\mathbf{x}_i)$, it can be regarded as a weight factor that is applied to each training sample. This weight depends on $f^{(m-1)}(\mathbf{x}_i)$, so it changes during each iteration. It is observed that

$$y_i T(\mathbf{x}_i) = \begin{cases} +1, & \text{if } y_i = T(\mathbf{x}_i); \\ -1, & \text{if } y_i \neq T(\mathbf{x}_i). \end{cases} \tag{2.30}$$

Hence, the criterion in (2.28) can be expressed as

$$e^{-\alpha} \sum_{y_i = T(\mathbf{x}_i)} w_i^{(m)} + e^{\alpha} \sum_{y_i \neq T(\mathbf{x}_i)} w_i^{(m)}, \tag{2.31}$$

which can be further rewritten as

$$(e^{\alpha} - e^{-\alpha}) \sum_{i=1}^{N} w_i^{(m)} \mathbb{I}(y_i \neq T(\mathbf{x}_i)) + e^{-\alpha} \sum_{i=1}^{N} w_i^{(m)}. \tag{2.32}$$

Apply gradient descent method to (2.32) and solve for α, by taking partial derivative respect to α and setting the resulting equation to 0, we get

$$\alpha^{(m)} = \frac{1}{2} \log \frac{1 - \epsilon^{(m)}}{\epsilon^{(m)}}, \tag{2.33}$$

where $\epsilon^{(m)}$ is the minimized weighted error rate:

$$\epsilon^{(m)} = \frac{\sum_{i=1}^{N} w_i^{(m)} \mathbb{I}[y_i \neq T^{(m)}(\mathbf{x}_i)]}{\sum_{i=1}^{N} w_i^{(m)}}, \tag{2.34}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Then, the ensemble is updated by

$$f^{(m)}(\mathbf{x}) = f^{(m-1)}(\mathbf{x}) + \alpha^{(m)} T^{(m)}(\mathbf{x}). \tag{2.35}$$

The sample weights for the next iteration are updated by

$$w_i^{(m+1)} = \exp(-y_i f^{(m)}(\mathbf{x}_i)) = w_i^{(m)} \cdot e^{-\alpha^{(m)} y_i T^{(m)}(\mathbf{x}_i)}. \tag{2.36}$$

Considering the fact that

$$-y_i T^{(m)}(\mathbf{x}_i) = 2 \cdot \mathbb{I}(y_i \neq T(\mathbf{x}_i)) - 1, \tag{2.37}$$

the updating scheme of sample weights becomes

$$w_i^{(m+1)} = w_i^{(m)} \cdot e^{\beta^{(m)} \mathbb{I}(y_i \neq T(\mathbf{x}_i))} \cdot e^{-\alpha(m)} \tag{2.38}$$

where $\beta^{(m)} = 2\alpha(m)$. The multiplication factor $e^{-\alpha(m)}$ is applied to all weights so it can be ignored. The conventional AdaBoost algorithm is summarized in Table 2.1.

2.3.3 Distances and Kernels for Time Series

A time series is an ordered finite set (a sequence) of data points, typically consisting of measurements observed successively over a time interval. Some commonly used distance measures for time series classification include dynamic time warping (DTW) [149], edit distance with real penalty (ERP) [150], time warp edit distance (TWED) [151]. Based on these distances, kernel functions can be constructed to measure the similarity between time series. Although these kernel functions perform relatively well, they are not strictly positive definite [152]. Positive definiteness is a preferable property for kernel functions ensuring that the optimization problem is convex and the solution is unique [153]. To this end, some positive definite kernels for time series classification have been suggested, e.g., global alignment (GA) kernels [154], recursive edit distance kernels (REDK) [152], which are shown to perform better than indefinite kernels in general.

Table 2.1: Summary of AdaBoost algorithm [147].

1. Initialize sample weight for each training sample, $w_i = 1/N$, $i = 1, 2, \cdots, N$, and the total number of iterations M.
2. For $m = 1$ to M:
   (a) Fit a weak learner $T^{(m)}(\mathbf{x})$ to the training dataset using weights w_i.
   (b) Compute weighted training error rate for the weak learner:
   $$\epsilon^{(m)} = \sum_{i=1}^{N} w_i \mathbb{I}\left(y_i \neq T^{(m)}(\mathbf{x}_i)\right) / \sum_{i=1}^{N} w_i.$$
   (c) If $\epsilon^{(m)} \leq 0$ or $\epsilon^{(m)} \geq 0.5$, then abort loop.
   (d) Compute ensemble weight for the weak learner:
   $$\beta^{(m)} = \log \frac{1 - \epsilon^{(m)}}{\epsilon^{(m)}}.$$
   (e) Update sample weight for each training sample:
   $$w_i \leftarrow w_i \cdot \exp\left(\beta^{(m)} \cdot \mathbb{I}\left(y_i \neq T^{(m)}(\mathbf{x}_i)\right)\right),$$
   for $i = 1, 2, \cdots, N$.
   (f) Re-normalize the distribution of sample weights: $w_i \leftarrow w_i / \sum_{i=1}^{N} w_i$.
3. Output class predictions:
   $$C(\mathbf{x}) = \arg \max_{k \in \{-1, +1\}} \sum_{m=1}^{M} \beta^{(m)} \cdot \mathbb{I}\left(T^{(m)}(\mathbf{x}) = k\right)$$

2.4 Tracking Methods

2.4.1 Sequential Bayesian Estimation

The aim of sequential Bayesian estimation is to estimate the posterior pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ of state vector $\mathbf{x}_t$, given all observations $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \cdots, \mathbf{y}_t\}$ up to time t [155]. Two common criteria for estimating state $\mathbf{x}_t$ are:

- Minimum mean square error (MMSE):
$$\hat{\mathbf{x}}_t^{\text{MMSE}} = \arg\min_{\hat{\mathbf{x}}} \mathbb{E}[\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 | \mathbf{y}_{1:t}] = \mathbb{E}[\mathbf{x}_t | \mathbf{y}_{1:t}]. \tag{2.39}$$

- Maximum a posteriori (MAP):

$$\hat{\mathbf{x}}_t^{\text{MAP}} = \arg\max_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}}_t|\mathbf{y}_{1:t}). \tag{2.40}$$

Based on Bayes theorem, the law of total probability and Markov assumption, the posterior pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be rewritten as

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_{1:t})} \\ &= \frac{p(\mathbf{y}_t,\ \mathbf{y}_{1:t-1}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_t,\ \mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\ \mathbf{x}_t)p(\mathbf{y}_{1:t-1}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\ \mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})p(\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})p(\mathbf{x}_t)} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}. \end{aligned} \tag{2.41}$$

The second term in the numerator of (2.41) can be further expanded by marginalizing over the previous state $\mathbf{x}_{t-1}$:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) &= \int p(\mathbf{x}_t,\ \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1},\ \mathbf{y}_{1:t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}. \end{aligned} \tag{2.42}$$

The denominator of (2.41) is the normalization constant

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) &= \int p(\mathbf{y}_t,\ \mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t \\ &= \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t. \end{aligned} \tag{2.43}$$

Combining (2.41), (2.42) and (2.43) yields

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}, \tag{2.44}$$

which is the recursive formula for Bayesian estimation. As shown in (2.44), the posterior density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is characterized by three terms:

- The *likelihood* $p(\mathbf{y}_t|\mathbf{x}_t)$.

- The *priori* $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$.
- The *state transition probability* $p(\mathbf{x}_t|\mathbf{x}_{t-1})$.

Hence, posterior pdf can be calculated sequentially, given (i) the prior pdf $p(\mathbf{x}_0)$; (ii) the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$; (iii) the observation model $p(\mathbf{y}_t|\mathbf{x}_t)$.

2.4.2 Particle Filters

Based on Monte Carlo sampling approximation, *particle filter* estimates the posterior pdf by a weighted sum of $N \gg 1$ independent and identically distributed (i.i.d.) samples drawn from the posterior space

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} \delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(i)}), \tag{2.45}$$

where $\omega_t^{(i)}$ are the importance weights that sum up to 1.

It is practically not feasible to sample from the true posterior pdf. Instead, a *proposal distribution* $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ is used, and the weights are defined as

$$\omega_t^{(i)} = \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}. \tag{2.46}$$

For recursive update of the weights, the proposal distribution is supposed to have the following factorized form:

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}). \tag{2.47}$$

Similar to the derivation steps in (2.41), the posteriori $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ can be factorized as

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) \frac{p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \tag{2.48}$$

Plugging (2.47) and (2.48) into (2.46) yields

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}. \tag{2.49}$$

Based on *Markov assumption*, (2.49) is modified to

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}. \tag{2.50}$$

Using (2.50), expression to approximate the posterior pdf can be written as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}).$$

There are many different types of particle filters, we only briefly review sequential importance sampling (SIS) here. For SIS, it is commonly assumed that the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the state transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, i.e., $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Hence, particle weights are updated by

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}), \tag{2.51}$$

followed by weight normalization. To avoid the degeneracy phenomenon, re-sampling is performed according to the criterion based on *effective sample size* N_{eff} [155], when its estimate $\hat{N}_{eff}$ is found below a threshold N_T:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_t^{(i)})^2} < N_T, \tag{2.52}$$

where N_T can be either a predefined value (say $N/2$ or $N/3$) or the median of the weights, and N is the total number of particles. After re-sampling, (2.51) can be further simplified as

$$\omega_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)}). \tag{2.53}$$

The pseudo code for SIS with re-sampling is summarized in Table 2.2, and the process is visualized in Fig. 2.12.

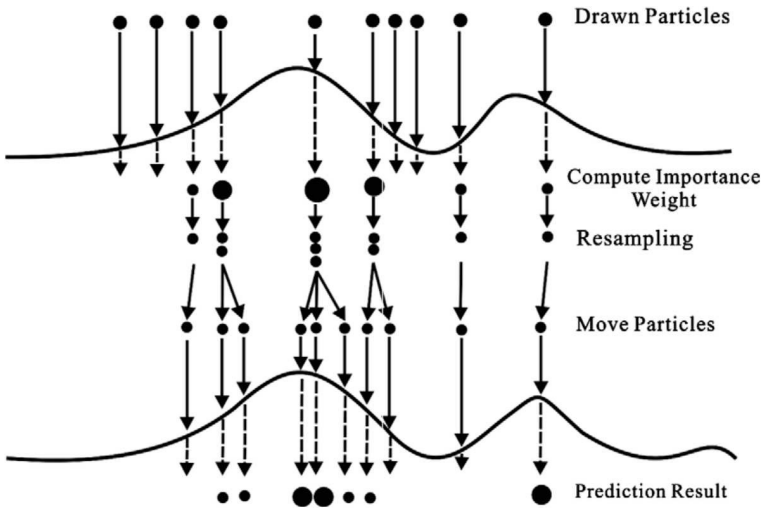


Figure 2.12: Visualization of sequential importance sampling with re-sampling (The picture is taken from [156]).

Table 2.2: Pseudo code for SIS with re-sampling [155].

1. **Input**: number of particles N, number of time steps T.
2. **Initialization** ($t = 0$): initial true state $\mathbf{x}_0$; for $i = 1, \cdots, N$, generate particles $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, with equal weights $\omega_0^{(i)} = 1/N$.
3. **For** time steps $t = 1,\ 2,\ \cdots,\ T$, **Do**

(a) Importance sampling: for $i = 1, \cdots, N$, generate particles $\hat{\mathbf{x}}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$.

(b) Weight update: calculate particle weights $\omega_t^{(i)}$ according to (2.51).

(c) Weight normalization: normalize the weights $\tilde{\omega}_t^{(i)} = \omega_t^{(i)} / \sum_{j=1}^{N} \omega_t^{(j)}$.

(d) Re-sampling **only if** (2.52): generate new particle set $\{\mathbf{x}_t^{(j)}\}_{j=1}^{N}$ by re-sampling with replacement from the set $\{\hat{\mathbf{x}}_t^{(i)}\}_{i=1}^{N}$ according to the normalized weights $\tilde{\omega}_t^{(i)}$, s.t. $P(\mathbf{x}_t^{(j)} = \hat{\mathbf{x}}_t^{(i)}) = \tilde{\omega}_t^{(i)}$, then reset the weights $\tilde{\omega}_t^{(i)} = 1/N$.

4. **End** $\{t\}$

2.5 Multiple View Geometry for Vision Tasks

2.5.1 Camera Calibration

Camera calibration is the process of approximating a camera with a model (e.g., pinhole model, as shown in Fig. 2.13) and estimating its model parameters using 2-D images of the 3-D scene. The model parameters can be used to correct for lens distortion, measure the size of an object in world coordinates, or determine the location of the camera in the scene. For each camera, the model parameters are represented in a 3×4 matrix called the camera matrix, or projection matrix $\mathbf{P}$, which maps a 3-D (homogeneous) point position $\mathbf{X} = [x, y, z, 1]^T$ in world coordinates to a 2-D point position $\mathbf{x} = [u, v, 1]^T$ in pixel coordinates (in image plane). Mathematically, this projective mapping is denoted as $\mathbf{x} = \mathbf{PX} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}$, where $\mathbf{K}$ is a 3×3 upper triangular matrix containing *intrinsic* parameters (focal length, optical center, etc.), and the remaining are *extrinsic* parameters $\mathbf{R}$ as a 3×3 rotation matrix and $\mathbf{t}$ as a 3×1 translation vector [158].

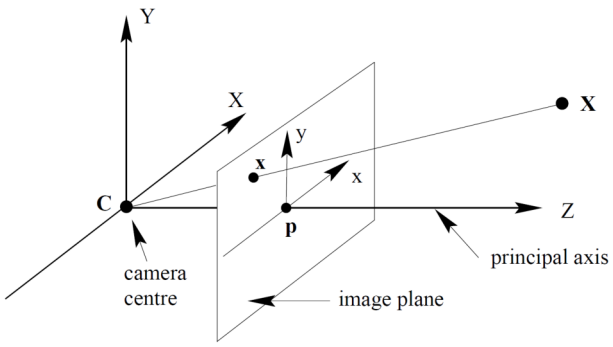


Figure 2.13: Pinhole camera geometry (Pictures are taken from [158]). **C** is the camera center (optical center), and **p** the principal point. The camera center is placed at the origin. The image plane is placed in front of the camera center.

To estimate the camera parameters, one needs to have 3-D world points and their corresponding 2-D image points. These correspondences can be obtained using multiple images of a calibration pattern, such as a checkerboard [159]. Using the correspondences, the camera parameters in the camera matrix $\mathbf{P}$ can be solved. For example, for each camera, given a set of images of a checkerboard viewed from different angles (sizes of the checkerboard and its square grids are known), followed by detecting corner points and origin in the images, all intrinsic and extrinsic parameters can be es-

timated (e.g., using calibration toolboxes [160]). Intrinsic parameters only have to be estimated once. Extrinsic parameters have to be estimated for each set of measurement separately, in case of changes in camera's relative position in the scene.

Remark: One category of multi-view tracking methods handles occlusions through using calibrated cameras, where the camera parameters (intrinsic/extrinsic) are known for projecting 3-D points into the image plane of each camera. For outdoor scenarios where objects are located at large distances to cameras, it is difficult to accurately estimate 3-D point correspondences, where accurate camera calibration is non-trivial. Another category of methods uses uncalibrated cameras, where the camera parameters are unknown. These methods exploit cross-view correspondences and transformations directly, without the attempt to compute camera parameters. Some useful geometric constraints and an example of using uncalibrated cameras for cross-view matching are described in subsequent subsections.

2.5.2 Planar Homography

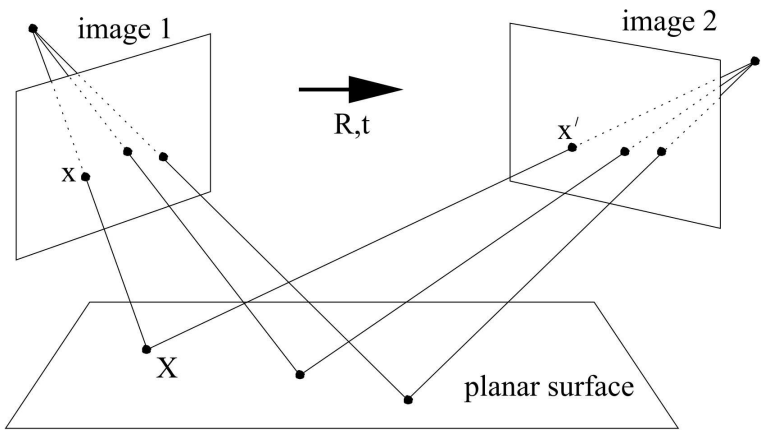


Figure 2.14: Illustration of planar homography (The picture is taken from [158]).

When a planar object is imaged from two views, the two images are related by a unique homography. The planar homography (2-D projective transformation) is a non-singular linear relationship between points on planes. Images of points on a plane in one view are related to corresponding image points in another view by this planar homography using a homoge-

neous representation. Is is a projective transformation since it only depends on the intersection of planes with lines. The homography can be used to map points from one view to the other if the points are on a common plane, as shown in Fig. 2.14.

2.5.3 Epipolar Geometry

Consider two different camera views of a scene $\mathcal{S}$ as shown in Fig.2.15, where $\mathbf{c}^i$ and $\mathbf{c}^j$ are their optical centers, respectively. Given a 3-D point $\mathbf{X} \in \mathcal{S}$, if $\mathbf{x}$ is the image of $\mathbf{X}$ in the image plane $\mathcal{I}^i$ of i-th view, then its corresponding point $\mathbf{x}'$ in the image plane $\mathcal{I}^j$ of j-th view is constrained to lie on a line $\ell' \in \mathcal{I}^j$ (the *epipolar line*) associated with $\mathbf{x}$. The epipolar line ℓ' is the intersection of $\mathcal{I}^j$ and the plane $\mathbf{\Pi}$ (the *epipolar plane*), which is defined by $\mathbf{x}$, $\mathbf{c}^i$ and $\mathbf{c}^j$.

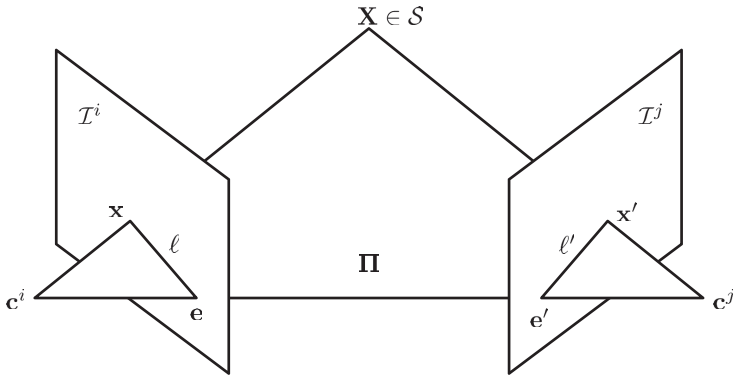


Figure 2.15: Illustration of 2-view epipolar geometry (The picture is reproduced from [158]). Conjugate epipolar lines (ℓ, ℓ') are generated by intersecting any plane $\mathbf{\Pi}$ containing the baseline $(\mathbf{c}^i \times \mathbf{c}^j)$ with the pair of image planes $(\mathcal{I}^i, \mathcal{I}^j)$, where $\times$ is the homogeneous cross product operation. The epipoles $(\mathbf{e}, \mathbf{e}')$ are obtained by intersecting $(\mathbf{c}^i \times \mathbf{c}^j)$ with $(\mathcal{I}^i, \mathcal{I}^j)$.

Mathematically, this relation is expressed by the *fundamental matrix* $\mathbf{F}$ satisfying $\mathbf{x}'\mathbf{F}\mathbf{x} = 0$, where the epipolar line is defined as $\ell' = \mathbf{F}\mathbf{x}$ [158]. Thus, the epipolar geometry constrains the corresponding points that lie on the conjugate pairs of epipolar lines, such that the match to a point $\mathbf{x} \in \ell$ must lie on ℓ' and vice versa.

2.5.4 Vertical Vanishing Point

It is observed that, in perspective projection, the image of an object that stretches off to infinity can have finite extent. For example, parallel lines in real world such as railway lines appear to converge in a camera view. The intersection in the image is the vanishing point for the direction of the railway. This phenomenon is illustrated in Fig. 2.16.

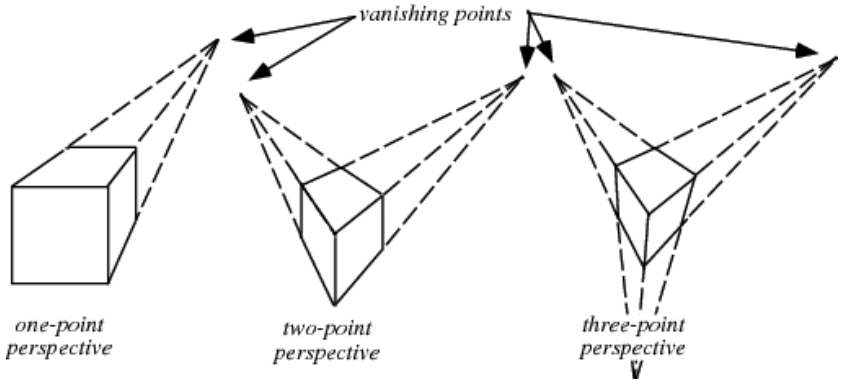


Figure 2.16: Illustration of vanishing points in different perspective views (The picture is taken from [161]).

Geometrically, the vanishing point of a line is obtained by intersecting the image plane with a ray that is parallel to a line in the real world and passing through the camera center [158]. Therefore, a vanishing point depends only on the direction of a line, not on its position. As a result, a set of parallel lines in the real world have a common vanishing point. As shown in Fig. 2.16, a three-point perspective view gives 2 horizontal vanishing points and 1 vertical vanishing point.

2.5.5 Cross-View Warping of Vertical Axis by Combining Geometric Constraints

To establish the relation of object between different views, one way is through exploiting the correspondences of object's vertical axes. The vertical axis of an object is the line segment connecting its top and ground points (see the dotted line segment in Fig. 2.17). Under the assumption that objects move or stand uprightly on a planar ground, which usually holds for outdoor scenes, the constraints of planar homography, epipolar

geometry and vertical vanishing point are combined to warp the vertical axis of tracked object between views [157].

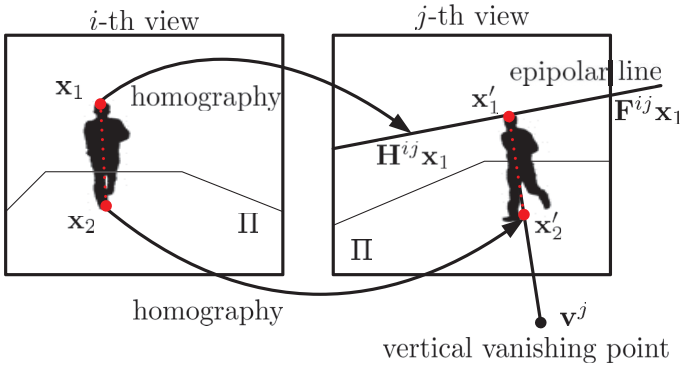


Figure 2.17: Warping vertical axis of a tracked object from i-th view to j-th view by combining the constraints of planar homography, epipolar geometry and vertical vanishing point (The picture is reproduced from [157]).

Let 2-D homogeneous points $\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1$ and $\mathbf{x}_2 \leftrightarrow \mathbf{x}'_2$ denote the corresponding top and ground points of object between i-th and j-th views. Given the homography $\mathbf{H}^{ij}$ induced by the plane Π from the i-th view to the j-th view, the correspondence of object ground position is related by $\mathbf{x}'_2 = \mathbf{H}^{ij}\mathbf{x}_2$. However, the top point $\mathbf{x}_1$ is off the plane Π, $\mathbf{x}'_1 \neq \mathbf{H}^{ij}\mathbf{x}_1$ (see Fig. 2.17). Homography is not sufficient for warping the vertical axis of object, additional geometric constraints should be added.

Given $\mathbf{x}_1$ in the i-th view, its corresponding point in the j-th view $\mathbf{x}'_1$ lies on the projection of the preimage of $\mathbf{x}_1$ onto the j-th view. This relation is expressed by using the fundamental matrix $\mathbf{F}^{ij}$ satisfying $\mathbf{x}'^{\top}_1\mathbf{F}^{ij}\mathbf{x}_1 = 0$. Since the preimage of $\mathbf{x}_1$ is a line, the projection of this line onto the j-th view gives the line $L(\mathbf{x}_1) = \mathbf{F}^{ij}\mathbf{x}_1$, which is the epipolar line associated with $\mathbf{x}_1$ (see Fig. 2.17). Thus, the epipolar geometry constrains the corresponding points that lie on the conjugate pairs of epipolar lines.

To obtain the warped axis inclination, the vertical vanishing point $\mathbf{v}^j$ of j-th view is used. As depicted in Fig. 2.17, the warped axis lies on a straight line passing through $\mathbf{v}^j$ and $\mathbf{x}'_2$. The top point $\mathbf{x}'_1$ is obtained as the intersection between the epipolar line and the straight line of the

axis, $\mathbf{x}'_1 = (\mathbf{F}^{ij}\mathbf{x}_1) \times (\mathbf{v}^j \times \mathbf{x}'_2)$, where $\times$ is the homogeneous cross product operation [158]. Using the same procedure, the vertical axis of tracked object in the j-th view may be warped to the i-th view.

Chapter 3

Contributions of this Thesis Work

In this section, we first summarize the main work of this thesis. We then describe each method and its main contributions in detail.

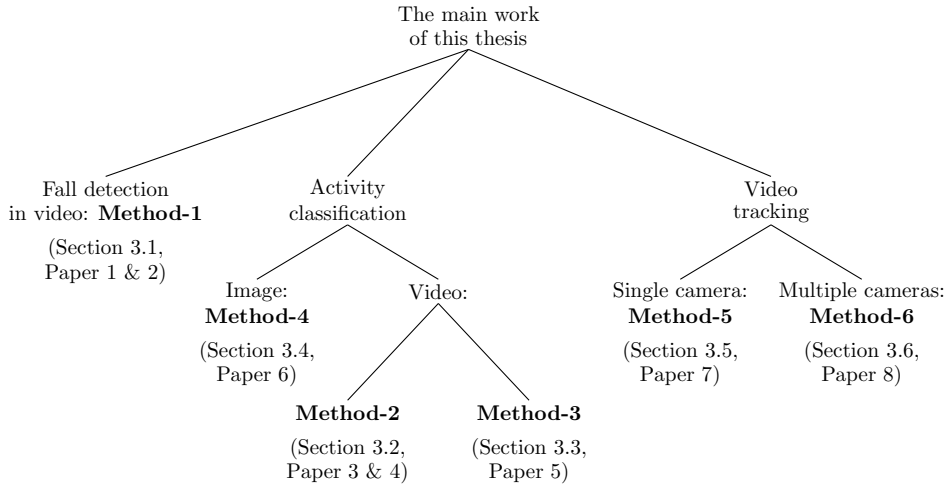


Figure 3.1: The main work of this thesis.

As shown in Fig. 3.1, 6 different methods for visual analysis of human activities are listed. Method-1 (Section 3.1, Paper 1 & 2) conducts fall detection in video. Method-2 (Section 3.2, Paper 3 & 4) and Method-3 (Section

3.3, Paper 5) deal with activity classification in video, while Method-4 (Section 3.4, Paper 6) does it in image. Method-5 (Section 3.5, Paper 7) and Method-6 (Section 3.6, Paper 8) handle video tracking by using a single-view video and multi-view videos, respectively.

All these methods utilize Riemannian manifolds, however they differ in the following aspects:

- Method-1 converts the problem of fall detection to the study of velocity statistics of points moving on the manifold.
- Method-2 uses 3 manifolds in layers (each corresponds to a different type of features) by adopting a divide and conquer strategy.
- Method-3 represents video activities as time sequences of BoW features on the manifold, and classifies them by a kernel machine based on DTW and geodesic distances.
- Method-4 exploits static cues in image and unifies different types of features as a point on the manifold for activity representation.
- Method-5 conducts online learning of target model by using one-class SVM and a kernel based on Riemannian manifolds.
- Method-6 employs multiple view geometry for occlusion handling with online learning on Riemannian manifolds.

3.1 Method-1: Fall Detection in Video by Analyzing Velocity Statistics of Manifold Features

(Summary for Papers 1 & 2)

Problem Addressed by Method-1: This method addresses issues in fall detection from videos. In this method, fall detection is formulated as a binary classification problem (total number of classes $K = 2$) that distinguishes falls from other activities. That is, all remaining activities are treated as one negative class.

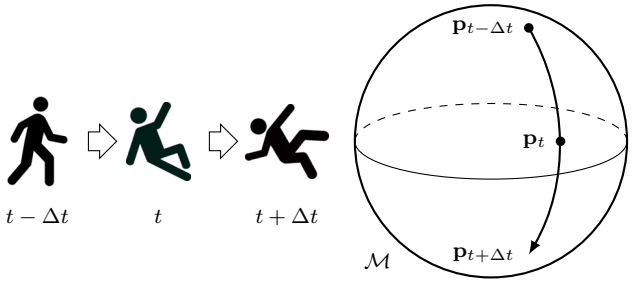


Figure 3.2: The basic idea of Method-1 that converts the analysis of dynamic features of a fall to the study of velocity statistics of points (e.g., $\mathbf{p}_{t-\Delta t}$, $\mathbf{p}_t$, and $\mathbf{p}_{t+\Delta t}$) on a manifold $\mathcal{M}$.

Basic Ideas: It is observed that a falling person undergoes large appearance change, shape deformation, and physical displacement, thus the focus of this method is to analyze these features that vary drastically when a fall occurs. This method performs the analysis on Riemannian manifolds, due to the following reasons:

- The nonlinear nature of manifolds is suitable for characterizing such dynamic processes caused by non-planar motion;
- Dynamic appearance, shape and motion can be effectively represented by points on low-dimensional Riemannian manifolds;
- Riemannian geometry provides a metric for measuring distances on the manifold, which allows the study of dynamics of manifold features.

By representing dynamic features of each type as points on a Riemannian manifold, the analysis of these features are converted to the study of velocity

statistics on that manifold. Intuitively, the more drastically the appearance changes, shape deforms or the physical movement varies, the more rapidly the corresponding manifold point moves. This idea is depicted in Fig. 3.2.

The Big Picture: The big picture of Method-1 is given in Fig. 3.3, with 4 major steps: (i) manifold representation of dynamic features; (ii) extraction of statistical features from the manifold; (iii) weighting of statistical features from different manifolds; (iv) fall detection by boosting and fusion. As the rectangular area in dashed line indicates, the core part of this method lies in step (ii), (iii) and (iv).

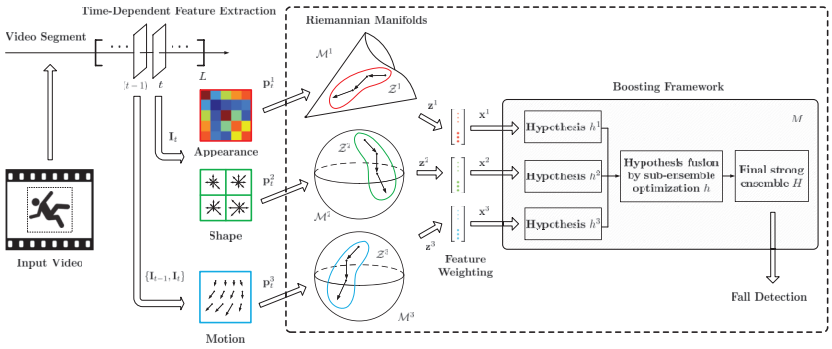


Figure 3.3: The big picture of Method-1.

Main Contributions:

- Dynamic features are represented as points moving on manifolds;
- Falls are characterized by velocity statistics of manifold points based on geodesics;
- Statistical features are combined and weighted by mutual information;
- Results are comparable to multi-camera and multi-modal methods.

Results: Method-1 is tested on 2 video datasets for fall detection. *Dataset-A* (collected from [162]) contains 184 "*Fall*" videos and 216 other activities. *Dataset-B* (collected from [165]) contains 60 "*Fall*" videos and 40 other activities . Key frames from the two datasets are shown in Fig. 3.4.

Method-1 is compared with 6 existing methods on these two datasets in terms of sensitivity and specificity [167], as shown in Table 3.1. It can be

(a) *Dataset-A*

(b) *Dataset-B*

Figure 3.4: Key frames from the two video datasets on fall detection. For each dataset, upper row: falls; lower row: other activities.

observed that this method achieves comparable results with multi-camera methods on Dataset-A, and outperforms other methods on *Dataset-B*.

Table 3.1: Comparison of Method-1 and other existing methods: sensitivity (Sens) and specificity (Spec) on the test set of *Dataset-A* and *Dataset-B*.

(a) *Dataset-A*

Method	Sensor Type	Sens (%)	Spec (%)
Auvinet *et al.* [109]	Multiple RGB Views	80.6	**100**
Rougier *et al.* [104]	Multiple RGB Views	95.4	95.8
Hung *et al.* [163]	Multiple RGB Views	95.8	**100**
Ma *et al.* [164]	RGB+Depth	**99.93**	91.97
Method-1	**Arbitrary RGB View**	98.55	95.84

(b) *Dataset-B*

Method	Sensor Type	Sens (%)	Spec (%)
Kwolek *et al.* [165]	Depth+Accelerometer	**100**	96.67
Bourke *et al.* [166]	Accelerometer	**100**	90.00
Method-1	**Arbitrary RGB View**	**100**	**97.25**

3.2 Method-2: Activity Classification in Video Using 3 Riemannian Manifolds

(Summary for Papers 3 & 4)

Problem Addressed by Method-2: This method addresses the problem of classifying human activities in video. Activities of interest include (but not limited to) activities of daily living, e.g., "*Eat*", "*Drink*", "*Use-laptop*", "*Read*", "*Lie-down*", "*Walk*", "*Sit-down*", and anomalies like "*Fall*".

Basic Ideas: Method-2 extracts part-based features from body parts (head, hands, waist center, feet) that matter to perform a certain type of activity, as they are key to characterize and distinguish that type of activity from remaining ones. The main idea for employing both local and global features is to extract features from body parts that are the key in performing one set of human activities (e.g., "*Eat*", "*Drink*", "*Use-laptop*", "*Read*"), while extract features from global body features they are key to another set of activities (e.g., "*Fall*", "*Lie-down*", "*Walk*", "*Sit-down*").

The motivations for exploiting Riemannian manifolds in feature representation are threefold:

1. The nonlinear nature of manifolds enables effective description of dynamic processes of human activities involving non-planar movement, which lie on a nonlinear manifold other than a vector space;

2. Many video features of human activities may be effectively described by low-dimensional data points on the Riemannian manifold while still maintaining the important property of human activities such as topology and geometry;

3. The Riemannian geometry provides a way to measure the distances of different activities on the nonlinear manifold, hence is suitable tool for the classification.

The essence for using 3 different Riemannian manifolds in a layered structure is to solve the classification problem with a divide and conquer strategy.

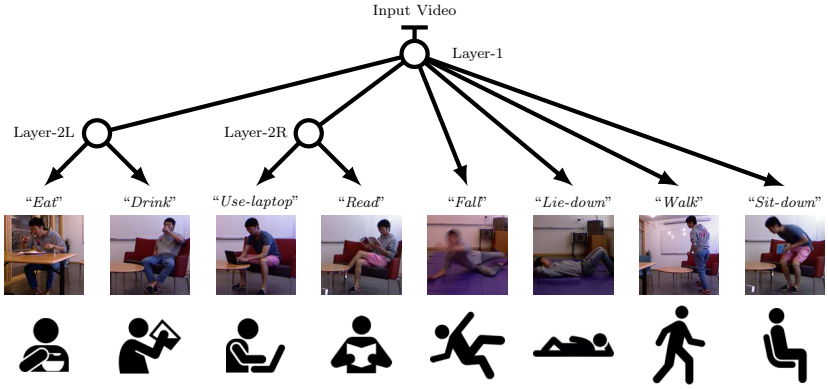


Figure 3.5: The big picture of Method-2 that has a layered structure, consisting of 3 Riemannian manifolds for activity classification using a divide and conquer strategy.

The Big Picture: As shown in Fig. 3.5, Method-2 is based on Riemannian manifolds that uses a tree structure of two layers, where nodes in each tree branch are on a Riemannian manifold, and correspond to different part-based covariance features and induce a geodesic-based kernel machine for classification. In the first layer, activities are classified according to the dynamics of body pose and the movement of hands or arms, where activities with similar body pose and motion but different human-object interaction are coarsely classified into the same category. In the second layer, the coarsely classified activities are further fine classified according to

the appearance of local image patches at hands in key frames, where the interacting objects as discriminative cues are likely to be attached.

Main Contributions:

- Motion of body parts for each video activity is characterized by global features, i.e., a covariance matrix of distances between each pair of key points and the orientations of lines that connect them;
- Human-object interaction is described by local features, i.e., the appearance of local regions around hands in key frames, where key frames are selected using the proximity of hands to other key points;
- Classification of human activities is formulated by a geodesic distance-induced kernel machine by exploiting pairwise geodesics on Riemannian manifolds under the log-Euclidean metric.

Results: Method-2 is tested on 2 video datasets for activity classification. *Dataset-A*, made on our university campus, contains a total number of 943 video activities from 8 classes (key frames shown in Fig. 3.6). Test results on *Dataset-A* in Table 3.2 have shown high classification accuracy (average 94.27%) and small false alarm rate (average 0.80%).

Figure 3.6: Key frames from *Dataset-A* containing activities from 8 classes. Upper row from left to right: "*Eat*", "*Drink*", "*Use-laptop*", and "*Read*". Lower row from left to right: "*Fall*", "*Lie-down*", "*Walk*", and "*Sit-down*".

Dataset-B [168] contains a total of 224 videos from 7 activity classes (key frames shown in Fig. 3.7). For *Dataset-B*, test results from Method-2 are compared with 6 existing methods in Table 3.3, where Method-2 has

Table 3.2: Performance of Method-2 on activity classification (8 classes) using *Dataset-A*: classification accuracy, and false positive rate (FPS) on the test set.

	Accuracy (%)	FPR (%)
"*Eat*"	96.30	0
"*Drink*"	90.74	0.24
"*Use-laptop*"	88.46	1.71
"*Read*"	90.38	2.21
"*Fall*"	96.30	0.24
"*Lie-down*"	100	0.48
"*Walk*"	94.57	0.77
"*Sit-down*"	97.33	0.77
Overall (*)	94.46	–
Average	**94.27**	**0.80**

(*) Overall: the total number of true positives for all classes divided by the total number of videos in the test set.

outperformed all these existing methods.

Figure 3.7: Key frames from *Dataset-B* containing activities from 7 classes. Upper row from left to right: "*Drink*", "*Eat*", "*Use-laptop*", and "*Read-cellphone*". Lower row from left to right: "*Make-phonecall*", "*Read-book*", and "*Use-remote*".

It is worthing noting the performance drop on *Dataset-B*, comparing to *Dataset-A*. This is probably due to the fact that key points used for

Table 3.3: Comparison of Method-2 and other existing methods on activity classification (7 classes) using *Dataset-B*: classification accuracy on the test set.

Method	Accuracy (%)
All the features + Boosting [168]	71.4
All the features + SVM [168]	68.7
Skeleton + LoP [173]	66.0
DSTIP + DCSF [170]	61.7
EigenJoints [171]	49.1
Moving Pose [172]	38.4
Method-2	**74.11**

experiments on *Dataset-B* are automatically estimated by Kinect, which may be less accurate than manually marking.

3.3 Method-3: Activity Classification in Video Using Time-Dependent BoW on Manifolds

(Summary for Paper 5)

Problem Addressed by Method-3: This method also addresses the problem of classifying human activities in video. Activities of interest are similar to that of Method-2.

Basic Ideas: The basic idea of Method-3 is to model the dynamic process of each video activity as a temporal sequence of bag of words (BoW) features on a Riemannian manifold, and classifies such time series with a kernel based on dynamic time warping (DTW), taking into account the underlying manifold geometry. The main idea for extracting both appearance and structural features from body parts is that one may give important cues for local human-object interaction while the other provide information on the global body pose and motion. The motivations for exploiting Riemannian manifolds in feature representation are similar to that of Method-2. The essence for using DTW-based kernels and geodesic distance-based local kernels is to fit for the classification of video activities with different lengths and the dynamic processes caused by non-planar movement of human that are described on the manifold.

The Big Picture: As shown in Fig. 3.8, Method-3 treats each video activity as a temporal sequence of BoW features on a Riemannian manifold,

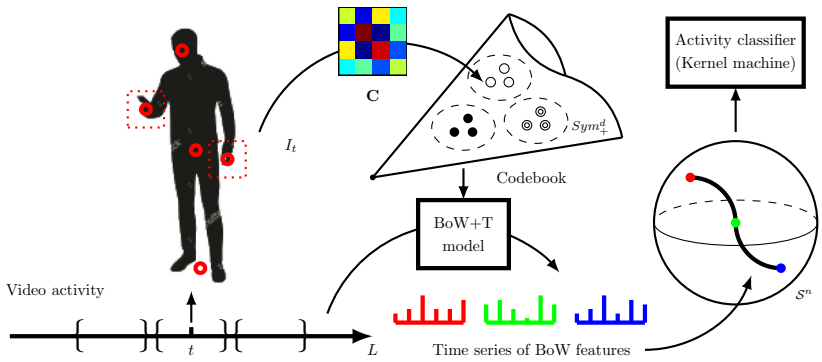


Figure 3.8: The big picture of Method-3. I_t is the t-th frame of the video activity, and L is the total number of frames. In frame I_t, "$\circ$" are key points (head, hands, waist center, midpoint of feet), and the areas with dotted edges are local patches centered at hands. $\mathbf{C} \in Sym^d_+$ is the covariance descriptor of local appearance and global structural features extracted from frame I_t. The codebook is generated by clustering covariance descriptors on the manifold of SPD matrices. The video activity is temporally segmented and represented by the BoW+T model as a sequence of BoW features. It is a time series of manifold points on a unit n-sphere that is classified by a kernel machine based on geodesic distance on the sphere.

and classifies such time series with a kernel based on dynamic time warping (DTW) and geodesic distances.

Main Contributions:

- A unified covariance matrix is used to represent structural features of body pose and appearance features of interacting objects at hands in each frame as a manifold point in the space of SPD matrices;
- Time-dependent BoW features on a unit n-sphere are extracted from each video activity as a time sequence of covariance descriptors;
- A positive definite kernel is formulated based on DTW and geodesic distances on the unit n-sphere for activity classification using time-dependent BoW features.

Results: Method-3 is tested on 2 video datasets (the same datasets used by Method-2) for activity classification. *Dataset-A*, made on our university

campus, contains a total number of 943 video activities from 8 classes (key frames shown in Fig. 3.6). Test results on *Dataset-A* in Table 3.4 have shown high classification accuracy (average 89.66%) and small false alarm rate (average 1.43%).

Table 3.4: Performance of Method-3 on activity classification (8 classes) using *Dataset-A*: classification accuracy, and false positive rate (FPS) on the test set.

	Accuracy (%)	FPR (%)
"*Eat*"	90.74	1.67
"*Drink*"	85.19	1.18
"*Use-laptop*"	86.54	2.36
"*Read*"	88.46	2.84
"*Fall*"	92.59	0.95
"*Lie-down*"	92.45	0.95
"*Walk*"	89.33	0
"*Sit-down*"	92.00	1.50
Overall (*)	89.77	–
Average	**89.66**	**1.43**

(*) Overall: the total number of true positives for all classes divided by the total number of videos in the test set.

Dataset-B [168] contains a total of 224 videos from 7 activity classes (key frames shown in Fig. 3.7). For *Dataset-B*, test results from Method-2 are compared with 6 existing methods in Table 3.5, where Method-3 has outperformed all these existing methods.

It is worth noting the performance drop on *Dataset-B*, comparing to *Dataset-A*. This is probably due to the fact that key points used for experiments on *Dataset-B* are automatically estimated by Kinect, which may be less accurate than manually marking.

Table 3.5: Comparison of Method-3 and other existing methods on activity classification (7 classes) using *Dataset-B*: classification accuracy on the test set.

Method	Accuracy (%)
All the features + Boosting [168]	71.4
All the features + SVM [168]	68.7
Skeleton + LoP [173]	66.0
DSTIP + DCSF [170]	61.7
EigenJoints [171]	49.1
Moving Pose [172]	38.4
Method-3	**72.34**

3.4 Method-4: Activity Classification in Image

Using Part-Based Features on Manifolds

(Summary for Paper 6)

Problem Addressed by Method-4: This method addresses the problem of activity classification in image, where only static cues are available.

Basic Ideas: Despite the fact that temporal information and motion cues in video provide discriminative features for activity classification, many human activities can be identified from individual images by purely exploiting static cues. Body pose is often used as the main feature for recognizing activities (e.g., sport actions) in still images. However, for activities of daily living in indoor environments, body poses are sometimes similar. Therefore, Method-4 also uses appearance of local image regions that may contain human-object interactions for feature representation of activities.

The reason for using image patches centered at hands is that an interacting object can be a useful cue for activity classification, and it is likely to be in the vicinity of a human hand (e.g. "*Eat*", "*Drink*", "*Use-laptop*", "*Read*"). The idea of employing covariance descriptors is to effectively and efficiently combine the appearance and structural features using a low-dimensional representation. The main motivation of applying geodesic-based kernel for SVM classification is that covariance descriptors are SPD matrices residing on a Riemannian manifold. Hence, the underlying Riemannian geometry shall be exploited for improved results.

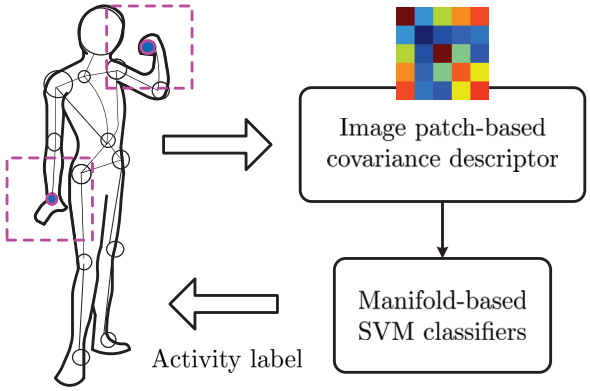


Figure 3.9: The big picture of Method-4 for activity classification in image. Areas with dashed edges are local image patches centered at detected hand points.

The Big Picture: As shown in Fig. 3.9, Method-4 mainly consists of the following steps (for each image):

1. Extraction of appearance features from local image patches at hands, and structural features from key points (head, hands, waist center) of human upper body;
2. Representation of activity by a unified covariance matrix as a fusion of different types of features;
3. Activity classification taking into account the underlying Riemannian geometry of covariance matrices.

Main Contributions:

- Activities in images are represented by appearance features from local patches at hands containing interacting objects, and by structural features formed from key parts of human upper body;
- A kernel function is formulated based on geodesics on Riemannian manifolds under the log-Euclidean metric;
- Results achieved outperform non-manifold or tangent-space methods.

Results: Method-4 is tested on an image dataset (collected from [173]) containing a total number of 2750 still images from 7 different classes (example images shown in Fig. 3.10). Results are reported in Table 3.6, with comparisons to three closely related classification methods.

- *Classifier-1 (C1)* (non-manifold SVM): directly applies SVM with RBF kernel to covariance descriptors;
- *Classifier-2 (C2)*: uses the identity matrix $\mathbf{I} \in Sym_d^+$ as the base point, and applies linear SVM in the tangent space of $\mathbf{I}$;
- *Classifier-3 (C3)*: uses a global mean $\boldsymbol{\mu} \in Sym_d^+$ as the base point, and applies linear SVM in the tangent space of $\boldsymbol{\mu}$.

Figure 3.10: Example images of human activities in each class. Upper row from left to right: (1) "*Drink*", (2) "*Eat*", (3) "*Read*", (4) "*Make-phonecall*". Lower row from left to right: (5) "*Use-laptop*", (6) "*Vacuum-clean*", and (7) "*Play-guitar*".

From Table 3.6, one can see that Method-4 achieves good classification accuracy (average 95.83%) while maintaining small false alarms (average 0.71%). Comparing with other classifiers, the method has significantly improved the performance.

Table 3.6: Comparison of Method-4 and three other methods: classification accuracy and false positive rate on the test set.

Activities	Accuracy (%)			
	C1	*C2*	*C3*	**Method-4**
"*Drink*"	58.33	45.83	36.67	**87.50**
"*Eat*"	75.00	**100**	83.33	**100**
"*Read*"	95.00	94.17	92.50	**100**
"*Make-phonecall*"	13.33	60.00	54.17	**83.33**
"*Use-laptop*"	**100**	**100**	99.17	**100**
"*Vacuum-clean*"	82.50	86.67	85.00	**100**
"*Play-guitar*"	95.83	**100**	**100**	**100**
Average	74.29	83.81	78.69	**95.83**
Activities	False positive rate (%)			
	C1	*C2*	*C3*	**Method-4**
"*Drink*"	5.69	8.54	10.90	**2.12**
"*Eat*"	5.71	**0**	2.95	**0**
"*Read*"	0.88	1.03	1.36	**0**
"*Make-phonecall*"	13.03	6.45	7.86	**2.82**
"*Use-laptop*"	**0**	**0**	0.14	**0**
"*Vacuum-clean*"	2.46	2.19	2.50	**0**
"*Play-guitar*"	0.7	**0**	**0**	**0**
Average	4.07	2.60	3.67	**0.71**

3.5 Method-5: Single-Camera Video Tracking by Manifold-Based One-Class SVM with Online Learning

(Summary for Paper 7)

Problem Addressed by Method-5: This method addresses the problem of object tracking in single-camera video, with online learning of the target model to mitigate tracking drift.

Basic Ideas: Tracking can be regarded as a one-class classification problem of domain-shift objects. The proposed tracker is inspired by the fact that the positive samples can be bounded by a closed hypersphere generated by one-class support vector machines (SVM), leading to a solution for robust learning of target model online.

In Method-5, object appearances are represented by covariance matrices. This is motivated by the fact that covariance matrices are low-dimensional descriptions of object appearances on a Riemannian manifold, and object dynamics such as appearance change and non-planar (or, out-of-plane) pose change may be more efficiently described by this nonlinear smooth manifold.

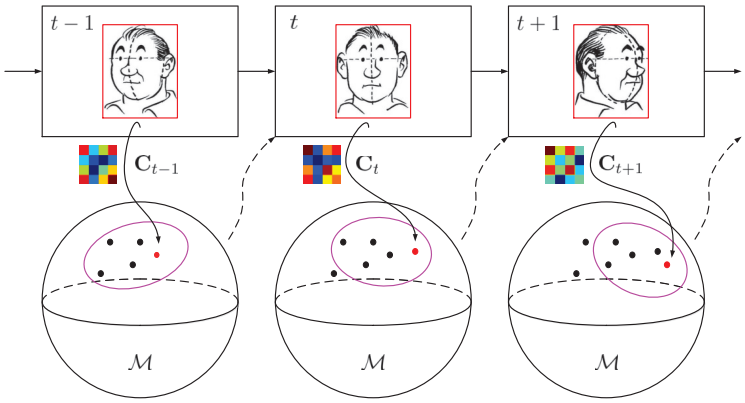


Figure 3.11: Illustration of Method-5 for tracking with online learning by one-class SVM on Riemannian manifolds. $\mathbf{C}_{t-1}$, $\mathbf{C}_t$ and $\mathbf{C}_{t+1}$ are covariance matrices computed from tracked object regions at $t-1$, t and $t+1$, respectively. The closed loops encircling manifold points are hyperspheres generated by one-class SVM on the manifold $\mathcal{M}$.

The Big Picture: Manifold points corresponding to tracked object regions in a temporal window of fixed size are kept. These points are used as positive samples for online learning of target model by one-class SVM, taking into account the underlying geometry of Riemannian manifolds. This will yield a hypersphere as a closed loop encircling the points on the manifold, where newly observed points corresponding to candidate object regions in next frame are classified as inliers if they are inside the loop, and outliers otherwise. Inliers with highest score (or, confidence) are picked as the detected object region. Then, the temporal window of manifold points is updated on a first-in-first-out (FIFO) basis, as depicted in Fig.3.11. Hence, the hypersphere generated by one-class SVM essentially characterize the object dynamics, as a cluster of points flowing on the manifold.

Main Contributions:

- Target model is represented by a set of positive samples as a cluster of points on Riemannian manifolds;
- Online learning is performed as obtaining a dynamic cluster of points that flows on the manifold, in an alternate manner with tracking;
- A kernel function for one-class SVM is formulated based on geodesics on the manifold under the log-Euclidean metric.

Results: The method is compared with 5 other methods, namely *Online Boosting* (OB) [174], *Beyond SemiBoosting* (BSB) [175], *Mean Shift* (MS) [176], *Compressive Tracker* (CT) [177], and *Covariance Tracker* [178]. The tracking results in some key frames and error curves on center deviation of the bounding boxes are shown in Fig. 3.12, 3.13 and 3.14, respectively.

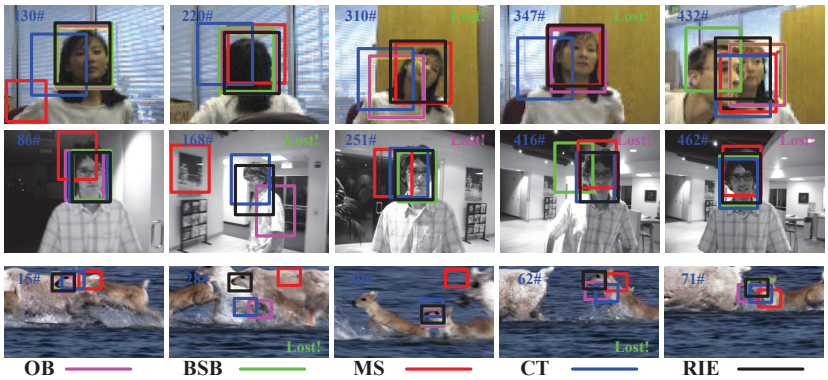


Figure 3.12: Tracking results of Method-5 (RIE) in key frames of "girl", "David" and "deer" sequences, compared with 4 other methods.

Observing the results in Fig. 3.12, 3.13 and 3.14, one can see that Method-5 follows the object with high accuracies and consistently low errors, under various appearance and pose changes in different videos.

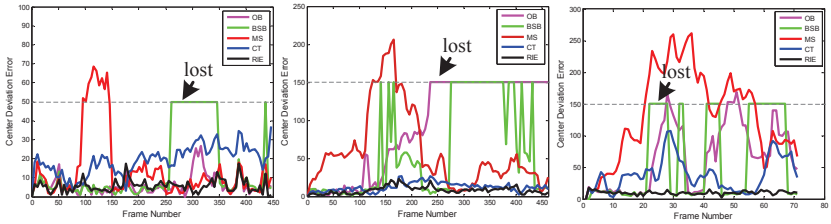


Figure 3.13: From left to right columns: center deviation error of Method-5 (RIE) on "girl", "David" and "deer" sequences, compared with 4 other methods. Horizontal line means that tracking is completely lost.

Figure 3.14: Tracking results of Method-5 (green box) in key frames of "race" sequence, compared with [178] (red box).

3.6 Method-6: Multi-Camera Video Tracking with Manifold-Based Online Learning and Occlusion Handling by Exploiting Multiple View Geometry

(Summary for Paper 8)

Problem Addressed by Method-6: This method addresses the problem of object tracking using multi-camera video, with online learning of the target model to mitigate tracking drift caused by occlusions. It is assumed that objects are visible in at least one view and move uprightly on a common planar ground that may induce a homography relation between views.

Basic Ideas: In Method-6, multiple uncalibrated cameras with overlapping fields of view are employed. The essence for using multiple cameras to exploit the wide spatial coverage that is advantageous in handling complex scenarios, including long-term full occlusions. We treat an object in different views as different points on a same manifold. Hence, the solution of multi-

view object tracking is equivalent to defining a similarity measure on the manifold, finding the best view object under the measure for a given reference set, and mapping it to the desired view under geometrical constraints. The reason to use uncalibrated cameras is that for outdoor scenarios where objects are located at large distances to cameras, it is difficult to accurately estimate 3-D point correspondences, where accurate camera calibration is non-trivial. Instead, we directly exploit underlying multi-view geometric constraints, without the attempt to estimate camera parameters.

Similar to Method-5, object appearances are represented by covariance matrices. This is motivated by the fact that covariance matrices are low-dimensional descriptions of object appearances on a Riemannian manifold, and object dynamics such as appearance change and non-planar (or, out-of-plane) pose change may be more efficiently described by this nonlinear smooth manifold.

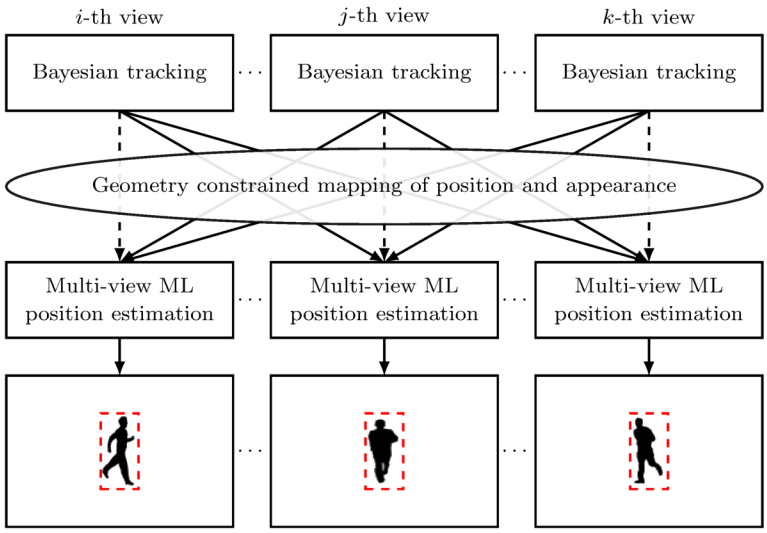


Figure 3.15: The big picture of Method-6.

The Big Picture: Method-6 consists of two major parts: (i) multi-view Maximum Likelihood (ML) tracking; (ii) online learning of object appearances on the Riemannian manifold. These two process are performed in an alternative fashion.

As shown in Fig. 3.15, for the tracking part, we adopt a three-layer scheme where tracking is first done independently in each individual view. Then, tracking results are mapped from each view to the remaining views by geometrical constraints. Finally, a manifold-based maximum likelihood (ML) criterion is applied to obtain the optimal tracking result.

Main Contributions:

- A similarity measure is defined, based on geodesics between a candidate object and a set of mapped references from multiple views on a Riemannian manifold;
- Multi-view maximum likelihood (ML) is employed for the estimation of object bounding box parameters, based on Gaussian-distributed geodesics on the manifold;
- Online learning of target model is performed as updating the position of a point on the manifold, with a criterion to detect possible occlusions;
- Projective transformations are used for mapping objects between views, where parameters are estimated from warped vertical axis by combining planar homography, epipolar geometry and vertical vanishing point;

Results: Method-6 is tested on different sets of multi-view videos containing occlusions. Key frames in several cases are shown in Fig. 3.16, 3.17, and 3.18 as examples of the tracking performance. It can be observed that the tracker follows the target person accurately despite of frequent occlusions. Also, the method is evaluated with objective measures and compared with some existing multi-view trackers. For example, Table 3.7 shows the comparison with [179], where Method-6 gives smaller tracking errors.

(a) Top view (b) Camera views

Figure 3.16: Tracking results of Method-6 in Case-e. (a) Trajectories of tracked object (magenta) and the ground truth (yellow) on the ground plane through planar homography mapping. (b) Rows 1-2: Camera views 1-2. Key frames (# 19, 73) are selected.

Figure 3.17: Tracking results of Method-6 in Case-f. Rows 1-3: Camera views 1-3. Key frames (# 3201, 3272, 3702, 3835) are selected.

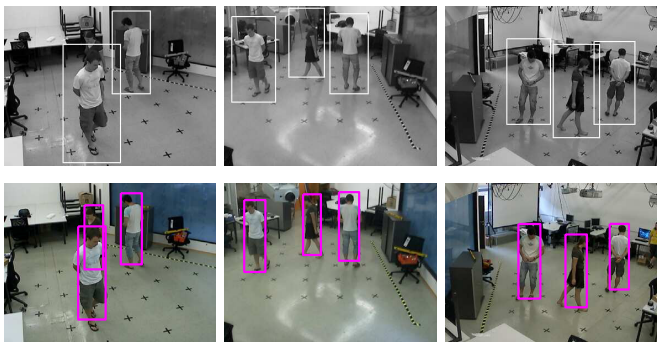


Figure 3.18: Comparison: Tracking results from Method-6 and [179] in Case-h. Upper row: from [179] for views 1-3 (Column 1-3). Lower row: from Method-6 for views 1-3 (Column 1-3). Key frames (# 510, 1089, 1481) are selected.

Table 3.7: Comparison: Tracking errors based on different criteria of Method-6 and [179] in Case-h.

(a) Euclidean distance

Method	View 1	View 2	View 3	Average
Roth *et al.* [179]	119.89	103.40	101.37	108.22
Method-6	**8.1616**	**6.4158**	**6.4999**	**7.0258**

(b) Bhattacharyya distance

Method	View 1	View 2	View 3	Average
Roth *et al.* [179]	0.4364	0.3360	0.2917	0.3547
Method-6	**0.0639**	**0.0506**	**0.0503**	**0.0549**

(c) Geodesic distance

Method	View 1	View 2	View 3	Average
Roth *et al.* [179]	2.8850	2.5056	2.6074	2.6660
Method-6	**0.5536**	**0.3805**	**0.5428**	**0.4923**

3.7 Discussion and Comparison of Proposed Methods

Method-1 (fall detection in video): Since falls are dynamic processes containing abrupt motion that cannot be easily detected from still images or be distinguished from other activities by using static cues alone, we addressed the problem of fall detection in videos, instead of images.

Method-2, 3, 4 (activity classification): Each of Method-2, Method-3, and Method-4 has a multi-class problem of activity classification. Different from the other two, Method-4 deals with still images, based on the fact that many activities can be identified from individual images using the information of body pose and human-object interaction. Inspired by this image-based method, Method-2 recognizes some video activities from key frames that are automatically selected based on criteria. Also, Method-3 extracts similar features that are used in Method-4 from each frame, and represent video activities as time sequences of manifold points based on these features.

Method-3 and Method-4 were tested on the same video datasets, where experimental results showed that Method-3 has better performance. However, one cannot simply draw a conclusion that Method-3 is better. From a methodological point of view, Method-3 is less dependent on time, which may not be suitable for distinguishing activities where temporal information is important, e.g., "*Sit-down*" vs "*Sit-up*". On the other hand, Method-4 has more parameters to tune, for example, the number of clusters for learning the codebook, and the number of segments for each video activity. The comparison between Method-3 and Method-4 is summarized in Table 3.8.

Table 3.8: Comparison between Method-2 and Method-3 on activity classification in video.

Method	Description	Pros	Cons
Method-2	3 layered manifolds	Suitable for activities that can be identified from key frames	Less dependent on time, not suitable for activities that are heavily dependent on time
Method-3	Temporal BoW on manifolds	Suitable for time-dependent activities	Parameter tuning is not trivial

Method-5, 6 (video tracking): Method-5 deals with tracking drift and possible occlusions in a single camera view. It can handle occlusions to some extent, but is not suitable for tracking objects that undergo long-term full occlusions. Method-6 that employs multiple cameras is more suitable for handling complex scenarios including full occlusions, at the cost of increased complexity. The comparison between Method-5 and Method-6 is summarized in Table 3.9.

Table 3.9: Comparison between Method-5 and Method-6 on video tracking.

Method	Description	Pros	Cons
Method-5	Single camera	Less complex, can handle short-term partial occlusions	Not suitable for long-term full occlusions
Method-6	Multiple cameras	Suitable for complex scenarios including long-term full occlusions	Increased complexity related to the number of cameras

Chapter 4

Conclusion

In this thesis, six different methods for visual analysis of human activities are introduced, including fall detection in video, activity classification in image and video, and video tracking using single camera and multiple cameras. Considering the contribution in theoretical aspects, we have investigated the use of Riemannian manifolds for mathematical modeling of video activities, and developed new methods for characterizing and distinguishing different activities. The methods can be used to recognize activities of daily living, to detect abnormal activities, and to track targets in various scenarios. Experiments on real-world datasets were conducted to evaluate the performance of the proposed methods. Results, comparisons, and evaluations showed that the methods achieved state-of-the-art performance. From the perspective of application, the methods have a wide range of potential applications such as assisted living, smart home, eHealthcare, smart vehicles, office automation, safety systems and services, security systems, situation-aware human-computer interfaces, robot learning, etc.

4.1 Future Work

Despite the notable progress and promising results, there remain many open issues in visual analysis of human activities. Among them are robust estimation of body pose or skeleton that serves as a crucial pre-processing step for feature extraction, accurate segmentation of video activities that clearly defines the start and end of an activity, as well as privacy concerns and the related ethic or safety issues. Moreover, deep learning can be employed and combined with the proposed methods to further improve the performance.

References

[1] M. Mubashir, L. Shao, L. Seed, "A survey on fall detection: principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[2] P. Kostopoulos *et al.*, "F2D: a fall detection system tested with real data from daily life of elderly people," *IEEE International Conference on E-Health Networking, Application and Services (Healthcom)*, 2015.

[3] Y. Yun, I.Y.H. Gu, H. Aghajan, "Riemannian manifold-based support vector machine for human activity classification in images," *IEEE International Conference on Image Processing (ICIP)*, pp. 3466–3469, 2013.

[4] Y. Yun *et al.*, "Human activity recognition in images using SVMs and geodesics on smooth manifolds," *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2014.

[5] Y. Yun, I.Y.H. Gu, "Human fall detection via shape analysis on Riemannian manifolds with applications to elderly care," *IEEE International Conference on Image Processing (ICIP)*, 2015.

[6] Y. Yun *et al.*, "Fall detection in RGB-D videos for elderly care," *IEEE International conference on E-Health Networking, Application & Services (Healthcom)*, 2015.

[7] G. Guo, A. Lai, "A survey on still images based human action recognition," *Pattern Recognition*, vol. 47, pp. 3343–3361, 2014.

[8] J.K. Aggarwal, M.S. Ryoo, "Human activity analysis: A review," *ACM Computing Survey (CSUR)*, vol. 43, no. 3, article 16, 2011.

[9] S. Vishwakarma, A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *The Visual Computer*, vol. 29, pp. 983–1009, 2013.

[10] G. Cheng *et al.*, "Advances in Human Action Recognition: A survey," arXiv:1501.05964, 2015.

[11] A.F. Bobick, J.W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 23, no. 3, pp. 257-267, 2001.

[12] Y. Tian *et al.*, "Hierarchical filtered motion for action recognition in crowded videos," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 3, pp. 313-323, 2012.

[13] M. Blank *et al.*, "Actions as space-time shapes," *IEEE Internatonal Conference on Computer Vision (ICCV)*, vol. 2, pp. 1395-1402, 2005.

[14] D. Weinland, R. Ronfard, E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding (CVIU)*, vol. 104, no. 23, pp. 249–257, 2006.

[15] A. Yilmaz, M. Shah, "Actions sketch: a novel action representation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 984-989, 2005.

[16] S. Sadanand, J.J. Corso, "Action bank: A high-level representation of activity in video," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1234-1241, 2012.

[17] L. Shao *et al.*, "Spatio-temporal laplacian pyramid coding for action recognition," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 817-827, 2014.

[18] I. Laptev, "On space-time interest points," *International Journal of Computer Vision (IJCV)*, vol. 64, no. 2, pp. 107-123, 2005.

[19] G. Willems, T. Tuytelaars, L. Gool, "An efficient dense and scale-invariant spatiotemporal interest point detector," *European Conference on Computer Vision (ECCV)*, pp. 650-663, 2008.

[20] A. Kläser, M. Marszaek, C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," *British Machine Vision Conference (BMVC)*, 2008.

[21] I. Laptev *et al.*, "Learning realistic human actions from movies," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[22] V. Kantorov, I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2593-2600, 2014.

[23] G. Zhao, M. Pietikäinen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 29, no. 6, pp. 915928, 2007.

[24] V. Kellokumpu, G. Zhao, M. Pietikäinen, "Human activity recognition using a dynamic texture based method," *British Machine Vision Conference (BMVC)*, pp. 885-894, 2008.

[25] R. Messing, C. Pal, H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," *International Conference on Computer Vision (ICCV)*, pp. 104-111, 2009.

[26] P.K. Matikainen, M. Hebert, R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," *Workshop on Video-Oriented Object and Event Classification (in conjunction with ICCV)*, 2009.

[27] Y.G. Jiang *et al.*, "Trajectory-based modeling of human actions with motion reference points," *European Conference on Computer Vision (ECCV)*, pp. 425-438, 2012

[28] H. Wang, C. Schmid, "Action recognition with improved trajectories," *International Conference on Computer Vision (ICCV)*, pp. 3551-3558, 2013.

[29] P. Dollar *et al.*, "Behavior recognition via sparse spatio-temporal features," *International Conference on Computer Communications and Networks*, pp. 65-72, 2005.

[30] H. Wang *et al.*, "Action recognition by dense trajectories," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169-3176, 2011.

[31] D. Oneata, J. Verbeek, C. Schmid. "Action and event recognition with fisher vectors on a compact feature set," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1817-1824, 2013.

[32] X. Peng *et al.*, "Action Recognition with Stacked Fisher Vectors," *European Conference on Computer Vision (ECCV)*, pp. 581-595, 2014.

[33] H. Jgou *et al.*, "Aggregating local descriptors into a compact image representation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3304-3311, 2010.

[34] R. Arandjelović, A. Zisserman. "All about VLAD," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[35] A. Veeraraghavan, A.K. Roy-Chowdhury, R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 12, pp. 1896–1909, 2005.

[36] P. Turaga, A. Veeraraghavan, R. Chellappa, "From videos to verbs: mining videos for events using a cascade of dynamical systems," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[37] P. Turaga, A. Veeraraghavan, R. Chellappa, "Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[38] P. Turaga, A. Veeraraghavan, A. Srivastava, R. Chellappa, "Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2273–2286, 2011.

[39] M.F. Abdelkader, W. Abd-Almageeda, A. Srivastava, R. Chellappa, "Silhouette-based gesture and action recognition via modeling trajectories on Riemannian manifolds," *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 439–455, 2011.

[40] Y.M. Lui, J.R. Beveridge, M. Kirby, "Action classification on product manifolds," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[41] Y.M. Lui, J.R. Beveridge, M. Kirby, "Tangent bundle for human action recognition," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.

[42] A. Veeraraghavan, A. Srivastava, A.K. Roy-Chowdhury, R. Chellappa, "Rate-invariant recognition of humans and their activities," *IEEE Transactions on Image Processing*, vol. 6, pp. 1326–1339, 2009.

[43] R. Li, R. Chellappa, "Aligning spatio-temporal signals on a special manifold," *European Conference on Computer Vision (ECCV)*, 2010.

[44] K. Guo, P. Ishwar, J. Konrad, "Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels," *International Conference on Pattern Recognition*, 2010.

[45] K. Guo, P. Ishwar, J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," *International Conference on Advanced Video and Signal-Based Surveillance*, 2010.

[46] R. Chaudhry, Y. Ivanov, "Fast approximate nearest neighbor methods for non-euclidean manifolds with applications to human activity analysis in videos," *European Conference on Computer Vision*, 2010.

[47] P. Turaga, R. Chellappa, "Nearest-neighbor algorithms on non-euclidean manifolds for computer vision applications," *Indian Conference on Computer Vision, Graphics, and Image Processing*, 2011.

[48] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[49] A.J. Robinson, F. Fallside, "Static and dynamic error propagation networks with application to speech coding," *Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 632-641, 1988.

[50] S. Ji *et al.*, "3d convolutional neural networks for human action recognition," *International Conference on Machine Learning (ICML)*, 2010.

[51] A. Karpathy *et al.*, "Large-scale video classification with convolutional neural networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725-1732, 2014.

[52] J.Y.H. Ng *et al.*, "Beyond short snippets: Deep networks for video classification," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694-4702, 2015.

[53] M. Baccouche *et al.*, "Sequential deep learning for human action recognition," *International Workshop on Human Behavior Understanding*, 2011.

[54] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625-2634, 2015.

[55] K. Simonyan, A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 568-576, 2014.

[56] L. Wang, Y. Qiao, X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4305-4314, 2015.

[57] Z. Wu *et al.*, "Multi-stream multi-class fusion of deep networks for video classification," *ACM Multimedia Conference (MM)*, 2016.

[58] Xing Yan *et al.*, "Modeling video dynamics with deep dynencoder," *European Conference on Computer Vision (ECCV)*, pp. 215-230, 2014.

[59] N. Srivastava, E. Mansimov, R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," arXiv:1502.04681, 2015.

[60] I. Goodfellow *et al.*, "Generative adversarial nets," *Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2672-2680, 2014.

[61] M. Mathieu, C. Couprie, Y. LeCun, "Deep multi-scale video prediction beyond mean square error," arXiv:1511.05440, 2015.

[62] R. Goroshin *et al.*, "Unsupervised learning of spatiotemporally coherent metrics," *International Conference on Computer Vision (ICCV)*, pp. 4086-4093, 2015.

[63] I. Misra, C.L. Zitnick, M. Hebert, "Unsupervised learning using sequential verification for action recognition," arXiv:1603.08561, 2016.

[64] X. Wang, A. Farhadi, A. Gupta, "Actions∼transformations," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[65] Y. Wu, T. Yu, G. Hua, "Tracking appearances with occlusions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 789–795, 2003.

[66] Y. Huang, I. Essa, "Tracking multiple objects through occlusions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 1051–1058, 2005.

[67] J. Pan, B. Hu, "Robust occlusion handling in object tracking," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[68] N. Amezquita, R. Alquezar, F. Serratosa, "Dealing with occlusion in a probabilistic object tracking method," *IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum (in conjunction with CVPR)*, pp. 1–8, 2008.

[69] N. Papadakis, A. Bugeau, "Tracking with occlusions via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 33, no. 1, pp. 144–157, 2011.

[70] G. Chao, S. Jeng, S. Lee, "An improved occlusion handling for appearance-based tracking," *IEEE International Conference on Image Processing (ICIP)*, pp. 465–468, 2011.

[71] S. Kwak *et al.*, "Learning occlusion with likelihoods for visual tracking," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1551–1558, 2011.

[72] Z.H. Khan, I.Y.H. Gu, "Bayesian online learning on Riemannian manifolds using a dual model with applications to video object tracking," *IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition (in conjunction with ICCV)*, pp. 1402–1409, 2011.

[73] X. Li *et al.*, "Visual tracking via incremental log-Euclidean Riemannian subspace learning," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, Jun. 23 - 28, 2008.

[74] Y. Wu *et al.*, "Real-time visual tracking via incremental covariance tensor learning," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1631–1638, 2009.

[75] H. Aghajan, A. Cavallaro, "Multi-Camera Networks: Principles and Applications," *Academic Press*, edition 1, 2009.

[76] A. Mittal, L.S. Davis, "M_2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," *International Journal on Computer Vision (IJCV)*, vol. 51, no. 3, pp. 189–203, 2003.

[77] J. Chen, Q. Ji, "Efficient 3D upper body tracking with self-occlusions," *IAPR International Conference on Pattern Recognition (ICPR)*, pp. 3636–3639, 2010.

[78] J. Harguess, C. Hu, J.K. Aggarwal, "Occlusion robust multi-camera face tracking," *IEEE International Workshop on Machine Learning for Vision-based Motion Analysis (in conjunction with CVPR)*, pp. 31–38, 2011.

[79] J. Kang, I. Cohen, G. Medioni, "Multi-views tracking within and across uncalibrated camera streams," *ACM SIGMM International Workshop on Video Surveillance*, pp. 21–33, 2003.

[80] Y.D. Wang, J.K. Wu, A.A. Kassim, "Particle filter for visual tracking using multiple cameras," *IAPR International Conference on Machine Vision Applications*, pp. 298–301, 2005.

[81] J. Fan *et al.*, "Distributed multi-camera object tracking with Bayesian inference," *IEEE International Symposium on Circuits and Systems*, pp. 357–360, 2011.

[82] Y. Zhou, H. Nicolas, J. Benois-Pineau, "A multi-resolution particle filter tracking in a multi-camera environment," *IEEE International Conference on Image Processing (ICIP)*, pp. 4065–4068, 2009.

[83] C. Chu *et al.*, "Tracking across multiple cameras with overlapping views based on brightness and tangent transfer functions," *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–6, 2011.

[84] W. Qu, D. Schonfeld, M. Mohamed, "Decentralized multiple camera multiple object tracking," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 245–248, 2006.

[85] A.C. Sankaranarayanan, R. Chellappa, "Optimal multi-view fusion of object locations," *IEEE Workshop on Motion and Video Computing*, pp. 1–8, 2008.

[86] W. Du, J. Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," *Asian Conference on Computer Vision (ACCV)*, vol. 1, pp. 365–374, 2007.

[87] Z. Yue, S.K. Zhou, R. Chellappa, "Robust two-camera tracking using homography," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1–4, 2004.

[88] B. Kwolek, "Multi camera-based person tracking using region covariance and homography constraint," *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 294–299, 2010.

[89] S. Calderara, A. Prati, R. Cucchiara, "HECOL: homography and epipolar-based consistent labeling for outdoor park surveillance," *Computer Vision and Image Understanding (CVIU)*, vol. 111, no. 1, pp. 21–42, 2008.

[90] T.V. Duong, H.H. Bui, D.Q. Phung, S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 838–845, 2005.

[91] T.V. Duong, D.Q. Phung, H.H. Bui, S. Venkatesh, "Human behavior recognition with generic exponential family duration modeling in the hidden semi-Markov model," *International Conference on Pattern Recognition (ICPR)*, vol. 3, pp. 202–207, 2006.

[92] Z. Zhou *et al.*, "Activity analysis, summarization, and visualization for indoor human activity monitoring," *IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT)*, vol. 18, no. 11, pp. 1489–1498, 2008.

[93] K. Avgerinakis, A. Briassouli, I. Kompatsiaris, "Recognition of activities of daily living for smart home environments," *International Conference on Intelligent Environment*, 2013.

[94] Y. Yan *et al.*, "It's all about habits - Exploiting multi-task clustering for activities of daily living analysis," *IEEE International Conference on Image Processing (ICIP)*, 2014.

[95] A. Ghali, A.S. Cunningham, T.P. Pridmore, "Object and event recognition for stroke rehabilitation," *Visual Communications and Image Processing*, pp. 980–989, 2003.

[96] J. Gao *et al.*, "Dining activity analysis using a hidden markov model," *International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 915–918, 2004.

[97] M. Goffredo *et al.*, "Markerless human motion analysis in GaussLaguerre transform domain: An application to sit-to-stand in young and elderly people," *IEEE Transactions on Information Technology in Biomedicine (T-ITB)*, vol. 13, no. 2, pp. 207–216, 2009.

[98] A. Leu, D. Ristic-Durrant, A. Graser, "A robust markerless vision-based human gait analysis system," *IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 415–420, 2011.

[99] Y. Li, S. Miaou, C.K. Hung, J.T. Sese, "A gait analysis system using two cameras with orthogonal view," *IEEE International Conference on Multimedia Technology (ICMT)*, pp. 2841–2844, 2011.

[100] C. Liu *et al.*, "Understanding of human behaviors from videos in nursing care monitoring systems," *Journal of High Speed Networks - Broadband Multimedia Sensor Networks in Healthcare Applications*, vol. 16, no. 1, pp. 91–103, 2007.

[101] F.E. Martinez-Perez, J.A. Gonzalez-Fraga, M. Tentori, "Artifacts' roaming beats recognition for estimating care activities in a nursing home," *International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2010.

[102] J.S. Tham, Y.C. Chang, M.F.A. Fauzi, "Automatic identification of drinking activities at home using depth data from RGB-D camera," *International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2014.

[103] K. Zhan, F. Ramos, S. Faux, "Activity recognition from a wearable camera," *International Conference on Control, Automation, Robotics & Vision (ICARCV)*, 2012.

[104] C. Rougier *et al.*, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT)*, vol. 21, no. 5, pp. 611–622, 2011.

[105] T. Banerjee *et al.*, "Day or night activity recognition from video using fuzzy clustering techniques," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 483–493, 2013.

[106] X. Ma *et al.*, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[107] S. Fleck, W. Straßer, "Smart camera based monitoring system and its application to assisted living," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1698–1714, 2008.

[108] I. Charfi *et al.*, "Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification," *Journal of Electronic Imaging*, vol. 22, no. 4, 041106, pp. 1–17, 2013.

[109] E. Auvinet *et al.*, "Fall detection with multiple cameras: an occlusion-resistant method based on 3-D silhouette vertical distribution," *IEEE Transactions on Information Technology in Biomedicine (T-ITB)*, vol. 15, no. 2, pp. 290–300, 2011.

[110] T. Banerjee *et al.*, "Sit-to-stand measurement for in-home monitoring using voxel analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1502–1509, 2014.

[111] E.E. Stone, M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 290–301, 2015.

[112] J.M. Lee, "Introduction to Smooth Manifolds," *Springer*, 2012.

[113] X. Pennec, P. Fillard, N. Ayache, "A Riemannian framework for tensor computing," *International Journal of Computer Vision (IJCV)*, vol. 66, no. 1, pp. 41–66, 2006.

[114] V. Arsigny *et al.*, "Geometric means in a novel vector space structure on symmetric-positive definite matrices," *SIAM Journal on Matrix Analysis and Applications (SJMAEL)*, vol. 29, no. 1, pp. 328–347, 2008.

[115] Z.H. Khan, I.Y.H. Gu, "Online domain-shift learning and object tracking based on nonlinear dynamic models and particle filters on Riemannian manifolds," *Computer Vision and Image Understanding (CVIU)*, vol. 125, pp. 97–114, 2014.

[116] R. Subbarao, P. Meer, "Nonlinear mean shift over Riemannian manifolds," *International Journal of Computer Vision (IJCV)*, vol. 84, no. 1, pp. 1–20, 2009.

[117] O. Tuzel, F. Porikli, P. Meer, "Region covariance: a fast descriptor for detection and classification," *European Conference on Computer Vision (ECCV)*, pp. 589–600, 2006.

[118] O. Tuzel, F. Porikli, P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 30, no. 10, pp. 1713–1727, 2008.

[119] S.T. Lovett, "Differential Geometry of Manifolds," *A K Peters/CRC Press*, 2010.

[120] S. Jayasumana *et al.*, "Combining multiple manifold-valued descriptors for improved object recognition," *IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2013.

[121] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[122] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.

[123] T. Ojala, M. Pietikäinen, T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 24, no. 7, pp. 971–987, 2002.

[124] N. Dalal, "Finding people in images and videos," *PhD Thesis*, Grenoble Institute of Technology (INPG), 2006.

[125] M. Sonka, V. Hlavac, R. Boyle, "Image Processing, Analysis and Machine Vision," *Springer*, 1993.

[126] D. Sun, S. Roth, M.J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision (IJCV)*, vol. 106, no. 2, pp. 115–137, 2014.

[127] J. Perš *et al.*, "Histograms of optical flow for efficient representation of body motion," *Pattern Recognition Letters*, vol. 31, pp. 1369–1376, 2010.

[128] R. Chaudhry *et al.*, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[129] R.V.H.M. Colque, C.A.C. Júnior, W.R. Schwartz, "Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos," *SIBGRAPI Conference on Graphics, Patterns and Images*, 2015.

[130] T.S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 18, no. 10, pp. 959–971, 1996.

[131] L. Fei-Fei, P. Perona, "A Bayesian hierarchical model for learning natural scene categories," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[132] L. Fei-Fei, "Object recognition: bag of words models & part-based generative models," *Lecture Notes, Stanford Vision Lab*, 2011.

[133] L. Ballan, L. Seidenari, "Hands on advanced bag-of-words models for visual object recognition," *IAPR International Conference on Pattern Recognition (ICPR)*, 2014. Software (available online): https://github.com/lambertoballan/handsonbow.

[134] P.F. Felzenszwalb *et al.*, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[135] S. Lazebnik, C. Schmid, J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[136] Y. Bengio, A. Courville, P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 35, no. 8, pp. 17981828, 2013.

[137] Y. LeCun, M. Ranzato, "Deep Learning Tutorial," *International Conference on Machine Learning (ICML)*, 2013.

[138] H. Lee *et al.*, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, 2011.

[139] G.E. Hinton, R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.

[140] Y. Bengio, Y. LeCun, "Scaling learning algorithms towards AI," *Large-Scale Kernel Machines, MIT Press*, 2007.

[141] I. Goodfellow *et al.*, "Measuring invariances in deep networks," *Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.

[142] A. Dosovitskiy *et al.* , "Discriminative unsupervised Ffeature learning with convolutional neural networks," *Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[143] Z. Zuo *et al.* , "Convolutional recurrent neural networks: learning spatial dependencies for image representation," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015.

[144] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[145] C.W. Hsu, C.J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[146] R.E. Schapire, Y. Singer, "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[147] Y. Freund, R.E. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[148] T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *Springer Series in Statistics*, Second Edition, 2009.

[149] S. Gudmundsson, T.P. Runarsson, S. Sigurdsson, "Support vector machines and dynamic time warping for time series," *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2008.

[150] L. Chen, R. Ng, "On the marriage of Lp-norms and edit distance," *International Conference on Very Large Data Bases (VLDB)*, pp. 792–803, 2004.

[151] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 31, no. 2, pp. 306–318, 2008.

[152] P.-F. Marteau, S. Gibet, "On recursive edit distance kernels with application to time series classification," *IEEE Transactions on Neural Networks and Learning Systems (T-NNLS)*, vol. 26, no. 6, pp. 1121–1133, 2015.

[153] T. Hofmann, B. Schölkopf, A.J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[154] M. Cuturi, "Fast global alignment kernel," *International Conference on Machine Learning (ICML)*, 2011.

[155] Z. Chen, "Bayesian filtering: from Kalman filters to particles filters, and beyond," *Statistics*, pp. 169, 2003.

[156] W. Caesarendra, G. Niu, B.S. Yang, "Machine condition prognosis based on sequential Monte Carlo method," *Expert Systems with Applications*, vol. 37, pp. 2412–2420, 2010.

[157] S. Calderara, A. Prati, R. Cucchiara, "HECOL: homography and epipolar-based consistent labeling for outdoor park surveillance," *Computer Vision and Image Understanding (CVIU)*, vol. 111, no. 1, pp. 21–42, 2008.

[158] R. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision," *Cambridge University Press*, edition 2, 2004.

[159] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 22, no. 11, pp. 1330-1334, 2000.

[160] J.Y. Bouguet, "Camera calibration toolbox for MATLAB," *Computational Vision at California Institute of Technology*, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[161] E.W. Weisstein, "Perspective," *MathWorld–A Wolfram Web Resource*, http://mathworld.wolfram.com/Perspective.html.

[162] E. Auvinet *et al.*, "Multiple cameras fall dataset," *Technical Report*, no. 1350, Department of Computer Science and Operations Research (DIRO), University of Montreal, 2010.

[163] D.H. Hung, H. Saito, "Fall detection with two cameras based on occupied area," *Japan-Korea Joint Workshop on Frontiers in Computer Vision (FCV)*, pp. 33–39, 2012.

[164] X. Ma *et al.*, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[165] B. Kwolek, M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer Methods and Programs in Biomedicine*, vol. 117, no. 3, pp. 489–501, 2014.

[166] A.K. Bourke, J.V. O'Brien, G.M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[167] N.A. Macmillan, C.D. Creelman, "Detection Theory: A User's Guide," *Taylor & Francis*, 2004.

[168] G. Yu, Z. Liu, J. Yuan, "Discriminative orderlet mining for real-time recognition of human-object interaction," *Asian Conference on Computer Vision (ACCV)*, 2014.

[169] J. Wang, *et al.*, "Mining actionlet ensemble for action recognition with depth cameras," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[170] L. Xia, J.K. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[171] X. Yang, Y. Tian, "EigenJoints-based action recognition using Naive-Bayes-Nearest-Neighbor," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[172] M. Zanfir, M. Leordeanu, C. Sminchisescu, "The moving pose: an efficient 3d kinematics descriptor for low-latency action recognition and detection," *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[173] J. Wang *et al.*, "Mining actionlet ensemble for action recognition with depth cameras," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[174] H. Grabner, H. Bischof, "On-line boosting and vision," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[175] S. Stalder, H. Grabner, L. Van Gool, "Beyond semi-supervised tracking: tracking should be as simple as detection, but not simpler than recognition," *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2009.

[176] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 25, no. 5, pp. 564–577, 2003.

[177] K. Zhang, L. Zhang, M.H. Yang. "Real-time compressive tracking," *European Conference on Computer Vision (ECCV)*, 2012.

[178] F. Porikli, P. Tuzel, P. Meer, "Covariance tracking using model update based on Lie algebra," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[179] P.M. Roth *et al.*, "Online learning of pedestrian detectors by co-training from multiple cameras," *Multi-Camera Networks: Principles & Applications*, pp.313–334, Academic Press, 2009.

Part II

Included papers