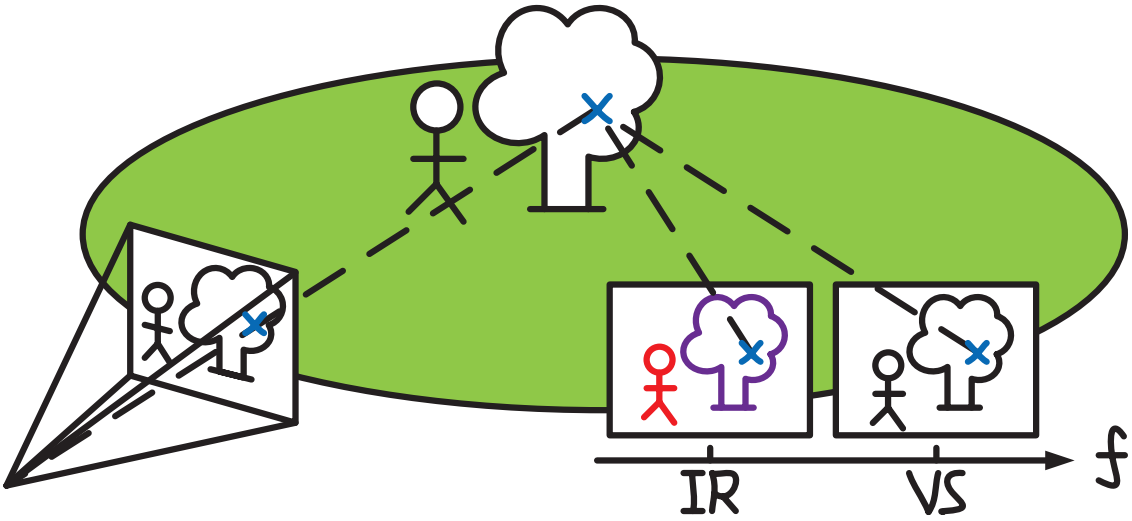
CHALMERS




# Visual Object Tracking and Classification Using Multiple Sensor Measurements

YIXIAO YUN

Department of Signals and Systems
Signal Processing Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2013

Thesis for the degree of Licentiate of Engineering

# Visual Object Tracking and Classification Using Multiple Sensor Measurements

Yixiao Yun


CHALMERS

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

Gothenburg 2013

Yun, Yixiao
Visual Object Tracking and Classification Using Multiple Sensor Measurements.

Department of Signals and Systems
Technical Report No. R021/2013
ISSN 1403-266X

Signal Processing Group
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone: + 46 (0)31-772 1837
Email: yixiao@chalmers.se



This thesis has been prepared using LaTeX.

Printed by Chalmers Reproservice,
Gothenburg, Sweden, November 2013.

To my parents and Jielin

# Abstract

Multiple sensor measurements have gained in popularity for computer vision tasks such as visual object tracking and visual pattern classification. The main idea is that multiple sensors may provide rich and redundant information, due to wide spatial or frequency coverage of the scene, which are advantageous over single sensor measurements in learning object model/feature and inferring target state/attribute in complex scenarios.

This thesis mainly addresses two problems, both exploiting multiple sensor measurements. One is video object tracking through occlusions using multiple uncalibrated cameras with overlapping fields of view, the other is multi-class image classification through sensor fusion of visual-band and thermal infrared (IR) cameras.

Paper A proposes a multi-view tracker in an alternate mode with online learning on Riemannian manifolds by cross-view appearance mapping. The mapping of object appearance between views is achieved by projective transformations that are estimated from warped vertical axis of tracked object by combining multi-view geometric constraints. A similarity metric is defined on Riemannian manifolds, as the shortest geodesic distance between a candidate object and a set of mapped references from multiple views. Based on this metric, a criterion of multi-view maximum likelihood (ML) is introduced for the inference of object state.

Paper B proposes a visual-IR fusion-based classifier by multi-class boosting with sub-ensemble learning. In our scheme, a multi-class AdaBoost classification framework is presented where information obtained from visual and thermal IR bands interactively complement each other. This is accomplished by learning weak hypotheses for visual and IR bands independently and then fusing them as learning a sub-ensemble.

Proposed methods are shown to be effective and have improved performance compared to previous approaches that are closely related, as demonstrated through experiments based on real-world datasets.

**Keywords**: visual object tracking, visual pattern classification, multiple sensor measurements, sensor fusion, multiple view geometry, Riemannian manifold, boosting

# List of Publications

**This thesis is based on the following appended publications**

**Paper A**
Y. Yun, I.Y.H. Gu, H. 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, Jun. 2013.

(Part of this paper is also presented in)
Y. Yun, I.Y.H. Gu, H. Aghajan, "Maximum-likelihood object tracking from multi-view video by combining homography and epipolar constraints," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–6, Hong Kong, Oct. 30 - Nov. 2, 2012.

**Paper B**
Y. Yun, I.Y.H. Gu, "Multi-view face pose classification by boosting with weak hypothesis fusion using visual and infrared images," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1949–1952, Kyoto, Japan, Mar. 25 - 30, 2012.

(Part of this paper is also presented in)
Y. Yun, I.Y.H. Gu, "Image classification by multi-class boosting of visual and infrared fusion with applications to object pose recognition," in *Proceedings of Swedish Symposium on Image Analysis (SSBA)*, Gothenburg, Sweden, Mar. 14 - 15, 2013.

**Other publications by the author, omitted in the thesis**

Y. Yun, I.Y.H. Gu, J. Provost, K. Åkesson, "Multi-view hand tracking using epipolar geometry-based consistent labeling for an industrial application," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, Palm Springs, California, USA, Oct. 29

- Nov. 1, 2013.

M.H. Changrampadi, Y. Yun, I.Y.H. Gu, "Multi-class ada-boost classification of object poses through visual and infrared image information fusion," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, pp. 2865–2868, Tsukuba, Japan, Nov. 11 - 15, 2012.

# Contents

# Part I

# Introductory chapters

# Chapter 1

# Introduction

With the rapid technology advancement of optical electronics and data storage over the past few decades, digital camera sensors have become ubiquitous, leading to the rise of image/video signal processing and analysis. Computer vision can be considered as a subset of image/video signal processing and analysis, and the principle is mainly based on machine learning and pattern recognition techniques. Tracking and classification of visual objects have been two important tasks within the field of computer vision.

In the context of computer vision, where the sensors are cameras, object tracking is concerned with the estimation of the position and shape of objects in the image plane, as they change poses and move around the scene [1], while pattern classification generally aims to assign each input image or video that contains an event or a scene to one of a given set of classes, taking into account their statistical variation [2]. These two subjects have attracted a great deal of research interest in recent years, largely driven by their real-world applications, for example, video surveillance in public areas such as airports and banks, human-computer interaction (HCI), traffic safety such as monitoring of driver attentiveness, ambient intelligence, and computer-assisted elderly care.

Designing an effective and robust visual object tracking or classification system is far from being a simple task due to a variety of challenges and constraints. Commonly encountered difficulties include illumination variance, background clutter, occlusions, and real time constraint. Additionally, for a reliable tracker, complex object shape and motion need to be accommodated, and for an accurate classifier, intra-class variation needs to be addressed. Despite much effort that is made and numerous methods that are proposed in last decades, achieving improved performance for a guaranteed effectiveness and robustness of trackers and classifiers remains an open issue.

# 1 Addressed Problems

This thesis mainly addresses two problems. One is video object tracking under occlusion, the other is multi-class image classification, both using multiple sensor measurements, summarized as follows.

- Tracking of visual objects containing long-term partial/full occlusions or frequent intersections, where multiple uncalibrated cameras with overlapping fields of view are exploited. The assumptions are that targets are visible in at least one camera view and move uprightly on a common planar ground that may induce a homography relation between views, as shown in Fig. 1.1.

**Figure 1.1:** Example of a three-camera case with overlapping fields of view, where a dominating ground plane is present [61].

- Classification of multi-class visual objects ranging from face poses (Fig. 1.2) to various activities (Fig. 1.3). The emphasis is on the methodologies, with applications to the above two applications (preliminary studies are performed for the 2nd applicational scenarios [60]). For face pose classification addressed in this thesis, it also includes sensor fusion where visual-band and thermal infrared (IR) cameras are employed. That is, for each face pose to be classified, a pair of visual-band and thermal IR images is captured. The class labels for face poses are defined as front, left, right, up and down, as shown in Fig. 1.2.

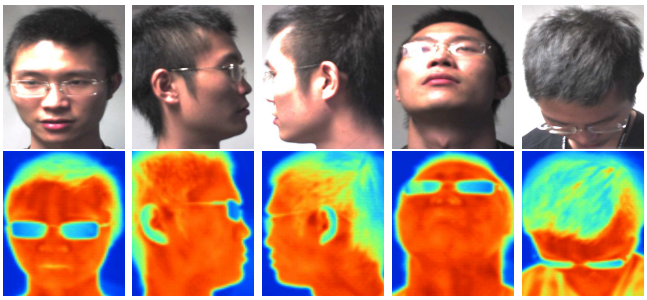


**Figure 1.2:** Example of face images in visual (1st row) and thermal infrared (2nd row) bands with five poses.

**Figure 1.3:** Example images of human activities in different classes [62]. From left to right columns: drinking, eating, reading, phone calling, using laptop, vacuum cleaning, and playing guitar.

# 2 State of the Art: an Overview

This section gives a brief introduction of state-of-the-art techniques in visual object tracking and classification that are relevant to the addressed problems in this thesis, respectively.

## 2.1 Visual Object Tracking with Occlusion Handling

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.

Many existing approaches deal with occlusions in a single camera view. Wu *et al.* [3] employ a dynamic Bayesian network which accommodates an extra hidden process for occlusion to cope with occlusions. Huang and Essa [4] represent and estimate occlusion relationships between objects by using hidden variables of depth ordering of objects towards the camera. Pan and Hu [5] analyze occlusion by exploiting spatio-temporal context information and indicate occluded pixels by template matching. Amezquita *et al.* [6] 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 [7] propose to track occluded objects by segmenting them into visible and occluded parts based on graph cuts. Chao *et al.* [8] 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.* [9] 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 [10–12], 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.

On the other hand, object tracking using multiple cameras has drawn growing interest in recent years [13], 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 [13]. 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 [14] 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 [15] 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.* [16] 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.* [17] map object trajectories across different views by registering multiple cameras via series of concatenated homography matrices (or, projective transformations). Wang *et al.* [18] and Fan *et al.* [19] 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.* [20] 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.* [21] use ground plane homography and Qu *et al.* [22] use epipolar geometry. Sankaranarayanan and Chellappa [23] 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 [24] 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.* [25] 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 [26] 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 [25] [26] 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.* [27] 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.

## 2.2 Multi-Class Visual Object Classification and Sensor Information Fusion

Multi-class problem for visual object classification is the task of classifying visual objects in images and videos into more than two classes. Commonly in vision-based human-computer interactions (HCI), the objects of interest range from face poses, facial expressions, hand gestures, to various human activities. As one of the most important HCI tasks, multi-view face pose classification is a problem that classifies face images containing out-of-plane pose changes into different classes. It can be considered as a special case of face pose estimation, as essentially pose change is a continuous process whereas classification produces discrete class labels. Nevertheless, it is still a challenging problem due to high intra-class variation, such as hairstyle, eye glasses and facial expressions, besides aforementioned challenges including varied illumination, background clutter and occlusions.

Several face pose classification methods have been proposed and developed recently. Commonly formulated as a multi-class classification problem, many approaches focus on designing the structure or framework of face pose classifiers. For example, Guo *et al.* [28] use PCA-based face features and soft margin AdaBoost to detect the frontal views. Baluja *et al.* [29] extract features inspired by [30] and builds five separate AdaBoost classifiers for face images in each class. Huang *et al.* [31] present a nested cascade detector for face poses in 5 classes using confidence-rated AdaBoost [32] based on Haar-like features. Yang *et al.* [33] introduce a tree-structured classifier for face poses in 7 classes, and each node is a three-class classifier trained by AdaBoost.MH. Islam *et al.* [34] suggest a subspace learning approach

for feature extraction and classifies five different face poses by k-NN technique. Good results have been achieved, however, these methods mainly adopt one-against-all or one-against-one strategies for multi-class problems, so model complexities may be increased.

According to feature types used, existing methods for face pose classification can be roughly categorized into facial geometry-based and appearance-based methods. The facial geometry-based methods utilize the location of facial feature points such as eye corners, mouth corners, and nose tip to determine face pose from their relative configuration [35]. These methods are efficient and effective given accurate detection of facial feature points, but on the other hand are very sensitive to the detection accuracy. Moreover, the robustness depends on the condition that the configuration of facial feature points does not change significantly under different facial expressions. The appearance-based methods exploit salient features from face appearance. The basic idea of these methods is to construct a feature subspace that efficiently describes face pose only while ignoring other sources of image variations. For instance, Raytchev *et al.* [36] and Li *et al.* [37] each extract a low-dimensional feature representation for face pose classification by isometric feature mapping (Isomap) and independent component analysis (ICA), respectively.

To improve the classification of objects, approaches are proposed on fusion of multi-modal observations, e.g., visual and infrared information. Hanif and Ali [38], Ulusoy and Yuruk [39], Wang and Li [40] each present a fusion method at the sensor level. Neagoe *et al.* [41] use decision fusion of neural classifiers for real time face recognition. Apatean et al. [42] introduce fusion scheme at different levels for SVM-based obstacle classification. These methods usually combine multiple individual features or decisions in a one-off manner, however, the interactive relations between visual and infrared observations during the learning process are seldom considered. Barbu *et al.* [43] propose a fusion scheme in original AdaBoost framework, where multi-modal information interacts through boosting iterations. Though effective, the drawback is that this method is limited to binary classes, and the simple fusion is given without problem formulation through a proper cost function.

# 3 Motivations

The main purposes for carrying out the studies on the aforementioned problems are summarized as follows:

- Conduct two aspects of studies.

  One is on the methodology part, by extending theories and methods on machine learning and manifold learning. The other is on the application

part, by using real-world measurement data for visual object tracking and classification.

- Exploit the benefits of using multiple sensors for vision tasks.

  For the tracking problem addressed in this thesis, multiple camera sensors are distributed with overlapping fields of view, providing a wide spatial coverage of the scene. For the classification problem addressed in this thesis, observations of face poses from different frequency bands using visual-band and thermal IR cameras are obtained. Hence, multiple sensor measurements may provide rich and redundant information, which are advantageous over single sensor measurements in learning object features/model and inferring target state/attribute in complex scenarios.

- Investigate the combination and interaction of multiple sensor measurements.

  For this purpose, multi-camera object tracking is mainly involved with collaborative networking of camera sensors in modeling object appearances and inferring object states based on multiple view geometry, while classification with sensor fusion is mostly concerned with complementary encoding of multi-modal information and its interactions with underlying classifier framework.

- Extend and improve existing methods in visual object tracking and classification.

  The aim is to extend previous endeavors devoted to build effective and robust tracking and classification systems, and to show how multiple sensor measurements can be integrated to generate improved results.

# 4 Outline of this Thesis

The thesis is divided into two parts. The first part briefly introduces the background and the proposed work. The second part includes publications resulted from this thesis.

The first part of the thesis is organized as follows: Chapter 2 briefly reviews related theories and work for tracking and classification. Chapter 3 makes a summary of the proposed methods, followed by Chapter 4 on conclusion and possible future work.

# Chapter 2

# Review of Related Work

This chapter briefly reviews the underlying theories and previous work that are closely related to the addressed problems in this thesis, upon which the proposed methods are built. It begins with revisiting related methods for our manifold-based multi-camera tracker in Section 1, 2 and 3, followed by a review of related techniques for our fusion-based multi-class classifier in Section 4.

Section 1 describes sequential Bayesian estimation [44] and particle filtering as an approximation to its general solution, emphasizing on sequential importance sampling (SIS) with re-sampling algorithm [45]. Section 2 introduces the geometry of Riemannian manifolds [46] [47] [48] and the region covariance descriptor [51]. Section 3 describes three geometrical constraints that are commonly used in multiple view geometry for computer vision tasks [55] [13], and how their combination relates the vertical axes of a multi-view object [27]. Section 4 introduces AdaBoost algorithms [56] and their relationship to Support Vector Machines (SVM) [57], with emphasis on a true multi-class solution *SAMME* [59].

## 1 Bayesian Tracking Using Particle Filters

### 1.1 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 $t$ [44]. Three common criteria for estimating state $\mathbf{x}_t$ are:

- Minimum mean square error (MMSE):

$$\hat{\mathbf{x}}_t^{\text{MMSE}} = \arg\min_{\hat{\mathbf{x}}} \mathbb{E}[\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2|\mathbf{y}_{1:t}] = \mathbb{E}[\mathbf{x}_t|\mathbf{y}_{1:t}] \tag{2.1}$$

- Maximum likelihood (ML):

$$\hat{\mathbf{x}}_t^{\mathrm{ML}} = \arg\max_{\hat{\mathbf{x}}} p(\mathbf{y}_{1:t}|\hat{\mathbf{x}}_t) \tag{2.2}$$

- Maximum a posteriori (MAP):

$$\hat{\mathbf{x}}_t^{\mathrm{MAP}} = \arg\max_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}}_t|\mathbf{y}_{1:t}) \tag{2.3}$$

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

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_{1:t})} \\ &= \frac{p(\mathbf{y}_t,\ \mathbf{y}_{1:t-1}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_t,\ \mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\ \mathbf{x}_t)p(\mathbf{y}_{1:t-1}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\ \mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})p(\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})p(\mathbf{x}_t)} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \end{aligned} \tag{2.4}$$

The second term in the numerator of (2.4) can be further expanded by marginalizing over the previous state $\mathbf{x}_{t-1}$:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) &= \int p(\mathbf{x}_t,\ \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1},\ \mathbf{y}_{1:t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \end{aligned} \tag{2.5}$$

The denominator of (2.4) is the normalization constant

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) &= \int p(\mathbf{y}_t,\ \mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t \\ &= \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t \end{aligned} \tag{2.6}$$

Combining (2.4), (2.5) and (2.6) yields

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \tag{2.7}$$

which is the recursive formula for Bayesian estimation. As shown in (2.7), 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 $p(\mathbf{y}_t|\mathbf{x}_t)$.

## 1.2 Particle Filtering

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

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} \delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(i)}) \tag{2.8}$$

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

It is usually impossible to sample from the true posterior pdf. Instead, a *proposal distribution* denoted by $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ is used, and the weights are defined as

$$\omega_t^{(i)} = \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} \tag{2.9}$$

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

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) \tag{2.10}$$

Similar to the derivation steps in (2.4), the posteriori $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ can be factorized as

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) \frac{p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \tag{2.11}$$

Plugging (2.10) and (2.11) into (2.9) yields

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})} \tag{2.12}$$

Based on *Markov assumption*, (2.12) is modified to

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})} \tag{2.13}$$

Using (2.13), expression to approximate the posterior pdf can be written as

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

For sequential importance sampling (SIS), it is commonly assumed that the proposal distribution is the state transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, so particle weights are updated by

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) \tag{2.14}$$

followed by weight normalization.

To mitigate degeneracy phenomenon, re-sampling is performed according to the criterion based on *effective sample size* $N_{eff}$ [44], when its estimate $\hat{N}_{eff}$ is found below a threshold $N_T$:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_t^{(i)})^2} < N_T \tag{2.15}$$

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, (2.14) can be further simplified as

$$\omega_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) \tag{2.16}$$

The pseudo code for SIS with re-sampling is summarized in Table 2.1.

**Table 2.1:** Pseudo code for SIS with re-sampling [44].

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 (2.14).

(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** (2.15): 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 Riemannian Geometry and Region Covariance

Manifold-based object representation has been used by several visual object tracking methods [10] [11] [12] [54]. It gives a low-dimensional description of objects, and efficiently describes object dynamics by a nonlinear smooth manifold, since different object appearances relating to out-of-plane pose changes lie on the same manifold. Online learning of object appearance can be performed on the manifolds for reducing tracking drifts caused by various appearance and pose changes, taking into account the underlying geometry of that manifold space.

## 2.1 Manifold of Symmetric Positive Definite Matrices

A manifold is a topological space as low dimensional subspaces embedded in a high dimensional space, which is only locally Euclidean. Fig. 2.1 depicts a two-dimensional manifold embedded in $\mathbb{R}^3$. Our notation follows that in [10]: $\mathbf{P}$ is the starting point and $\mathbf{Q}$ is the end point on a manifold $\mathcal{M}$, and $\mathbf{\Delta}$ is the velocity vector in the tangent space $\mathcal{T}$. A Riemannian manifold is a differentiable manifold where the tangent space at each point has an inner product $<,>_{\mathbf{P}\in\mathcal{M}}$ that varies smoothly from point to point.

The space of $l \times l$ symmetric positive definite ($Sym_l^+$) matrices lies on a Riemannian manifold that constitutes the convex-half cone in a vector space of matrices [10]. To compute statistics on $Sym_l^+$, *affine-invariant* metric [49] and *log-Euclidean* metric [50] are commonly used, leading to similar numerical results. The log-Euclidean metric is adopted in this paper since it is computationally more efficient [50].

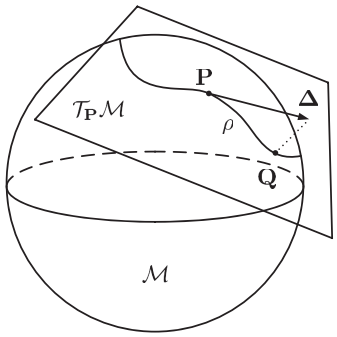



**Figure 2.1:** Example of 2-D manifold $\mathcal{M}$ embedded in $\mathbb{R}^3$. $\mathbf{P}$ and $\mathbf{Q}$ are manifold points, $\mathcal{T}_{\mathbf{P}}\mathcal{M}$ is the tangent space for $\mathbf{P}$, $\mathbf{\Delta}$ is the tangent vector whose projected point on the manifold is $\mathbf{Q}$. The geodesic $\rho$ is the minimum length curve between $\mathbf{P}$ and $\mathbf{Q}$ on the manifold.

As shown in Fig. 2.1, *Exponential map* ($\mathcal{T} \mapsto \mathcal{M}$) is the function that maps a tangent vector $\mathbf{\Delta}$ along the geodesic $\rho$ to a point $\mathbf{Q}$ on the manifold $\mathcal{M}$, given by

$$\exp_{\mathbf{P}}(\mathbf{\Delta}) = \exp(\log(\mathbf{P}) + \mathbf{\Delta}) = \mathbf{Q} \tag{2.17}$$

under the log-Euclidean metric, and

$$\exp_{\mathbf{P}}(\mathbf{\Delta}) = \mathbf{P}^{\frac{1}{2}} \exp(\mathbf{P}^{-\frac{1}{2}} \mathbf{\Delta} \mathbf{P}^{-\frac{1}{2}}) \mathbf{P}^{\frac{1}{2}} = \mathbf{Q} \tag{2.18}$$

under the affine invariant metric, where $\mathbf{P}$ is the starting point on $\mathcal{M}$.

Conversely, *Logarithmic map* ($\mathcal{M} \mapsto \mathcal{T}$) is the function that maps a manifold point $\mathbf{Q}$ (start from $\mathbf{P}$) to a vector $\mathbf{\Delta}$ in the tangent space $\mathcal{T}_{\mathbf{P}}$. It corresponds to a velocity vector, and is given by

$$\log_{\mathbf{P}}(\mathbf{Q}) = \log(\mathbf{Q}) - \log(\mathbf{P}) = \mathbf{\Delta} \tag{2.19}$$

under the log-Euclidean metric, and

$$\log_{\mathbf{P}}(\mathbf{Q}) = \mathbf{P}^{\frac{1}{2}} \log(\mathbf{P}^{-\frac{1}{2}} \mathbf{Q} \mathbf{P}^{-\frac{1}{2}}) \mathbf{P}^{\frac{1}{2}} = \mathbf{\Delta} \tag{2.20}$$

under the affine invariant metric.

*Geodesic* is the shortest curve $\rho$ between two points $\mathbf{P}$, $\mathbf{Q}$ on a manifold $\mathcal{M}$. The geodesic distance is the length of $\rho$ given by

$$d(\mathbf{P}, \mathbf{Q}) = \| \log_{\mathbf{P}}(\mathbf{Q}) \| = \| \log(\mathbf{Q}) - \log(\mathbf{P}) \| \tag{2.21}$$

under the log-Euclidean metric, and

$$d(\mathbf{P}, \mathbf{Q}) = \mathrm{tr}[\log^2(\mathbf{P}^{-\frac{1}{2}} \mathbf{Q} \mathbf{P}^{-\frac{1}{2}})] \tag{2.22}$$

under the affine invariant metric. Another alternative for computing the geodesic distance [51] is

$$d(\mathbf{P}, \mathbf{Q}) = \sqrt{\sum_{k=1}^{n} \ln^2 \lambda_k(\mathbf{P}, \mathbf{Q})} \tag{2.23}$$

where $\{\lambda_k(\mathbf{P}, \mathbf{Q})\}_{k=1}^{n}$ are the generalized eigenvalues of $\mathbf{P}$ and $\mathbf{Q}$.

## 2.2 Region Covariances as Object Descriptors

A region covariance matrix [51] enables an effective description of object features, and is shown to be robust and versatile to variations in illuminations, views and poses at modest computational cost by using integral images.

Given a rectangular image region $\mathcal{R}$, let $\{\mathbf{f}_k\}_{k=1}^{|\mathcal{R}|}$ be $l$-dimensional feature vectors for each pixel inside $\mathcal{R}$, 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 example, a feature vector can be formed as

$$\mathbf{f}_k = [x, y, r, g, b, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|, \sqrt{I_x^2 + I_y^2}, \arctan(\frac{I_y}{I_x})]^T \tag{2.24}$$

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 example of feature vector can be [10]

$$\mathbf{f}_k = [x, y, I, I_g^1, \cdots, I_g^M]^T \tag{2.25}$$

where $(x, y)$ is the pixel coordinate, $I$ is the image intensity, and $I_g^m$, $m = 1, \cdots, M$ are filtered images from 2-D Gabor filters of different orientations and frequencies [52]. The region $\mathcal{R}$ is described by a $l \times l$ covariance matrix

$$\mathbf{C}_\mathcal{R} = \frac{1}{|\mathcal{R}| - 1} \sum_{k=1}^{|\mathcal{R}|} (\mathbf{f}_k - \boldsymbol{\mu})(\mathbf{f}_k - \boldsymbol{\mu})^T \tag{2.26}$$

where $\boldsymbol{\mu}$ is the sample mean. Since covariance matrices $\mathbf{C}_\mathcal{R} \in Sym_l^+$, they may be viewed as connected points on a Riemannian manifold [53] [54].

# 3 Multiple View Geometry for Vision Tasks

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. 2.2). 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 [27].

## 3.1 Planar Homography

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 $\Pi$ 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 $\Pi$, $\mathbf{x}_1' \neq \mathbf{H}^{ij}\mathbf{x}_1$ (see Fig. 2.2). Homography is not sufficient for warping the vertical axis of object, additional geometric constraints should be added.

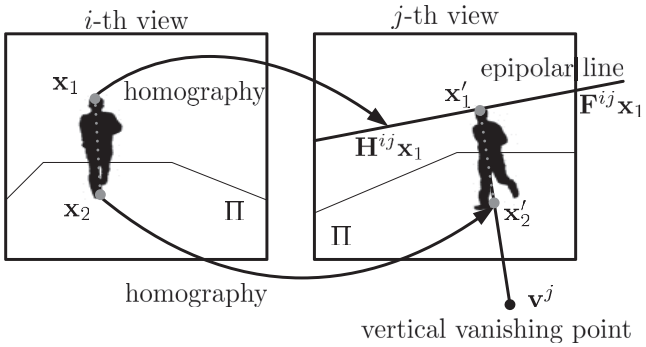



**Figure 2.2:** Warping vertical axis of a tracked object from $i$-th view to $j$-th view by combining the constraints of planar homography, epipolar geometry and vertical vanishing point [27].

### 3.2 Epipolar Geometry

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 $\mathbf{x}'_1\mathbf{F}^{ij}\mathbf{x}_1 = 0$. 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. 2.2). Thus, the epipolar geometry constrains the corresponding points that lie on the conjugate pairs of epipolar lines.

### 3.3 Vertical Vanishing Point

To obtain the warped axis inclination, the vertical vanishing point $\mathbf{v}^j$ of $j$-th view is used. As depicted in Fig. 2.2, 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 [55].

Using the same procedure, the vertical axis of tracked object in the $j$-th view may be warped to the $i$-th view.

## 4 Boosting and Multi-Class AdaBoost

All classification algorithms are based on the assumption that the data in question holds one or more features, each of which belongs to one of several distinct and exclusive classes.

Classification algorithms typically include two successive procedures: training and testing. In the initial training phase, a unique description of each class is made by learning with typical features extracted from the

training samples and separating them in the feature space. In the subsequent testing phase, these feature space separations are used to classify newly input feature vectors extracted from the testing dataset. Therefore, the classification problem can also be viewed as determining to which subspace class each feature vector belongs.

One of the most popular techniques in machine learning is AdaBoost. It is widely used for object recognition and classification, due to its outstanding performance and the ease to use. Moreover, its capabilities to automatically select the most relevant feature descriptors from large feature sets are also often exploited.

AdaBoost, short for Adaptive Boosting, is a technique which can be used to improve the performance of many learning algorithms [56]. Generally, AdaBoost sequentially applies a given learning algorithm with respect to a set of training samples and adds each prediction to an ensemble. When being added to the emsemble, the prediction is typically weighted according to its accuracy. After this, the dataset is also reweighted: samples that are misclassified gain weights and samples that are correctly classified lose weights. Thereby each successive classifier is forced to focus on those samples that are misclassified by previous ones in the sequence. AdaBoost is chosen in this thesis work since its basic idea is quite simple but still very successful, with performances comparable to more complex methods such as Support Vector Machines (SVM) [57].

## 4.1 Conventional AdaBoost

In fact, AdaBoost is originally intended only for boosting binary classifiers, so it can not be directly applied to multi-class cases. For multi-class problems, there are many extensions and modifications of AdaBoost, but all derive from the same kind of model, that is, the forward stagewise additive modeling.

Forward stagewise additive modeling approaches the optimization problem by sequentially adding new basis functions to the expansion without adjusting the parameters and coefficients of those that have been already added. AdaBoost is equivalent to this model and it uses the exponential loss function below for binary case:

$$L(y, f(\boldsymbol{x})) = \exp(-yf(\boldsymbol{x})) \tag{2.27}$$

For binary AdaBoost, training samples are input as feature vectors $\{\boldsymbol{x_i}\}$ with their desired outputs $\{y_i\} \in \{-1, 1\}$, where $i = 1, 2, \cdots, N$, the basis functions in the forward stagewise additive model are the weak learners $T^{(m)}(\boldsymbol{x}) \in \{-1, 1\}$. Using the exponential loss function, the problem becomes solving:

$$(\alpha^{(m)}, T^{(m)}) = \arg\min_{\alpha,T} \sum_{i=1}^{N} \exp[-y_i(f^{(m-1)}(\boldsymbol{x}_i) + \alpha T(\boldsymbol{x}_i))] \tag{2.28}$$

for the weak learner $T^{(m)}$ and its corresponding weight coefficient $\alpha^{(m)}$ to be added at each step. This can expressed as:

$$(\alpha^{(m)}, T^{(m)}) = \arg\min_{\alpha,T} \sum_{i=1}^{N} w_i^{(m)} \exp(-\alpha y_i T(\boldsymbol{x}_i)) \tag{2.29}$$

where

$$w_i^{(m)} = \exp(-y_i f^{(m-1)}(\boldsymbol{x}_i)). \tag{2.30}$$

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

It can be easily observed that

$$\begin{cases} \text{If } y_i = T(\boldsymbol{x}_i), & \text{then } y_i \cdot T(\boldsymbol{x}_i) = 1; \\ \text{If } y_i \neq T(\boldsymbol{x}_i), & \text{then } y_i \cdot T(\boldsymbol{x}_i) = -1. \end{cases} \tag{2.31}$$

Therefore, the criterion in (2.29) can be expressed as

$$e^{-\alpha} \sum_{y_i = T(\boldsymbol{x}_i)} w_i^{(m)} + e^{\alpha} \sum_{y_i \neq T(\boldsymbol{x}_i)} w_i^{(m)}, \tag{2.32}$$

which in turn can be rewritten as

$$(e^{\alpha} - e^{-\alpha}) \sum_{i=1}^{N} w_i^{(m)} \mathbb{I}(y_i \neq T(\boldsymbol{x}_i)) + e^{-\alpha} \sum_{i=1}^{N} w_i^{(m)} \tag{2.33}$$

Apply gradient descent method to (2.33) and solve for $\alpha$, by taking partial derivative respect to $\alpha$ and set the resulting equation is to 0, one obtain $\alpha$ as

$$\alpha^{(m)} = \frac{1}{2} \log \frac{1 - err^{(m)}}{err^{(m)}} \tag{2.34}$$

where $err^{(m)}$ is the minimized weighted error rate

$$err^{(m)} = \frac{\sum_{i=1}^{N} w_i^{(m)} \mathbb{I}(y_i \neq T^{(m)}(\boldsymbol{x}_i))}{\sum_{i=1}^{N} w_i^{(m)}}. \tag{2.35}$$

As a result, the approximation can be updated as

$$f^{(m)}(\boldsymbol{x}) = f^{(m-1)}(\boldsymbol{x}) + \alpha^{(m)} T^{(m)}(\boldsymbol{x}). \tag{2.36}$$

So the weight for the next iteration can be accordingly updated as

$$w_i^{(m+1)} = \exp(-y_i f^{(m)}(\boldsymbol{x}_i)) = w_i^{(m)} \cdot e^{-\alpha^{(m)} y_i T^{(m)}(\boldsymbol{x}_i)}. \tag{2.37}$$

Considering the fact that

$$-y_i T^{(m)}(\boldsymbol{x}_i) = 2 \cdot \mathbb{I}(y_i \neq T(\boldsymbol{x}_i)) - 1, \tag{2.38}$$

the updating scheme of sample weights becomes

$$w_i^{(m+1)} = w_i^{(m)} \cdot e^{\beta^{(m)} \mathbb{I}(y_i \neq T(\boldsymbol{x}_i))} \cdot e^{-\alpha(m)} \tag{2.39}$$

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

At this stage, the conventional AdaBoost algorithm can be summarized in Table 2.2.

**Table 2.2:** Algorithm summary of conventional AdaBoost [56].

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

When AdaBoost is asked to classify a previously unknown sample, each classifier in the ensemble contributes its own weight $\beta^{(m)}$ to either one of the two classes it predicts, and in the end, the class with the higher value is chosen as the final prediction.

During each boosting round, the weights of wrongly classified samples are increased. In this way, the weak learner for the next boosting round will be forced to pay attention to those misclassified. When combining the predictions after each boosting round, the training error rate is thus decreased to some extent.

Eventually, the training error rate is significantly reduced by AdaBoost, where each weak learner are combined together in a smart way, that is, assigning weights to each prediction made by the weak learners according to their accuracy.

It should be noted that the weight for each classifier in the ensemble should be a positive value, that is,

$$\beta^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} > 0. \tag{2.40}$$

The solution to this inequality is

$$0 < err^{(m)} < 0.5, \tag{2.41}$$

so each weak learner must have an accuracy greater than 50%, otherwise the weight distribution for the training dataset would not be updated or to be updated towards the wrong direction thus causing AdaBoost out of work. This is also the reason why conventional AdaBoost algorithm can easily fail to work when facing multi-class classification problems that are more complicated than binary cases.

## 4.2 Relationship to Support Vector Machines

From the perspective of margin theory, there is a strong connection between boosting and SVM [56]. For AdaBoost, the margin of sample $(\boldsymbol{x}, y)$ is defined to be

$$\frac{y \sum_m \beta^{(m)} T^{(m)}(\boldsymbol{x})}{\sum_m \beta^{(m)}} \tag{2.42}$$

It is a number in $[-1, 1]$ which is positive if and only if the sample is correctly classified. Moreover, the magnitude of the margin can be interpreted as a measure of confidence in the prediction.

Using the notation and the definition of margin given in (2.42), the goal of maximizing the minimum margin in AdaBoost can be written as

$$\max_{\boldsymbol{\beta}} \min_i \frac{(\boldsymbol{\beta} \cdot \mathbf{T}(\boldsymbol{x_i}))y_i}{\|\boldsymbol{\beta}\| \|\mathbf{T}(\boldsymbol{x}_i)\|} \tag{2.43}$$

where the norms in the denominator are defined as:

$$\|\boldsymbol{\beta}\|_1 \doteq \sum_m \beta^{(m)}, \qquad \|\mathbf{T}(\boldsymbol{x})\|_2 \doteq \max_m |T^{(m)}(\boldsymbol{x})| \tag{2.44}$$

In comparison, the goal of SVM is to maximize a minimal margin of the form described in (2.43), but where the norms are instead Euclidean:

$$\|\boldsymbol{\beta}\|_2 \doteq \sqrt{\sum_m (\beta^{(m)})^2}, \qquad \|\mathbf{T}(\boldsymbol{x})\|_\infty \doteq \sqrt{\sum_m (T^{(m)}(\boldsymbol{x}))^2} \tag{2.45}$$

Thus, SVM uses the $\ell_2$ norm for both $\boldsymbol{\beta}$ and $\mathbf{T}(\boldsymbol{x})$, while AdaBoost uses the $\ell_{infty}$ norm for $\mathbf{T}(\boldsymbol{x})$ and $\ell_1$ norm for $\boldsymbol{\beta}$. In such a way, SVM and AdaBoost seem very similar. However, they differ in several aspects [56]:

- Different norms can result in very different margins.
- The computation requirements are different, where SVM corresponds to quadratic programming, while AdaBoost corresponds only to linear programming.
- AdaBoost is based on game theory and online learning, while SVM is developed under the statistical learning theory.
- A different approach is used to search efficiently in high dimensional space, where SVM employ the method of kernels and AdaBoost instead uses greedy search with weak learners.

## 4.3 Multi-Class Extensions of Conventional AdaBoost

As a result, many extensions of conventional AdaBoost to the multi-class classification problem have been designed, however, the weak classifiers are still required to have an accuracy higher than 50%. One possible and popular approach is to transform the multi-class problem into several binary subproblems, which can be done by using one-against-all or one-against-one strategy [58].

- **One-against-all strategy for each class**:

  The one-against-all strategy constructs one model for each class, where each model is trained to separate the samples of its corresponding class from the samples of all remaining classes. When a new sample of unknown class is taken in, it will be assigned to the class whose model has the maximum output value among all. In other words, each predefined class has a probabilistic binary classifier to distinguish its kind from others, and each classifier will make a class prediction for an unknown sample with some probability for that class. The class prediction of the classifier that returns the highest probability will thus be chosen for this unknown sample.

- **One-against-one strategy for all pairs of classes**:

  On the other hand, the one-against-one strategy constructs one model for each pair of classes, so for a multi-category problem with $K$ ($K > 2$) classes, $K(K-1)/2$ models in total are trained to divide the samples of one class from the samples of the other class in all pairs. When a new sample of unknown class is taken in, it will be sorted to the class with maximum voting, where each model votes for one class. This pairwise learning method may sound computational consuming, but in fact it is not, and if the classes are evenly distributed, it will be at least as fast as any other multi-class solution. The reason is that each pairwise subproblem only takes training samples of two classes into consideration, other than the whole training dataset. For example, if $N$ samples are divided evenly among $K$ classes, there will be $2N/K$ samples per subproblem. Suppose the runtime of a binary classification algorithm is proportional to the number of training samples it learns, then the total runtime will be proportional to $K(K-1)/2 \cdot 2N/K$, which is $(K-1)N$. That means, this method only scales linearly with the number of classes.

In a word, if the weak learners boosted by AdaBoost are inherently incapable of producing multi-class predictions, the above alternatives can be particularly useful.

## 4.4 Multi-Class AdaBoost

However, for this thesis work, an approach that handles multi-class cases directly without reducing them to multiple two-class problems will be used, known as multi-class AdaBoost.

As mentioned before, AdaBoost is originally designed only for boosting two-class cases. If using one-against-all or one-against-one strategies, it can be extended to solve multi-class problems. However, this still requires the weak learners to have a classification rate higher than 50%, which is quite difficult for multi-class cases, where the number of classes $K \geq 3$ and the probability for random guessing is $1/(K-1)$. As a result, the real multi-class AdaBoost algorithm, also called SAMME, is proposed in [59], which successfully avoids reducing the multi-class problems to two-class subproblems and only requires the classification rate of weak learners better than $1/(K-1)$.

In fact, SAMME algorithm is very similar to the conventional AdaBoost. It is also based on forward stagewise additive modeling using an exponential loss function. However, this time the exponential loss function has been modified into a multi-class version.

For multi-class (the number of classes $K \geq 3$) classification problem, SAMME encodes the class prediction (denoted by $c_i$) as $\boldsymbol{y}_i = (y_1, y_2, \cdots, y_K)^T$,

$i = 1, 2, \cdots, N$, with

$$y_k = \begin{cases} 1, & \text{if } c_i = k \\ -\frac{1}{K-1}, & \text{otherwise} \end{cases} \tag{2.46}$$

where $k = 1, 2, \cdots, K$.

Then if $f = (f_1, f_2, \cdots, f_K)^T$ and $\sum_{k=1}^{K} f_k = 0$, the multi-class loss optimized by SAMME is

$$L(y, f) = \exp\left(-\frac{1}{K}\boldsymbol{y}^T f\right) \tag{2.47}$$

This time, the basis functions in the forward stagewise additive model become multi-class weak learners $T^{(m)} \in \mathcal{Y}$, where

$$\mathcal{Y} = \left\{ \begin{array}{c} (1, -\frac{1}{K-1}, \cdots, -\frac{1}{K-1})^T \\ (-\frac{1}{K-1}, 1, \cdots, -\frac{1}{K-1})^T \\ \vdots \\ (-\frac{1}{K-1}, \cdots, -\frac{1}{K-1}, 1)^T \end{array} \right\} \tag{2.48}$$

Again, the problem becomes solving:

$$(\alpha^{(m)}, T^{(m)}) = \arg\min_{\alpha, T} \sum_{i=1}^{N} \exp[-\frac{1}{K}\boldsymbol{y}_i^T (f^{(m-1)}(\boldsymbol{x}_i) + \alpha T(\boldsymbol{x}_i))] \tag{2.49}$$

for the weak learner $T^{(m)}$ and its corresponding weight coefficient $\alpha^{(m)}$ to be added at each step. This can expressed as:

$$(\alpha^{(m)}, T^{(m)}) = \arg\min_{\alpha, T} \sum_{i=1}^{N} w_i^{(m)} \exp(-\frac{1}{K}\alpha \boldsymbol{y}_i^T T(\boldsymbol{x}_i)) \tag{2.50}$$

where

$$w_i^{(m)} = \exp(-\frac{1}{K}\boldsymbol{y}_i^T f^{(m-1)}(\boldsymbol{x}_i)). \tag{2.51}$$

Again, since $w_i^{(m)}$ is independent on $\alpha$ and $T(\boldsymbol{x}_i)$, it can be regarded as a weight factor that is applied to each training samples. This weight depends on $f^{(m-1)}(\boldsymbol{x}_i)$ so the weight changes during each iteration $m$.

In a similar way to the binary case, it can be obtained that

$$\begin{cases} \text{If } \boldsymbol{y}_i = T(\boldsymbol{x}_i), & \text{then } \boldsymbol{y}_i^T T(\boldsymbol{x}_i) = \frac{K}{K-1}; \\ \text{If } \boldsymbol{y}_i \neq T(\boldsymbol{x}_i), & \text{then } \boldsymbol{y}_i^T T(\boldsymbol{x}_i) = -\frac{K}{(K-1)^2}. \end{cases} \tag{2.52}$$

Therefore, the criterion in (2.50) can be expressed as

$$\exp\left(-\frac{\alpha}{K-1}\right)\cdot\sum_{\boldsymbol{y}_i=T(\boldsymbol{x}_i)}w_i^{(m)}+\exp\left(\frac{\alpha}{(K-1)^2}\right)\cdot\sum_{\boldsymbol{y}_i\neq T(\boldsymbol{x}_i)}w_i^{(m)}, \quad (2.53)$$

which in turn can be rewritten as

$$\left(e^{\frac{\alpha}{(K-1)^2}}-e^{-\frac{\alpha}{K-1}}\right)\cdot\sum_{i=1}^{N}w_i^{(m)}\mathbb{I}(\boldsymbol{y}_i\neq T(\boldsymbol{x}_i))+e^{-\frac{\alpha}{K-1}}\cdot\sum_{i=1}^{N}w_i^{(m)} \quad (2.54)$$

Apply gradient descent method to (2.54) and solve for $\alpha$, by taking partial derivative respect to $\alpha$ and set the resulting equation is to 0, one obtain $\alpha$ as

$$\alpha^{(m)}=\frac{(K-1)^2}{K}\left(\log\frac{1-err^{(m)}}{err^{(m)}}+\log(K-1)\right) \quad (2.55)$$

where $err^{(m)}$ is the minimized weighted error rate

$$err^{(m)}=\frac{\sum_{i=1}^{N}w_i^{(m)}\mathbb{I}(\boldsymbol{y}_i\neq T^{(m)}(\boldsymbol{x}_i))}{\sum_{i=1}^{N}w_i^{(m)}}. \quad (2.56)$$

As a result, the approximation of multi-class problem can be updated as

$$f^{(m)}(\boldsymbol{x})=f^{(m-1)}(\boldsymbol{x})+\alpha^{(m)}T^{(m)}(\boldsymbol{x}). \quad (2.57)$$

So the weight for the next iteration can be accordingly updated as

$$w_i^{(m+1)}=\exp(-\frac{1}{K}\boldsymbol{y}_i^T f^{(m)}(\boldsymbol{x}_i))=w_i^{(m)}\cdot\exp(-\frac{1}{K}\alpha^{(m)}\boldsymbol{y}_i^T T^{(m)}(\boldsymbol{x}_i)). \quad (2.58)$$

This is equivalent to

$$\begin{cases} w_i^{(m)}\cdot\exp(\frac{K-1}{K}\beta^{(m)}), & \text{if } c_i=T^{(m)}(\boldsymbol{x}_i));\\ w_i^{(m)}\cdot\exp(\frac{1}{K}\beta^{(m)}), & \text{if } c_i\neq T^{(m)}(\boldsymbol{x}_i)). \end{cases} \quad (2.59)$$

where

$$\beta^{(m)}=\log\frac{1-err^{(m)}}{err^{(m)}}+\log(K-1). \quad (2.60)$$

The algorithm of multi-class AdaBoost (SAMME) is summarized in Table 2.3.

**Table 2.3:** Algorithm summary of Multi-Class AdaBoost (SAMME) [59].

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

One can easily notice the extra term $\log(K-1)$ in the update scheme for classifier. It has been shown that this term derives from the forward stage-wise additive modeling which uses the multi-class exponential loss function. In additional, if $K = 2$, the algorithm returns to binary AdaBoost. Moreover, the updating of sample weights seems different from (2.59), but actually they are equal, since the one in algorithm is the normalized version.

# Chapter 3

# Summary of this Thesis Work

This chapter gives a summary of this thesis work on multi-camera tracking and multi-class classification tasks, respectively, by showing the basic ideas and big pictures, followed by the main novelties.

## 1 A Multi-Camera Tracker with Online Learning

### Basic Ideas

For the task of visual object tracking, we address issues in tracking with occlusion scenarios, where multiple uncalibrated cameras with overlapping fields of view are exploited. Although many robust single-camera trackers have been developed, challenges still remain especially when dynamic objects experience long-term partial/full occlusions, or intersections with other objects in crowded scenes. There are also many existing approaches dealing with occlusions in a single camera view. However, they can handle occlusions to some extent, but become less feasible when objects undergo long-term full occlusions. In view of this issue, multiple cameras is employed due to their multiple view coverage that is advantageous in handling complex scenarios, including full occlusions. 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.

We adopt a three-layer scheme where tracking is first done independently in each individual view then tracking results are mapped from different views to finally improve the tracking jointly. The cross-view mapping is achieved by combining multiple geometric constraints. 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. To accommodate appearance change caused by object dynamics, a method for online learning of object appearances on Riemannian manifolds is added to the tracker.

## The Big Picture

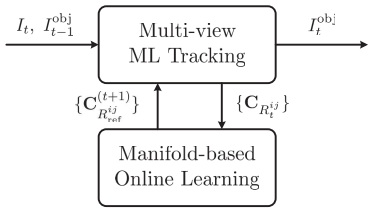



**Figure 3.1:** The block diagram of our multi-camera tracking scheme. $I_t$ is the image frame at $t$, $I_{t-1}^{\text{obj}}$ and $I_t^{\text{obj}}$ is the tracked object at $t-1$ and $t$, respectively. $\mathbf{C}_{R_t^{ij}}$ is the manifold point of tracked object, and $\mathbf{C}_{R_{\text{ref}}^{ij}}^{(t)}$ is the updated reference model.

The multi-camera tracking scheme consists of two major parts: (i) multi-view Maximum Likelihood (ML) tracking; (ii) online learning of object appearances on the Riemannian manifold. As shown in Fig. 3.1, these two process are performed in an alternative fashion.

### Tracking Part

As shown in Fig. 3.2, Tracking part is performed in three layers in cascade at each time instant. In the first layer, a single view Bayesian framework-based tracker is applied to track a candidate object from a given view. In the second layer, tracked object from each view is mapped to the remaining views by using planar homography plus other geometrical constraints. Once the correspondences between different views are established, a manifold-based maximum likelihood (ML) criterion is applied to obtain the best multi-view tracking result in the third layer.

The essence of such a multi-view tracker is to regard an object in different views as different points on a same manifold (see schematic description in Fig. 3.3). Hence, the solution of multi-view object tracking is equivalent to defining a similarity measure on the manifold, finding the best view object

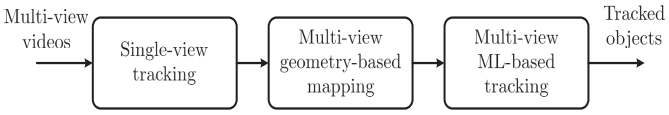



**Figure 3.2:** Three layers in the multiview tracking scheme.

under the measure for a given reference set, and mapping it to the desired view under geometrical constraints.

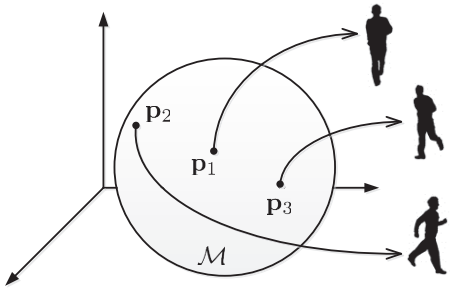



**Figure 3.3:** Description of different views of object as points on a smooth manifold.

### Online Learning Part

In order to maintain a timely reference object set as video objects may change with time, online learning of reference object set is an important issue for preventing tracking drift. However, one should also be careful of not learning the wrong objects when object partially/fully occurs. A strategy needs to be adopted to decide whether or not such an online learning should be activated at a given time instant. We propose to use a simple criterion to decide whether online learning is applied, based on the observation that changes caused by a dynamic object are usually less significant as comparing with changes caused by an occluding object. A simple measure is formed based on the histogram cross-correlation between the reference and tracked object (or, the Bhattacharyya coefficient) in combination of object mapping between different views.

## Main Novelties

The main novelties of the paper include: (a) define a similarity measure, based on geodesics between a candidate object and a set of mapped references from multiple views on a Riemannian manifold; (b) propose multi-view

maximum likelihood (ML) estimation of object bounding box parameters, based on Gaussian-distributed geodesics on the manifold; (c) introduce online learning of object appearances on the manifold, taking into account of possible occlusions; (d) utilize projective transformations for objects between views, where parameters are estimated from warped vertical axis by combining planar homography, epipolar geometry and vertical vanishing point; (e) embed single-view trackers in a three-layer multi-view tracking scheme.

# 2 A Multi-Class Classifier with Sensor Fusion

## Basic Ideas

For the task of multi-class visual object classification, we address the problem through sequential learning and sensor fusion. The strategy is to apply a two-stage ensemble learning method in a multi-class boosting structure, using images observed in visual and thermal infrared (IR) bands. A sub-ensemble is added to the learning process which combines hypotheses for both bands, with sub-ensemble weights according to their accuracies. The basic idea for introducing a sub-ensemble is that hypothesis sub-ensemble may have enhanced performance based on fusion of visual and infrared information. In this way, the final strong ensemble may have further improved accuracy.

As we observed, thermal IR images are blurred in edges and lack of texture details, which corresponds to energy concentrated on relatively lower frequency band. Viewing the special nature of thermal IR images, we suggest using Gabor wavelet features. The idea here is that a bank of Gabor wavelets with appropriately specified frequency bands and orientations is used to characterize an IR image, which may extract salient features in thermal IR images due to the spatial locality, frequency selectivity and orientation selectivity. DC component is added as a feature component covering the lower frequency band.

## The Big Picture

As shown in Fig. 3.4, our classification-fusion framework consists of three major parts: (a) independent weak hypothesis learning using visual and infrared features with the same sampling weight; (b) fusion by optimizing hypothesis sub-ensemble; (c) adding sub-ensemble to a final strong classifier and updating sampling weight distribution with a scale factor.

The essence for using the same sampling weight is to force weak classifiers for both visual and infrared bands to focus on the same objects, therefore weak hypotheses independently learned from visual and infrared features

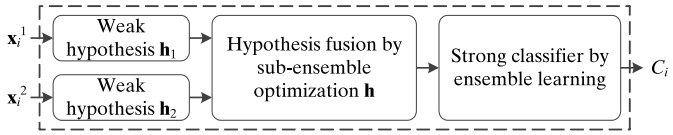



**Figure 3.4:** Block diagram of our classification-fusion scheme. The dashed box represents boosting structure. The notations $\mathbf{x}_i^1$, $\mathbf{x}_i^2$, $C_i$ denote visual features, infrared features, and predicted class labels of $i$-th object, respectively.

match each other. The basic idea for hypothesis optimization is to add hypotheses for both bands to the sub-ensemble, with sub-ensemble weights according to their accuracies, so that hypothesis sub-ensemble may have enhanced performance based on fusion of visual and infrared information. In this way, the final strong ensemble may have further improved accuracy. The main motivation for using a scale factor to update sampling weights is to make weak classifiers focus on those difficult objects misclassified in both visual and infrared bands. The main novelty lies in two-stage ensemble learning within multi-class boosting framework, by using visual and infrared information in this interactive manner, which may lead to better classification results.

## Main Novelties

The main contribution of this paper is a multi-class AdaBoost classification framework where information obtained from visual and infrared bands interactively complement each other. This is achieved by first learning weak hypotheses for visual and IR bands independently and then fusing them in sub-ensembles. In addition, an effective feature descriptor is introduced to thermal IR images.

# Chapter 4

# Conclusion and Future Work

## Conclusion

In this thesis, two new schemes are proposed for two computer vision tasks with multiple sensor measurements, respectively. One is for visual object tracking using multiple uncalibrated cameras with overlapping fields of view, through mapping tracked object between camera views, maximum likelihood estimation based on geodesics, and online learning on Riemannian manifolds. The other is for visual pattern classification with sensor fusion of visual-band and thermal infrared cameras, using fused hypotheses from visual and IR information in a unified multi-class AdaBoost. Proposed methods are shown to be effective as demonstrated through experiments based on real world datasets, especially for tracking objects containing long-term partial/full occlusions and frequent intersections, and classifying objects with large intra-class variation and small inter-class variation. However, each of the proposed schemes has limitations and disadvantages, which should be taken into account when designing applications.

For our multi-camera tracking framework, cross-view mapping of tracked object appearances plays the major role in contribution to the performance improvement. However, the computational load would be significantly heavier if the number of cameras increases above 3. Besides, the proposed tracker is based on the assumption that objects move uprightly on a common planar ground that may induce a homography relation between views. This means in scenarios where a dominating ground plane is not present or the target is constantly off the ground, the tracking framework would become ineffective. It is also noticed that the accuracy of cross-view mapping heavily relies on single-view tracking results which gives the top and ground points

of the object, thus leading to increased complexity in single-view tracker. Further, several empirically determined parameters pose another limitation to the general use of the proposed tracking scheme.

For our multi-class classification framework, visual-IR fusion in each boosting iteration is the major reason for increased classification rate, since each weak hypothesis added to the ensemble gets improved. However, as the number of classes continuously increases, say if we want to recognize face poses rotating in every 5 degrees, the scheme would eventually yield very low classification rate. In such a scenario, the dense feature space is beyond the limitation of our fusion strategy. Moreover, as our classification algorithm belongs to the category of supervised learning, it has a high demand of manually labeling of classes for the training data if the training set is relatively large.

# Future Work

The proposed schemes have shown some promising experimental results, however, they can still be improved in the following ways, but not limited to them.

For the multi-camera tracker, other geometric constraints need to be sought for cross-view mapping in more general cases without the assumption of a dominating ground plane. Also, it is preferred that a mechanism for adaptive parameterization is proposed for a more generic and ease-of-use tracker.

For the multi-class classifier, a robust algorithm for automatic detection of object region is needed to replace manually cropping. Besides, a faster and online training algorithm can be developed for our fusion-based classification framework.

Another possible research direction is to combine the tracking and classification modules for the aim of activity analysis, or to integrate multiple visual and infrared cameras for robust recognition of activities in day/night and occlusion scenarios.

# References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, article 13, Dec. 2006.

[2] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification," *John Wiley & Sons*, edition 2, 2000.

[3] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 789–795, Madison, USA, Jun. 16 - 22, 2003.

[4] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 1051–1058, San Diego, USA, Jun. 20 - 25, 2005.

[5] J. Pan and B. Hu, "Robust occlusion handling in object tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, Minneapolis, USA, Jun. 17 - 22, 2007.

[6] N. Amezquita, R. Alquezar, and F. Serratosa, "Dealing with occlusion in a probabilistic object tracking method," in *Proceedings of IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum (in conjunction with CVPR)*, pp. 1–8, Anchorage, Alaska, USA, Jun. 27, 2008.

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

[8] G. Chao, S. Jeng, and S. Lee, "An improved occlusion handling for appearance-based tracking," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 465–468, Brussels, Belgium, Sept. 11 - 14, 2011.

[9] S. Kwak *et al.*, "Learning occlusion with likelihoods for visual tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 1551–1558, Barcelona, Spain, Nov. 6 - 13, 2011.

[10] Z.H. Khan and I.Y.H. Gu, "Bayesian online learning on Riemannian manifolds using a dual model with applications to video object tracking," in *Proceedings of IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition (in conjunction with ICCV)*, pp. 1402–1409, Barcelona, Spain, Nov. 13, 2011.

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

[12] Y. Wu *et al.*, "Real-time visual tracking via incremental covariance tensor learning," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 1631–1638, Kyoto, Japan, Sept. 29 - Oct. 2, 2009.

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

[14] A. Mittal and L.S. Davis, "$M_2$Tracker: 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.

[15] J. Chen and Q. Ji, "Efficient 3D upper body tracking with self-occlusions," in *Proceedings of IAPR International Conference on Pattern Recognition (ICPR)*, pp. 3636–3639, Istanbul, Turkey, Aug. 23 - 26, 2010.

[16] J. Harguess, C. Hu, and J.K. Aggarwal, "Occlusion robust multi-camera face tracking," in *Proceedings of IEEE International Workshop on Machine Learning for Vision-based Motion Analysis (in conjunction with CVPR)*, pp. 31–38, Colorado Springs, USA, Jun. 25, 2011.

[17] J. Kang, I. Cohen, and G. Medioni, "Multi-views tracking within and across uncalibrated camera streams," in *Proceedings of ACM SIGMM International Workshop on Video Surveillance*, pp. 21–33, Berkeley, California, USA, Nov. 7, 2003.

[18] Y.D. Wang, J.K. Wu, and A.A. Kassim, "Particle filter for visual tracking using multiple cameras," in *Proceedings of IAPR International Conference on Machine Vision Applications*, pp. 298–301, Tsukuba, Japan, May 16 - 18, 2005.

[19] J. Fan, *et al.*, "Distributed multi-camera object tracking with Bayesian inference," in *IEEE International Symposium on Circuits and Systems*, pp. 357–360, Rio de Janeiro, Brazil, May 15 - 18, 2011.

[20] Y. Zhou, H. Nicolas, and J. Benois-Pineau, "A multi-resolution particle filter tracking in a multi-camera environment," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 4065–4068, Cairo, Egypt, Nov. 7 - 10, 2009.

[21] C. Chu *et al.*, "Tracking across multiple cameras with overlapping views based on brightness and tangent transfer functions," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–6, Ghent, Belgium, Aug. 22 -25, 2011.

[22] W. Qu, D. Schonfeld, and M. Mohamed, "Decentralized multiple camera multiple object tracking," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pp. 245–248, Toronto, Canada, Jul. 9 - 12, 2006.

[23] A.C. Sankaranarayanan and R. Chellappa, "Optimal multi-view fusion of object locations," in *Proceedings of IEEE Workshop on Motion and Video Computing*, pp. 1–8, Copper Mountain, Colorado, USA, Jan. 8 - 9, 2008.

[24] W. Du and J. Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," in *Proceedings of Asian Conference on Computer Vision*, vol. 1, pp. 365–374, Tokyo, Japan, Nov. 18 - 22, 2007.

[25] Z. Yue, S.K. Zhou, and R. Chellappa, "Robust two-camera tracking using homography," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1–4, Montreal, Canada, May 17 - 21, 2004.

[26] B. Kwolek, "Multi camera-based person tracking using region covariance and homography constraint," in *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 294–299, Boston, USA, Aug. 29 - Sept. 1, 2010.

[27] S. Calderara, A. Prati, and 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.

[28] Y. Guo *et al.*, "Soft margin adaboost for face pose classification," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 221–224, Hong Kong, Apr. 6 - 10, 2003.

[29] S. Baluja, M. Sahami, and H.A. Rowley, "Efficient face orientation discrimination," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 589–592, Singarpore, Oct. 24 - 27, 2004.

[30] P. Viola and M. Jones, "Robust real-time object detection," *International Journal on Computer Vision (IJCV)*, vol. 57, no. 2, pp. 137–154, 2001.

[31] C. Huang *et al.*, "Boosting nested cascade detector for multi-view face detection," in *Proceedings of IAPR International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 415–418, Aug. 23 - 26, Cambridge, UK, 2004.

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

[33] Z. Yang *et al.*, "Multi-view face pose classification by tree-structured classifier," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 358–361, Genoa, Italy, Sept. 11 - 14, 2005.

[34] E. Islam, A. Khan, and I. Kim, "Effective face pose classification method," in *Proceedings of IEEE International Conference on Computer, Control and Communication (IC4)*, pp. 1–6, Karachi, Pakistan, Feb. 17 - 18, 2009.

[35] J. Wang and E. Sung, "EM enhancement of 3D head pose estimated by point at infinity," *Image and Vision Computing*, vol. 25, no. 12, pp. 1864–1874, 2007.

[36] B. Raytchev, I. Yoda, and K. Sakaue, "Head pose estimation by nonlinear manifold learning," in *Proceedings of IAPR International Conference on Pattern Recognition (ICPR)*, pp. 462–466, Cambridge, UK, Aug. 23 - 26, 2004.

[37] S. Li *et al.*, "Learning multiview face subspaces and facial pose estimtation using independent component analysis," *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 705–712, 2005.

[38] M. Hanif and U. Ali, "Optimized visual and thermal image fusion for efficient face recognition," in *Proceedings of IEEE International Conference on Information Fusion*, pp. 1–6, Florence, Italy, Jul. 10 - 13, 2006.

[39] I. Ulusoy and H. Yuruk, "New method for the fusion of complementary information from infrared and visual images for object detection," *IET Image Processing*, vol. 5, no. 1, pp. 36–48, 2011.

[40] X. Wang and G. Li, "Fusion algorithm for infrared-visual image sequences," in *Proceedings of IEEE International Conference on Image and Graphics (ICIG)*, pp. 244–248, Hefei, China, Aug. 12 - 15, 2011.

[41] V.E. Neagoe, A.D. Ropot, and A.C. Mugioiu, "Real time face recognition using decision fusion of neural classifiers in the visible and thermal infrared spectrum," in *Proceedings of IEEE International Conference*

*on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 301–306, London, UK, Sept. 5 - 7, 2007.

[42] A. Apatean *et al.*, "Visible-infrared fusion in the frame of an obstacle recognition system," in *Proceedings of IEEE International Conference on Automation Quality and Testing Robotics (AQTR)*, vol. 1, pp. 1–6, Cluj-Napoca, Romania, May 28 - 30, 2010.

[43] C. Barbu, J. Peng, and G. Seetharaman, "Boosting information fusion," in *Proceedings of ISIF International Conference on Information Fusion (FUSION)*, pp. 1–8, Edinburgh, UK, Jul. 26 - 29, 2010.

[44] Z. Chen, "Bayesian filtering: from Kalman filters to particles filters, and beyond," *Statistics*, pp. 1–69, 2003.

[45] M.S. Arulampalam *et al.*, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[46] W.M. Boothby, "An Introduction to Differentiable Manifolds and Riemannian Geometry," *Academic Press*, edition 2, 2002.

[47] J.M. Lee, "Introduction to Smooth Manifolds," *Springer*, edition 1, 2002.

[48] P.A. Absil, R. Mahony, and R. Sepulchre, "Optimization Algorithms on Matrix Manifolds," *Princeton University Press*, 2008.

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

[50] A. Arsigny *et al.*, "Geometric means in a novel vector space structure on symmetric-positive definite matrices," *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, vol. 29, no. 1, pp. 328–347, 2007.

[51] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *Proceedings of European Conference on Computer Vision (ECCV)*, vol. 2, pp. 589–699, Graz, Austria, May 7 - 13, 2006.

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

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

[54] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on Lie algebra," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 728–735, New York, USA, Jun. 17 - 22, 2006.

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

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

[57] C.J.C. Burges, "A tutorial on support vector machine for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[58] M. Rodriguez, "Multi-class boosting," *Lecture notes, Department of Computer Science, University of California*, Santa Cruz, USA, 2009.

[59] J. Zhu *et al.*, "Multi-class AdaBoost," *Statistics and its Interface*, vol. 2, pp. 349–360, 2009.

[60] Y. Yun, I.Y.H. Gu, H. Aghajan, "Riemannian manifold-based support vector machine for human activity classification in images," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 3466–3469, Melbourne, Australia, Sept. 15 - 18, 2013.

[61] "PETS 2009 benchmark data," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (in conjunction with CVPR)*, Miami, USA, Jun. 25, 2009. [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2009/

[62] "MSR Daily Activity 3D dataset," *Microsoft Research*, USA. [Online]. Available: http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/default.htm

# Part II

# Included papers

# Paper A

## Multi-View ML Object Tracking with Online Learning on Riemannian Manifolds by Combining Geometric Constraints

Yixiao Yun, Irene Yu-Hua Gu, and Hamid Aghajan

Published in
*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, Jun. 2013

*The layout has been revised.*

**Abstract**

This paper addresses issues in object tracking with occlusion scenarios, where multiple uncalibrated cameras with overlapping fields of view are exploited. We propose a novel method where tracking is first done independently in each individual view and then tracking results are mapped from different views to improve the tracking jointly. The proposed tracker uses the assumptions 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. A method for online learning of object appearances on Riemannian manifolds is also introduced. The main novelties of the paper include: (a) define a similarity measure, based on geodesics between a candidate object and a set of mapped references from multiple views on a Riemannian manifold; (b) propose multi-view maximum likelihood (ML) estimation of object bounding box parameters, based on Gaussian-distributed geodesics on the manifold; (c) introduce online learning of object appearances on the manifold, taking into account of possible occlusions; (d) utilize projective transformations for objects between views, where parameters are estimated from warped vertical axis by combining planar homography, epipolar geometry and vertical vanishing point; (e) embed single-view trackers in a three-layer multi-view tracking scheme. Experiments have been conducted on videos from multiple uncalibrated cameras, where objects contain long-term partial/full occlusions, or frequent intersections. Comparisons have been made with three existing methods, where the performance is evaluated both qualitatively and quantitatively. Results have shown the effectiveness of the proposed method in terms of robustness against tracking drift caused by occlusions.

# 1 Introduction

Tracking visual objects from videos has been one of the important research topics in computer vision [1]. Many robust trackers have been developed, however, challenges remain especially when dynamic objects experience long-term partial/full occlusions, or intersections with other objects in crowded scenes. When an object of interest is partially or fully occluded by other objects in images, its appearance is more or less altered by the occluding objects. Tracking occluded objects becomes more difficult, which is likely to cause tracking drift. Occlusion handling is required for mitigating the drift.

**Related Work:** Many existing approaches deal with occlusions in a single camera view. Papadakis and Bugeau [2] propose to track occluded objects by segmenting them into visible and occluded parts based on graph cuts. Chao et al [3] recognizes 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 [4] divides target into regular grid cells and detects occlusion for each cell using a classifier. Riemannian manifold trackers with a single camera were applied in [16, 17, 19], 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.

On the other hand, object tracking using multiple cameras has drawn growing interest in recent years [5], 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 [5]. One category of multi-view tracking methods handles the occlusion issue through using calibrated cameras. For example, Mittal and Davis [6] detect 3-D points on an object by applying a region-based stereo algorithm, and analyzes object occlusions by pixel-based classification of visible and occluded parts under Bayesian framework. 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. These methods exploit underlying multi-view geometric constraints. Two constraints are often used, e.g., Chu et al [7] uses ground plane homography and Qu et al [8] uses epipolar geometry. Yue et al [9] conducts two-view tracking by using a particle filter in each view, and detects occlusions by comparing pixel differences between tracked and template object. Kwolek [10] 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 maintain the tracking in oc-

cluded 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, additional geometric constraints should be added.

Motivated by the above issues, we propose a novel method for multi-view object tracking using uncalibrated cameras. The method does not require occlusion detection, as tracking results are mapped between views at each time instant. Tracking drift in occluded scenes is mitigated by using information in un-occluded views. In addition to planar homography, the epipolar line and vertical vanishing point are added to jointly warp the object vertical axis. The main novelties of this paper include: defining a similarity measure on Riemannian manifolds. It is built upon geodesics between a candidate object and a set of mapped references from multiple views; proposing multi-view maximum likelihood (ML) estimation of object bounding box parameters, based on Gaussian-distributed geodesics on the manifold; introducing online learning of object appearances that takes into account of possible occlusion; utilizing projective transformations for mapping tracked objects between views, where parameters are estimated from warped vertical axis by combining planar homography, epipolar geometry and vertical vanishing point; embedding single-view trackers in a three-layer multi-view tracking scheme. Comparing with our previous work in [11], this paper makes further improvement on integrating the tracking scheme with an online learning scheme, using projective transformations and more extensive tests and performance evaluations.

The remainder of this paper is organized as follows: Section 2 briefly introduces the geometry of Riemannian manifolds, the region covariance descriptor and the vertical axis for multi-view object. Section 3 and 4 present the big picture and the details of proposed tracking and online learning schemes, respectively. Section 5 describes the single-view tracker used in each individual view. Section 6 shows experimental results on multi-view videos. Finally Section 7 concludes the paper.

## 2 Riemannian Manifold Geometry, Region Covariance Descriptor, and Vertical Axes for Multiview Object: Review

This section briefly reviews: Riemannian geometries with focus on the space of symmetric positive definite matrices [12] [13] [14] [15]; object descriptor using region covariances [18]; and relations vertical axes of multiview object under planar homography and other geometrical constraints [23, 24], for the

sake of mathematical convenience in subsequent sections.

## 2.1 Manifold of Symmetric Positive Definite Matrices

A manifold is a topological space as low dimensional subspaces embedded in a high dimensional space, which is only locally Euclidean. Fig. 1 depicts a two-dimensional manifold embedded in $\mathbb{R}^3$. Our notation follows that in [19]: $\mathbf{p}$ is the starting point and $\mathbf{q}$ is the end point on a manifold $\mathcal{M}$, and $\mathbf{\Delta}$ is the velocity vector in the tangent space $\mathcal{T}$. A Riemannian manifold is a differentiable manifold where the tangent space at each point has an inner product $<,>_{\mathbf{p}\in\mathcal{M}}$ that varies smoothly from point to point.

The space of $l \times l$ symmetric positive definite ($Sym_l^+$) matrices lies on a Riemannian manifold that constitutes the convex-half cone in a vector space of matrices [19]. To compute statistics on $Sym_l^+$, *affine-invariant* metric [20] and *log-Euclidean* metric [21] are commonly used, leading to similar numerical results. The log-Euclidean metric is adopted in this paper since it is computationally more efficient [21].

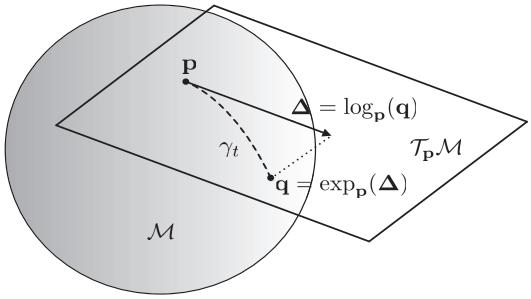



**Figure 1:** Example of 2-D manifold $\mathcal{M}$ embedded in $\mathbb{R}^3$. $\mathbf{p}$ and $\mathbf{q}$ are manifold points, $\mathcal{T}_{\mathbf{p}}\mathcal{M}$ is the tangent space for $\mathbf{p}$, $\mathbf{\Delta}$ is the tangent vector whose projected point on the manifold is $\mathbf{q}$. The geodesic $\gamma_t$ is the minimum length curve between $\mathbf{p}$ and $\mathbf{q}$ on the manifold.

As shown in Fig. 1, *Exponential map ($\mathcal{T} \mapsto \mathcal{M}$)* is the function that maps a tangent vector $\mathbf{\Delta}$ along the geodesic $\gamma_t$ to a point $\mathbf{q}$ on the manifold $\mathcal{M}$, given by $\exp_{\mathbf{p}}(\mathbf{\Delta}) = \exp(\log(\mathbf{p}) + \mathbf{\Delta})$ under the log-Euclidean metric, and $\exp_{\mathbf{p}}(\mathbf{\Delta}) = \mathbf{p}^{\frac{1}{2}} \exp(\mathbf{p}^{-\frac{1}{2}} \mathbf{\Delta} \mathbf{p}^{-\frac{1}{2}}) \mathbf{p}^{\frac{1}{2}}$ under the affine invariant metric, where $\mathbf{p}$ is the starting point on $\mathcal{M}$.

Conversely, *Logarithmic map ($\mathcal{M} \mapsto \mathcal{T}$)* is the function that maps a manifold point $\mathbf{q}$ (start from $\mathbf{p}$) to a vector $\mathbf{\Delta}$ in the tangent space $\mathcal{T}_{\mathbf{p}}$. It corresponds to a velocity vector, and is given by $\log_{\mathbf{p}}(\mathbf{q}) = \log(\mathbf{q}) - \log(\mathbf{p}) = \mathbf{\Delta}$ under the log-Euclidean metric, and $\log_{\mathbf{p}}(\mathbf{q}) = \mathbf{p}^{\frac{1}{2}} \log(\mathbf{p}^{-\frac{1}{2}} \mathbf{q} \mathbf{p}^{-\frac{1}{2}}) \mathbf{p}^{\frac{1}{2}}$ under the affine invariant metric.

*Geodesic* is the shortest curve between two points $\mathbf{p}$, $\mathbf{q}$ on a manifold

$\mathcal{M}$. The geodesic distance is given by

$$d(\mathbf{p},\mathbf{q}) = \| \log_{\mathbf{p}}(\mathbf{q})\| = \| \log(\mathbf{q}) - \log(\mathbf{p})\| \tag{1}$$

under the log-Euclidean metric, and $d(\mathbf{p},\mathbf{q}) = \mathrm{tr}[\log^2(\mathbf{p}^{-\frac{1}{2}}\mathbf{q}\mathbf{p}^{-\frac{1}{2}})]$ under the affine invariant metric. Another alternative for computing the geodesic distance [18] is $d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{k=1}^{n} \ln^2 \lambda_k(\mathbf{p},\mathbf{q})}$ where $\{\lambda_k(\mathbf{p},\mathbf{q})\}_{k=1}^{n}$ are the generalized eigenvalues of $\mathbf{p}$ and $\mathbf{q}$.

## 2.2 Region Covariances as Object Descriptors

A region covariance matrix [18] enables an effective description of object features, and is shown to be robust and versatile to variations in illuminations, views and poses at modest computational cost by using integral images.

Given a rectangular image region $R$, let $\{\mathbf{f}_k\}_{k=1}^{|R|}$ be $l$-dimensional feature vectors for each pixel inside $R$, where $|R|$ is the number of pixels in $R$. The features can be, e.g., intensity, color, gradient, or filter responses. For example, a feature vector can be formed as

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

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. The region $R$ is described by a $l \times l$ covariance matrix $\mathbf{C}_R = \frac{1}{|R|-1}\sum_{k=1}^{|R|}(\mathbf{f}_k - \boldsymbol{\mu})(\mathbf{f}_k - \boldsymbol{\mu})^T$, where $\boldsymbol{\mu}$ is the sample mean. Since $\mathbf{C}_R \in Sym_l^+$, covariance matrices $\mathbf{C}_R$ from different video frames can be viewed as connected points on a Riemannian manifold [22].

## 2.3 Mapping Vertical Axis of Object in Different Views

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). 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. We briefly describe the method described in [23].

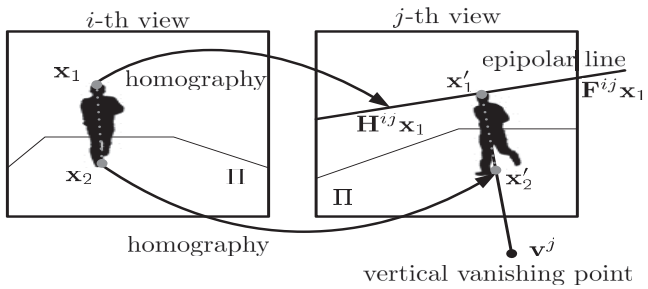



**Figure 2:** Warping vertical axis of a tracked object from $i$-th view to $j$-th view by combining the constraints of planar homography, epipolar geometry and vertical vanishing point.

### Planar Homography

Let 2D 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 $\Pi$ 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 $\Pi$, $\mathbf{x}'_1 \neq \mathbf{H}^{ij}\mathbf{x}_1$ (see Fig. 2). Homography is not sufficient for warping the vertical axis of object, additional geometric constraints should be added.

### Epipolar Geometry

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 $\mathbf{x}'_1\mathbf{F}^{ij}\mathbf{x}_1 = 0$. 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). Thus, the epipolar geometry constrains the corresponding points that lie on the conjugate pairs of epipolar lines.

### Vertical Vanishing Point

To obtain the warped axis inclination, the vertical vanishing point $\mathbf{v}^j$ of $j$-th view is used. As depicted in Fig. 2, 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 [24].

Using the same procedure, the vertical axis of tracked object in the $j$-th view may be warped to the $i$-th view.

# 3 The Big Picture: Overview of the Proposed Tracking Method

The proposed tracking scheme consists of two major parts: (i) multi-view Maximum Likelihood (ML) tracking; (ii) online learning of object appearances on the Riemannian manifold. As shown in Fig. 3, these two process are performed in an alternative fashion.

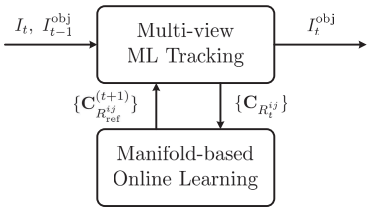



**Figure 3:** The block diagram of proposed scheme. $I_t$ is the image frame at $t$, $I_{t-1}^{\text{obj}}$ and $I_t^{\text{obj}}$ is the tracked object at $t-1$ and $t$, respectively. $\mathbf{C}_{R_t^{ij}}$ is the manifold point of tracked object, and $\mathbf{C}_{R_{\text{ref}}^{ij}}^{(t)}$ is the updated reference model.

*Tracking part:* As shown in Fig. 4, Tracking part is performed in three layers in cascade at each time instant. In the first layer, a single view Bayesian framework-based tracker is applied to track a candidate object from a given view. In the second layer, tracked object from each view is mapped to the remaining views by using planar homography plus other geometrical constraints. Once the correspondences between different views are established, a manifold-based maximum likelihood (ML) criterion is applied to obtain the best multi-view tracking result in the third layer.

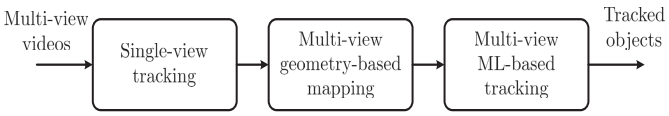



**Figure 4:** Three layers in the multiview tracking scheme.

The essence of such a multi-view tracker is to regard an object in different views as different points on a same manifold (see schematic description in Fig. 5). 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.

*Online Learning Part:* In order to maintain a timely reference object set as video objects may change with time, online learning of reference object

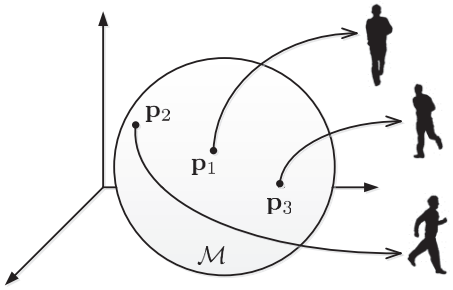



**Figure 5:** Description of different views of object as points on a smooth manifold.

set is an important issue for preventing tracking drift. However, one should also be careful of not learning the wrong objects when object partially/fully occurs. A strategy needs to be adopted to decide whether or not such an online learning should be activated at a given time instant. We propose to use a simple criterion to decide whether online learning is applied, based on the observation that changes caused by a dynamic object are usually less significant as comparing with changes caused by an occluding object. A simple measure is formed based on the histogram cross-correlation between the reference and tracked object (or, the Bhattacharyya coefficient) in combination of object mapping between different views.

Details of these two parts are described in the next section.

# 4 Multi-View ML Object Tracking with Manifold-based Online Learning

Let the total number of cameras be $M$ ($M \geq 2$). Given the positions and appearances of a tracked object from $M$ individual trackers in each view, our aim is to map these tracking results from the combination of $(M-1)$ views to an arbitrary view (e.g. $i$-th view), where the mapped positions and appearances are consistent with the destination view in terms of pixel coordinate, scale and 2D orientation. Here, the maximum likelihood (ML) measure of mapped object positions is applied. For each view, $(M-1)$ mapped estimates and one self-estimate of object position are collected to jointly improve the object position estimation in the $i$-th view in the ML sense. Once the correspondences of vertical axes of tracked object in different views are established (see the method in Section 2.3), the parameters of bounding box as well as the object image are then mapped as follows.

## 4.1 Mapping the Position and Appearance of Tracked Object

In each view, object motion is approximated by a 2D projective transformation. The state vector of tracked object bounding box at time $t$ in $i$-th view is specified by nine parameters:

$$\mathbf{s}_t^i = \left[x_t^i \;\; y_t^i \;\; \alpha_t^i \;\; \beta_t^i \;\; \theta_t^i \;\; \phi_t^i \;\; u_t^i \;\; v_t^i \;\; w_t^i\right]^T \tag{3}$$

where $x_t^i$ and $y_t^i$ are translations along $x$, $y$ directions, $\alpha_t^i$ and $\beta_t^i$ denote scalings of box in $x$, $y$ axes, $\theta_t^i$ is rotation angle, $\phi_t^i$ is a skew parameter, and $u_t^i$, $v_t^i$, $w_t^i$ are perspective projection parameters [24].

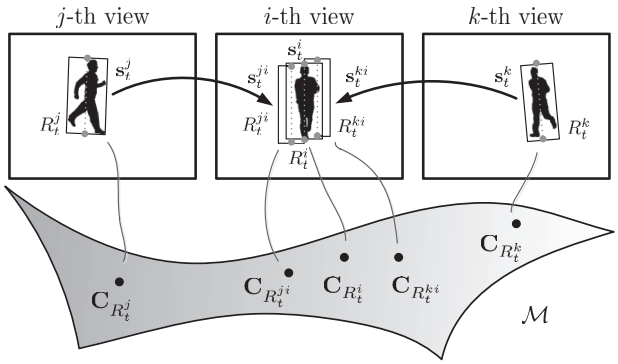



**Figure 6:** Mapping positions and appearances of a tracked object from $j$-th and $k$-th views to $i$-th view, based on the warped vertical axis.

Once the state vector $\hat{\mathbf{s}}_t^i$ is estimated, the object bounding box is obtained. We approximate the top point of object by the midpoint of top edge in the bounding box, and the ground point of object by the midpoint of bottom edge of box. So the vertical axis of object is obtained by the line segment connecting these two points. Using the offline estimated homography matrices $\mathbf{H}^{ij}$, fundamental matrices $\mathbf{F}^{ij}$ and vertical vanishing points $\mathbf{v}^j$, the vertical axis of tracked object in $i$-th view is warped to $j$-th view ($j = 1, \cdots, M$, $j \neq i$), and vice versa by combining $\mathbf{H}^{ji}$, $\mathbf{F}^{ji}$ and $\mathbf{v}^i$.

As shown in Fig. 6, based on the warped vertical axis of object from the $j$-th to the $i$-th view, tracking results in $j$-th view (including the MAP estimated object position $\hat{\mathbf{s}}_t^j$ and its corresponding region $R_t^j$) are projected to the $i$-th view (denoted as $\mathbf{s}_t^{ji}$ and $R_t^{ji}$) using the method as follows.
Mapping tracked object between two views can be approximated by a 2D projective transformation:

$$\mathbf{u}' = \begin{bmatrix} \mathbf{R}^{ji} & \mathbf{t}^{ji} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K}^{ji} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ (\mathbf{v}^{ji})^T & w^{ji} \end{bmatrix} \mathbf{u} \tag{4}$$

where $\mathbf{u}$ is a 2D homogeneous point at $[x\ y\ 1]^T$ in the $j$-th view, and $\mathbf{u}'$ is the transformed homogeneous point in the $i$-th view, $\mathbf{R}^{ji} = \begin{bmatrix} \cos\theta^{ji} & -\sin\theta^{ji} \\ \sin\theta^{ji} & \cos\theta^{ji} \end{bmatrix}$ is a rotation matrix, $\mathbf{t}^{ji} = \begin{bmatrix} x^{ji} & y^{ji} \end{bmatrix}^T$ is a translation vector, $\mathbf{K}^{ji} = \begin{bmatrix} \alpha^{ji} & \phi^{ji} \\ 0 & \beta^{ji} \end{bmatrix}$ is a skew and scale matrix (upper triangular), $\mathbf{v} = [u^{ji}\ v^{ji}]^T$ is a two-component vector that determines the position of the line at infinity, and $w^{ji}$ is a scalar [24].

The estimation of parameters in (4) requires at least 4 pairs of corresponding points. Using the top and ground point correspondences in vertical axes is not sufficient. Therefore, we use the state vector $\hat{\mathbf{s}}^i_{t-1}$ at time $(t-1)$ in $i$-th view, by assuming parameters change smoothly between consecutive frames. In this way, 9 pairs of point correspondences with respect to the bounding box are obtained: 4 corner points, midpoints of 4 edges in the bounding box, and 1 center point. These correspondences are used to estimate the parameters of projective transformation between views.

Since the concatenation of two projective transformations is a projective transformation [24], the parameters of $\mathbf{s}^{ji}_t$ that map the position of tracked object from the $j$-th to the $i$-th view may be derived by decomposing concatenated projective transformations in a similar way as in (4). Next, the appearance image of tracked object in $j$-th view is mapped to the $i$-th view by building correspondences of pixel coordinates in $R^j_t$ and $R^{ji}_t$ using (4), followed by inverse mapping of pixel values using 2D interpolations. By applying this procedure, positions and appearances of tracked objects from $(M-1)$ views are mapped to the $i$-th view. An example is given in Fig. 7.

**Figure 7:** Mapping positions and appearances of a tracked object between views, based on warped vertical axis. Column 1: detected vertical axis (dotted line) and object region in one view; Column 2: warped vertical axis in another view, where the vertical line passes through the warped axis, the inclined line is the epipolar line associated with object top point from the view in Column 1; Column 3: mapped appearance in the view in Column 2. Rows 1-2: two different video scenes.

## 4.2 Multi-View ML Estimation of Object Position

Before applying the ML criterion, a reference model $\mathcal{C}^i_{\text{ref}}$, containing all $M$ individual mapped views, is formed. To obtain the reference in $i$-th view, reference models in all views are exploited. Since object appearances in different views are not consistent with one another in terms of pixel coordinates, scale and 2D orientation, they are mapped to the $i$-th view before computing the appearance descriptor. The mapping is achieved by using the method described in Section 4.1, resulting in a mapped reference object region $R^{ji}_{\text{ref}}$ in the $i$-th view from the $j$-th view. For notational convenience, $R^{ii}_{\text{ref}} = R^i_{\text{ref}}$ is used.

We choose region covariance matrices as the feature descriptor of object appearance (see Section 2.2). Let the covariance matrix of reference object in $j$-th view mapped to $i$-th view be $\mathbf{C}_{R^{ji}_{\text{ref}}}$. Then, the reference model in $i$-th view $\mathcal{C}^i_{\text{ref}}$ is formed by a set of $M$ component views $\mathcal{C}^i_{\text{ref}} = \left\{\mathbf{C}_{R^{1i}_{\text{ref}}}, \cdots, \mathbf{C}_{R^{Mi}_{\text{ref}}}\right\}$. The dissimilarity measure is based on the geodesic on the Riemannian manifold that is computed between a candidate object in $R^{ji}_t$ and the reference model $\mathcal{C}^i_{\text{ref}}$ by:

$$d(\mathcal{C}^i_{\text{ref}}, \mathbf{C}_{R^{ji}_t}) = \min_{k=1,\cdots,M} \left\| \log(\mathbf{C}_{R^{ki}_{\text{ref}}}) - \log(\mathbf{C}_{R^{ji}_t}) \right\| \tag{5}$$

Given the region of tracked object $R^{ji}_t$ (i.e., mapped from the $j$-th to the $i$-th view), $\mathbf{C}_{R^{ji}_t}$ is computed. Similarly, denote $R^{ii}_t = R^i_t$ and $\mathbf{s}^{ii}_t = \mathbf{s}^i_t$. The likelihood is computed from the Gaussian-distributed *geodesic* between $\mathcal{C}^i_{\text{ref}}$ and $\mathbf{C}_{R^{ji}_t}$ by:

$$p(\mathbf{C}_{R^{ji}_t} | \mathcal{C}^i_{\text{ref}}) \propto \exp\left( -\frac{d^2(\mathcal{C}^i_{\text{ref}}, \mathbf{C}_{R^{ji}_t})}{2\sigma^2_{i,1}} \right) \tag{6}$$

where $d(\cdot)$ is the geodesic defined in (5), and $\sigma^2_{i,1}$ is empirically determined. Then, the ML estimate in the $i$-th view is obtained by:

$$\hat{\mathbf{s}}^i_t = \mathbf{s}^{j^* i}_t, \quad j^* = \arg \max_{j=1,\ldots,M} p(\mathbf{C}_{R^{ji}_t} | \mathcal{C}^i_{\text{ref}}) \tag{7}$$

If $j^* \neq i$, then, the individual tracker in the $i$-th view is re-initialized.

## 4.3 Online Learning of Object Appearances on the Manifold

Online updating of reference model containing multi-view object appearances is designed to mitigate tracking drifts due to object appearance changes through video frames. The basic idea behind the proposed online updating scheme is to apply Bhattacharyya coefficient to examine whether there is an

indication of object occlusion. The reference object update is only applied at frames where this metric indicates low possibility of occlusions. Fig. 8 shows the block diagram of online learning for an arbitrary $i$-th view.

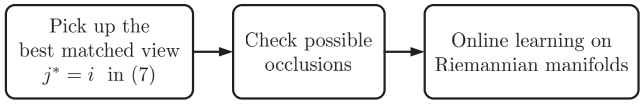



**Figure 8:** A block diagram of the proposed online learning scheme in one view. Without loss of generality, $j$-th view is selected.

Updating is only considered to the $i$-th view, where $j^* = i$ is the best view obtained in (7) from the tracking result. We consider a simple criterion, the Bhattacharyya coefficient between the currently tracked object and the $i$-th view reference object in the model:

$$\rho(p, q) = \sum_{l}^{m} \sqrt{p_l q_l} \in [0, 1] \tag{8}$$

where $p$ and $q$ are the normalized color histograms from the tracked object and the $i$-th view reference object, $m$ is the number of histogram bins. The basic idea is that an occluded object is less correlated to its own clean reference. If $\rho(p, q) > T$ ($T$ is a predetermined threshold value), it is an indication that the object is unlikely to be occluded.
If the following conditions are satisfied:

$$j^* \equiv i \qquad \text{and} \qquad \rho(p, q) > T \tag{9}$$

Then, the $i$-th component $\mathbf{C}_{R^{ij}_{\text{ref}}}$ in the reference model $\mathcal{C}^j_{\text{ref}}$ is updated as follows:

$$\mathbf{C}^{(t+1)}_{R^{ij}_{\text{ref}}} = \exp\left(\varepsilon \log \mathbf{C}^{(t)}_{R^{ij}_{\text{ref}}} + (1 - \varepsilon) \log_{\mathbf{C}^{(t)}_{R^{ij}_{\text{ref}}}} \mathbf{C}_{R^{ij}_t}\right) \tag{10}$$

where $j = 1, \cdots, M$, and $\varepsilon \in [0, 1]$ is a weighting factor. The update rule can be interpreted as the reference point $\mathbf{C}^{(t)}_{R^{ij}_{\text{ref}}}$ moving to nearby point $\mathbf{C}_{R^{ij}_t}$ due to appearance change, with the traveling speed restrained by a weighting factor $\varepsilon$.

# 5 Object Tracking in Individual Views

This section briefly describes the individual view-based tracker in the 1st layer, for the sake of completeness.

In Layer-1 (see Fig. 4), appearance-based tracking is performed independently in each view, using the multi-view tracking results from the previous

frame. This corresponds to a single-view object tracker if one camera is used.

To be consistent with the third layer, the region covariance is used to model the object appearance (see Section 2.2). Using the state vector $\mathbf{s}_t^i$ in (3), a Brownian motion model is used between states:

$$\mathbf{s}_t^i = \mathbf{s}_{t-1}^i + \mathbf{w}_t^i \tag{11}$$

where $\mathbf{w}_t^i \sim \mathcal{N}(0,\ \mathbf{\Omega}^i)$, and

$$\mathbf{\Omega}^i = \left(\text{diag}(\sigma_x^i, \sigma_y^i, \sigma_\alpha^i, \sigma_\beta^i, \sigma_\theta^i, \sigma_\phi^i, \sigma_u^i, \sigma_v^i, \sigma_w^i)\right)^2 \tag{12}$$

is the covariance containing diagonal elements, each corresponding to the variance of individual parameters of $\mathbf{s}_t^i$. These variances are determined empirically. A sequential importance sampling (SIS) particle filter with re-sampling [25] is used to approximate the recursive Bayesian estimation. The posterior pdf is estimated by $p(\mathbf{s}_t^i|\mathbf{z}_{0:t}^i) = \sum_{i=1}^N \omega_t^{i,n}\delta(\mathbf{s}_t^i - \mathbf{s}_t^{i,n})$ where $N$ is the total number of particles. Assuming $p(\mathbf{s}_t^{i,n}|\mathbf{s}_{t-1}^{i,n})$ is equal to the proposal $q(\mathbf{s}_t^{i,n}|\mathbf{s}_{t-1}^{i,n}, \mathbf{z}_t^i)$, particle weights $\omega_t^{i,n}$ are updated by $\omega_t^{i,n} \propto \omega_{t-1}^{i,n} p(\mathbf{z}_t^i|\mathbf{s}_t^{i,n})$, followed by the normalization. Further, re-sampling is applied if $N_{\text{eff}} = 1/\sum_{n=1}^N (\omega_t^{i,n})^2 < N_{\text{th}}$ is satisfied [25]. The likelihood of observations in candidate object regions $\{R_t^{i,n}\}_{n=1}^N$ is defined as

$$p(\mathbf{z}_t^i|\mathbf{s}_t^{i,n}) \propto p(\mathbf{z}_t^i|\mathbf{g}(\mathbf{s}_t^{i,n})) = \exp\left(-\frac{d^2(\mathbf{C}_{R_{\text{ref}}^i}, \mathbf{C}_{R_t^{i,n}})}{2\sigma_{i,2}^2}\right) \tag{13}$$

where $\mathbf{g}(\mathbf{s}_t^{i,n})$ is the covariance matrix describing the object appearance within the bounding box, $\sigma_{i,2}^2$ is empirically determined, and $d(\mathbf{C}_{R_{\text{ref}}^i}, \mathbf{C}_{R_t^{i,n}}) = \| \log(\mathbf{C}_{R_{\text{ref}}^i}) - \log(\mathbf{C}_{R_t^{i,n}}) \|$ is the geodesic on the manifold. Finally, the MAP (maximum a posteriori) estimate of the bounding box is computed according to

$$\mathbf{s}_t^i \leftarrow \mathbf{s}_t^{i,n^*}, \quad n^* = \arg\max_{n=1,\ldots,N} (\omega_t^{i,n}) \tag{14}$$

Further, anisotropic mean shift [26] is applied before using the particle filter. Applying mean shift is aimed to guide the particles to a nearby local mode, which leads to a reduced number of required particles due to smaller variances in $\mathbf{\Omega}^i$.

# 6 Experiments and Results

## 6.1 Experimental Setup

The proposed tracking method is tested on PETS 2001, 2006, 2007, 2009 [27] [28] [29] [30], TUG datasets [31] and EPFL datasets [32]. Each dataset

contains synchronized videos from multiple cameras, where 11 sets of multi-view videos containing full occlusion scenarios are selected for our experiments. The tested videos contain 4 two-view scenarios and 7 three-view scenarios. 9 different scenarios, included in the case-studies in this section, are listed in Table 1.

**Table 1:** Information on the tested multi-view videos in case studies.

| Dataset | No. views | Tested frames | No. full occlusions | Full occlusion frames: (shortest, longest) | Case |
|---|---|---|---|---|---|
| PETS'01/S3.Ts | 2 | 129 | 1 | 50 | a |
| PETS'06/S7 | 3 | 300 | 1 | 182 | b |
| PETS'07/S0 | 3 | 129 | 8 | (6,39) | c |
| PETS'09/S2.L1 | 2 | 39 | 3 | (1,7) | d |
| | 2 | 500 | 11 | (4,43) | e |
| TU Graz | 3 | 719 | 19 | (7,168) | f |
| | 3 | 1000 | 14 | (34,135) | g |
| | 3 | 1000 | 27 | (12,45) | h |
| EPFL/Campus | 3 | 290 | 10 | (8,120) | i |

For all tested multi-view videos, the initial bounding box of target object in each view is manually selected. Each object region is normalized to $32 \times 64$ pixels. For each single-view tracker, the number of particles is set to $N = 150$, $\sigma_{i,1}^2 = \sigma_{i,2}^2 = 0.1$, and $N_{\text{eff}} = N/3 = 50$ for re-sampling. For the mean shift tracker, number of histogram bins is $m = 16$, the maximum number of iterations is $n = 10$. For online learning, the threshold $T$ in (9) is $T \in [0.8, 0.85]$, the weighting factor $\varepsilon = 0.6$. For the region covariance, the feature vector is formed as in (2). Variances $\mathbf{\Omega}^i$ in (12) and $\eta_i$ for bandwidth matrix in the mean shift are determined empirically for each view. The ranges of these parameters are: $(\sigma_x^i)^2$, $(\sigma_y^i)^2 \in [3, 15]$; $(\sigma_\alpha^i)^2$, $(\sigma_\beta^i)^2 \in [0.01, 0.03]$; $(\sigma_\theta^i)^2 = 10^{-4}$; $(\sigma_\phi^i)^2 = 0$; $(\sigma_u^i)^2$, $(\sigma_v^i)^2$, $(\sigma_w^i)^2 \in [10^{-6}, 10^{-5}]$; $\eta_i \in [0.001, 0.2]$. These values are set as proportional to the changing rate of parameters. To better evaluate the performance of the proposed tracking method and to isolate the error caused from the estimation of Homography and fundamental matrices, we use manually marked 20 corresponding ground points in each video (from one starting frame) in each view for computing the homography matrix, also marked 20 corresponding salient points in each view for computing the fundamental matrix, and using Hough lines for estimating initial vertical lines in video. For automatic estimating vertical axes of objects, readers are referred to [23] for more details.

To quantitatively evaluate and compare the performance, the following 3 object criteria are applied:

*(a) Euclidean distance*: It is defined as the Euclidean distance between the 4 corners of tracked object bounding box and manually marked Ground

Truth (GT) box:

$$d_E = \frac{1}{4}\sum_{i=1}^{4}\sqrt{(\hat{x}_i - x_i^{GT})^2 + (\hat{y}_i - y_i^{GT})^2} \tag{15}$$

where $(\hat{x}_i, \hat{y}_i)$, $(x_i^{GT}, y_i^{GT})$ are the corresponding corner points.
*(b) Bhattacharyya distance*: It is defined between the tracked object image and GT object image:

$$d_B = \sqrt{1 - \rho(p, q)} \tag{16}$$

where $\rho(p, q)$ is defined in (8), and $d_B \in [0, 1]$ (the smaller value the better).
*(c) Geodesic distance*: It is defined in (1) on the Riemannian manifold between the two covariance matrices extracted from tracked and GT object regions.

For each of these criteria, smaller values correspond to less errors hence better tracking performance.

## 6.2 Test Results from the Proposed Scheme

Experiments were conducted on videos with a range of complexity and scenarios using the proposed tracker. Seven case studies included in this paper are: campus (Case-a, d, e), train station (Case-b), airport (Case-c), and indoor environment (Case-f, g), where objects experience a range of long-term full occlusions and intersections. Some tracking results are included.

**Figure 9:** Proposed tracker: tracking results on videos in Case-e. Videos are obtained from views 1-2 of PETS'09 S2.L1, where an object experiences frequent occlusions and intersections. Tracked boxes are marked in the image (magenta). Row 1-2: results in views 1-2. Key frames (# 19, 73, 157, 318, 491) in video are selected (zoomed in for better inspection of target objects).

Fig. 9 (Case-e) shows the tracking results on the two-view videos in an outdoor environment on campus, where several people are walking around. In this scenario, the density of crowd is low and the background is nearly homogeneous. The size of objects is small since they are located at large

distances to both cameras. The target person is occluded by a post in the 1st view for several times (short duration), and experiences frequent intersections with other people. Observing the tracking results, one can see that the proposed tracker is robust against occlusions and intersections for small objects with tight and accurate bounding boxes.

Fig. 10 (Case-e) shows the resulting trajectory of object in the top view from the proposed scheme as well as the ground truth trajectory (manually marked). It is observed from Fig. 10 that the two trajectories overlap to a great extent. Some jitters from the ground truth trajectory exist, however, not significant.

**Figure 10:** Proposed tracker: trajectories of tracked moving object on video PETS'09 S2.L1: test results and ground truth. The trajectory of tracked object on the ground plane (from the top view) through planer homography mapping. Magenta: from the proposed tracker in View 2. Yellow: ground truth trajectory. '×' and '∘': starting and end points.

Fig. 11 (Case-b) shows the tracking results on the three-view videos in a train station where a person with a suitcase is visible in the 3rd view, but becomes partially occluded by a trolley in some frames, and reappears afterwards in the 2nd view. The target is fully occluded with a long time duration by the same trolley in the later part of frames, and reappears in the end of the 1st view video. In this case, the crowd density is low and the background is not very complicated. The target person is relatively large in the 1st and 3rd views, but is small in the 2nd view. As we observed, even when the person is fully occluded during frames $[648, 829]$ in the 1st view, and partially occluded in frames $[580, 599]$ in the second view, the tracker still follows the target person as long as it is visible in one view, despite of different sizes appeared in each view.

Fig. 12 (Case-c) shows the tracking results on the three-view videos in an airport, where a person walks through a dense crowd. In this scenario, the crowd density is relatively high and the background is complicated. Further,

**Figure 11:** Proposed tracker: tracking results on videos in Case-b. Videos are obtained from views 1, 2, 4 of PETS'06 S7, where an object experiences partial and long-term full occlusions. Tracked boxes are marked in images (magenta). Rows 1-3: results in 3 views. Key frames (# 574, 592, 775, 828, 839) in the video are selected (zoomed in for better inspection of target objects).

illuminations are very different between the 3rd view and remaining views. The target person experiences partial/full occlusion constantly in the 1st and 2nd views. It can be seen from Fig. 12 that the proposed tracker is effective for tracking objects in dense crowd with complex background.

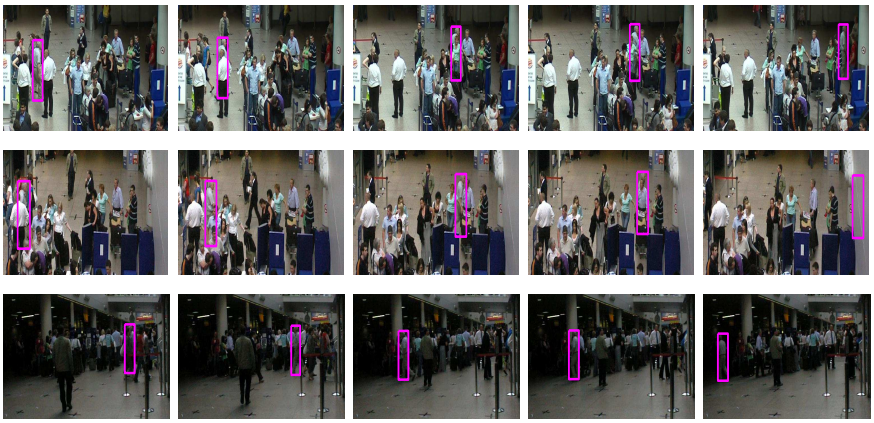


**Figure 12:** Proposed tracker: tracking results on videos in Case-c. Videos are obtained from views 1, 2 and 4 of PETS'07 S0, where an object experiences frequent intersections. Tracked boxes are marked in images (magenta). Row 1-3: results in the 3 views. Key frames (# 1852, 1864, 1934, 1940, 1977) in the video are selected (zoomed in for better inspection of target objects).

Fig. 13 (Case-f) shows the tracking results on the three-view videos in an indoor environment, where a person with others is walking around and experiences frequent intersections. Since all persons are moving in a small area, the frequency of intersections and occlusions are very high. The objects are near to the cameras, so it is more likely to cause full occlusions. It is observed in Fig. 13 that the proposed tracker follows the object with accurate and tight bounding box, despite the target person is invisible in some views due to occlusion.

**Figure 13:** Proposed tracker: tracking results on videos in Case-f. Videos are obtained from views 1-3 of TU Graz Multi-Camera dataset, where an object experiences frequent intersections. Tracked boxes are marked in images (magenta). Row 1-3: results in views 1-3. Key frames (# 3201, 3272, 3369, 3702, 3835) in the video are selected (zoomed in for better inspection of target objects).

Fig. 14 (Case-g) shows the tracking results on the three-view videos in an indoor environment, where a single person is walking around. An image of a book is superimposed to each view for synthetic occlusion. The target person then experiences frequent full occlusions with long time durations in each view. It is observed in Fig. 14 that the proposed tracker is able to follow the object with accurate and tight bounding box as long as it is visible in one view, despite the long durations of full occlusion.

## 6.3 Performance Evaluation of the Proposed Scheme

To quantitatively evaluate the proposed tracking scheme as well as the effectiveness of the proposed online learning, Case-g is studied. Since the occlusion is this case is synthetically added, the ground truth is available. Tests are done separately for the proposed trackers with and without online learning.

*Online Learning*: Fig. 15 shows the Bhattacharyya coefficient $\rho(p, q)$ in

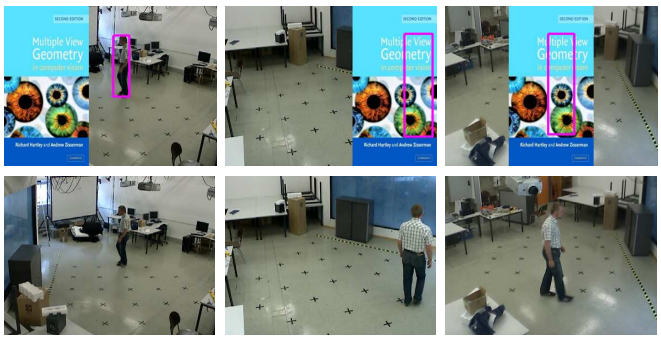


**Figure 14:** Proposed tracker: tracking results on videos in Case-g. Videos are obtained from view 1-3 of TU Graz Multi-Camera dataset, where an object experiences long-term full occlusion. Tracked boxes are marked in images (magenta). Row 1, Column 1-3: results in views 1-3. Row 2: corresponding ground truth for each view. Key frame # 533 is shown.

(8) between the single-view tracker and the reference object in each view. Observing Fig. 15, one can also see that $T$ is relatively easy to determine.

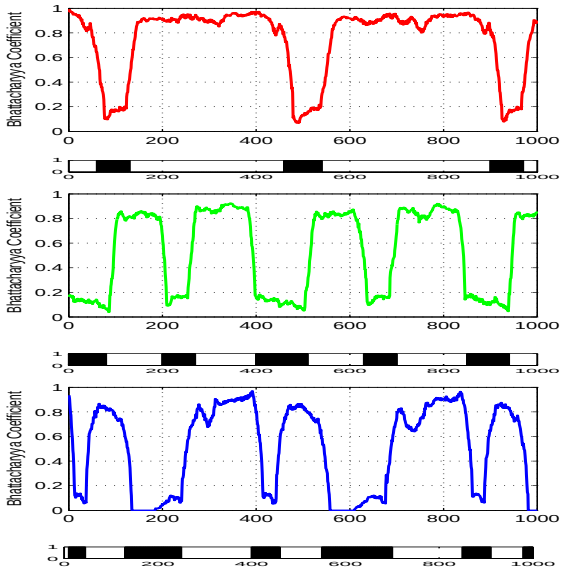



**Figure 15:** Evaluation of proposed tracker: Bhattacharyya coefficients versus frame numbers on videos from Case-g (views 1-3 of TU Graz Multi-Camera dataset), computed between the output of 1st layer tracker in Fig.4 and the reference object. Top to bottom plots: from views 1-3. The black bar indicates ground truth occlusion frames.

Fig. 16 shows selected tracking results within the tracked boxes from the proposed scheme with/without online learning. One can see that with online learning, the proposed tracker shows good tracking with visibly better tracking performance as compared with the one without online learning. Table 2 shows the mean and standard deviation of Euclidean distance in (15), Bhattacharyya distance in (16) and geodesic distance in (1) on videos in Case-g. Fig. 17 compares the error curves under each criterion averaged over views for Case-g with and without online learning. Observing Table 2 and Fig. 17, one can see that under each criterion, good performance is obtained, indicated by small error values with small variances, despite the frequent occurrences of full occlusions. Further, results show that online learning of object appearances has improved the tracking performance.

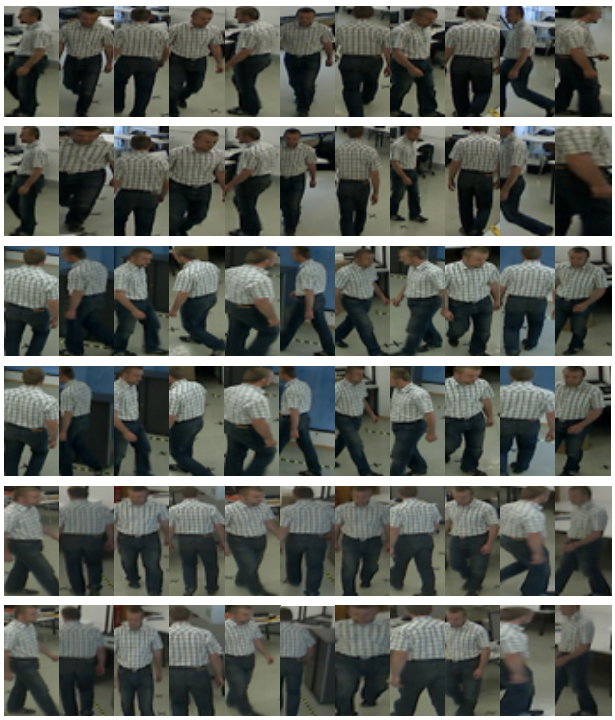


**Figure 16:** Proposed tracker: tracking results with/without online learning on videos in Case-g. Videos are obtained from views 1-3 of TU Graz Multi-Camera dataset). Odd rows: with online learning in 3 views; Even Rows: without online learning in 3 views. Key video frames (# 1, 100, 199, 298, 397, 496, 595, 694, 793, 892, 991) are selected in the figure. Note that in the 2nd row, synthetically added occlusions are removed for easy inspection on whether tracked box is related to a true object location.

**Table 2:** Evaluation of proposed tracker: tracking errors using the tracker with/without online learning on videos in Case-g (views 1-3 of TU Graz Multi-Camera datasets). Three objective criteria are used to calculate the tracking errors: Euclidean distance in (15), Bhattacharyya distance in (16) and geodesic distance in (1). The results are averaged over all frames for each view. OL: online learning; NOL: no online learning.

(a) Euclidean distance

| | Mean | | | Standard deviation | | |
|---|---|---|---|---|---|---|
| | View 1 | View 2 | View 3 | View 1 | View 2 | View 3 |
| OL | 9.3302 | 13.2432 | 9.5677 | 4.6573 | 5.0377 | 3.5362 |
| NOL | 22.080 | 24.4718 | 20.480 | 12.289 | 16.171 | 15.442 |

(b) Bhattacharyya distance

| | Mean | | | Standard deviation | | |
|---|---|---|---|---|---|---|
| | View 1 | View 2 | View 3 | View 1 | View 2 | View 3 |
| OL | 0.2950 | 0.3700 | 0.3608 | 0.0545 | 0.0675 | 0.0836 |
| NOL | 0.3348 | 0.4602 | 0.4611 | 0.0925 | 0.1286 | 0.1260 |

(c) Geodesic distance

| | Mean | | | Standard deviation | | |
|---|---|---|---|---|---|---|
| | View 1 | View 2 | View 3 | View 1 | View 2 | View 3 |
| OL | 1.7916 | 1.8620 | 1.7404 | 0.3818 | 0.3605 | 0.4015 |
| NOL | 2.1127 | 2.7277 | 2.6967 | 0.6733 | 1.0778 | 1.1419 |

## 6.4 Comparisons with Three Existing Trackers

The proposed tracker is compared with three existing multi-camera tracking methods:

- *Tracker-1* in [9], which uses particle filter-based tracking and pixel difference-based occlusion detection;
- *Tracker-2* in [10], which uses particle swarm optimization-based tracking and covariance distance-based occlusion detection.
- *Tracker-3* in [33], which performs object tracking through detection and multiview co-training.

*Qualitative Comparisons*: Fig. 18 (Case-a) shows the tracking results on occluded view of the two-view videos in an outdoor environment on campus, where a person is walking along the road, by the proposed tracker and *Tracker-1* in [9]. The person is fully occluded by a tree and does not reappear. Comparing the results in Fig. 18, one can see that the proposed tracker and *Tracker-1* performs similarly well. Fig. 19 (Case-d) shows the tracking results on the two-view videos in an outdoor environment on campus, where multiple persons are walking around, by the proposed tracker

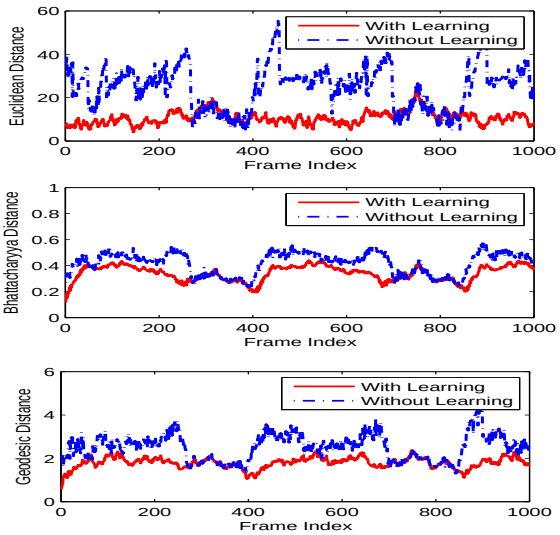



**Figure 17:** Evaluation of proposed tracker: tracking errors with/without online learning on videos in Case-g. Videos are obtained from views 1-3 of TU Graz Multi-Camera datasets. Tracking errors are evaluated from: (Top) Euclidean distance in (15); (Middle) Bhattacharyya distance in (16); (Bottom) Geodesic distance in (1). Curve values are averaged over all views.

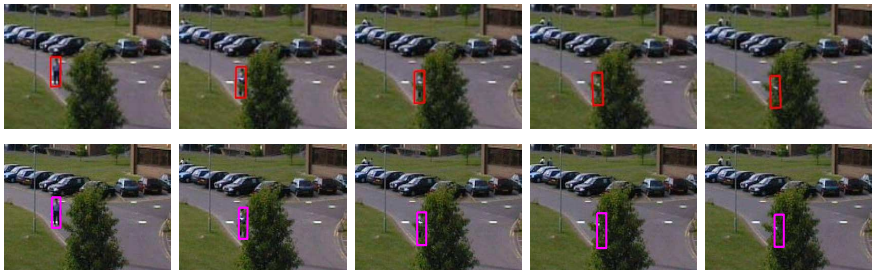


**Figure 18:** Comparison: tracking results from the proposed tracker and *Tracker-1* on two-view videos in Case-a. Videos are obtained from views 1-2 of testing set of PETS'01 S3, where an object experiences long-term full occlusion (only the occluded view is shown). Row 1: from *Tracker-1* [9] (red box); Row 2: from the proposed tracker (magenta box). Key frames (# 5019, 5074, 5095, 5105, 5118) in the video are selected (zoomed in for better inspection).

and *Tracker-2* in [10]. The target person moves very fast, and experiences short-term occlusion and intersections with other persons. Comparing these results, one can see from Fig. 19 that the proposed tracker performs better than *Tracker-2* in terms of tracked object boxes, where both views contain occlusions.

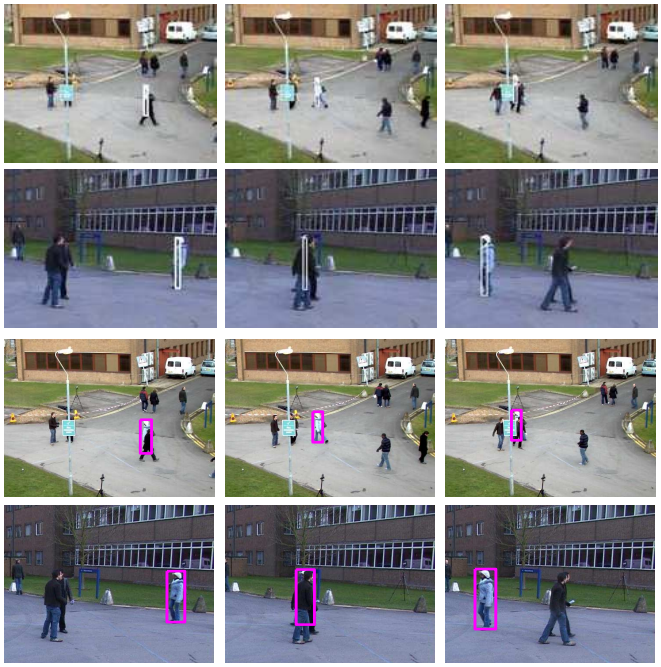


**Figure 19:** Comparison: Tracking results from the proposed tracker and *Tracker-2* on two-view videos in Case-d. Videos are obtained from view 1 and 5 of PETS'09 S2.L1, where an object experiences intersections. Row 1-2: from *Tracker-2* in [10] (white box) for the 2 views. Row 3-4: from the proposed tracker (magenta box) for the 2 views. Key frames (# 130, 145, 151) in the video are selected (zoomed in for better inspection of target objects).

Fig. 20 (Case-h) shows the tracking results on the three-view videos in an indoor environment, where multiple persons are moving around, by the proposed tracker and *Tracker-3* in [33]. The target persons move frequently, and experience short-term occlusion and intersections with each other. Since the proposed multi-view tracking scheme is designed to track individual object through videos, the results in rows 4-6 of Fig. 20 is a superposition of individual object tracking results. Comparing the results, one can see that the proposed tracker performs somewhat better than *Tracker-3* in terms of tightness of tracked object bounding boxes.

Fig. 21 (Case-i) shows the tracking results on the three-view videos in an outdoor environment on campus, where multiple persons are walking around, by the proposed tracker and *Tracker-3* in [33]. The target persons experience short-term and long-term occlusions and intersections with each other. Comparing these results, one can see that the proposed tracker performs somewhat better than *Tracker-3* in terms of tightness of tracked object boxes.

**Table 3:** Quantitative performance comparisons: Comparing tracking errors between the proposed tracker and *Tracker-2* on Case-i videos, and between the proposed tracker and *Tracker-3* on Case-h and Case-i videos. Sub-tables (a)-(c) show error values obtained from using the criterion of Euclidean distance in (15), Bhattacharyya distance in (16), and Geodesic distance in (1), respectively. Error values are averaged over all objects and all key frames (i.e. video frames shown in Fig. 19, Fig. 20 and Fig. 21) in each view.

(a) Euclidean distance

| Videos Case-d | View 1 | View 2 | Average |
|---|---|---|---|
| Proposed | 5.3115 | 5.2957 | 5.3036 |
| *Tracker-2* | 12.243 | 16.153 | 14.198 |

| Videos Case-h,i | View 1 | View 2 | View 3 | Average |
|---|---|---|---|---|
| Proposed | 8.1616 | 6.4158 | 6.4999 | 7.0258 |
| *Tracker-3* | 119.89 | 103.40 | 101.37 | 108.22 |
| Proposed | 6.0699 | 6.9204 | 7.3095 | 6.7666 |
| *Tracker-3* | 17.890 | 20.696 | 32.516 | 23.701 |

(b) Bhattacharyya distance

| Videos Case-d | View 1 | View 2 | Average |
|---|---|---|---|
| Proposed | 0.1665 | 0.0782 | 0.1224 |
| *Tracker-2* | 0.3374 | 0.2743 | 0.3058 |

| Videos Case-h,i | View 1 | View 2 | View 3 | Average |
|---|---|---|---|---|
| Proposed | 0.0639 | 0.0506 | 0.0503 | 0.0549 |
| *Tracker-3* | 0.4364 | 0.3360 | 0.2917 | 0.3547 |
| Proposed | 0.0689 | 0.1003 | 0.0608 | 0.0767 |
| *Tracker-3* | 0.2810 | 0.2678 | 0.3093 | 0.2860 |

(c) Geodesic distance

| Video Case-d | View 1 | View 2 | Average |
|---|---|---|---|
| Proposed | 1.2982 | 0.8992 | 1.0987 |
| *Tracker-2* | 3.4105 | 3.4654 | 3.4379 |

| Videos Case-h,i | View 1 | View 2 | View 3 | Average |
|---|---|---|---|---|
| Proposed | 0.5536 | 0.3805 | 0.5428 | 0.4923 |
| *Tracker-3* | 2.8850 | 2.5056 | 2.6074 | 2.6660 |
| Proposed | 0.9119 | 0.9196 | 0.8058 | 0.8791 |
| *Tracker-3* | 1.7674 | 1.4407 | 1.8271 | 1.6784 |

*Quantitative Comparisons*: To make quantitative comparisons between the proposed tracking scheme and *Tracker-2* and *Tracker-3*, three objective criteria, *Euclidean distance* in (15), *Bhattacharyya distance* in (16) and *Geodesic distance* in (1), are applied. Table 3 compares the proposed tracking scheme and *Tracker-2*, *Tracker-3* under the criterion of Euclidean distance in (15), Bhattacharyya distance in (16) and geodesic distance in (1) on videos in Case-d, h, i. Observing Table 3, one can see that under each

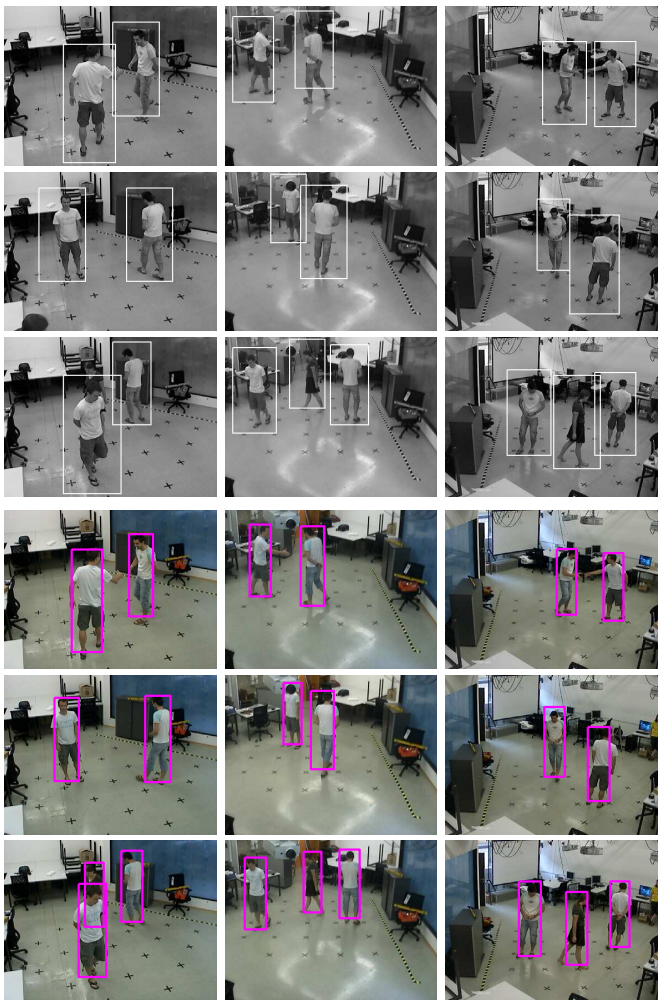


**Figure 20:** Comparison: Tracking results from proposed tracker and *Tracker-3* on 3-view videos in Case-h. Videos are obtained from views 1-3 of TU Graz Multi-Camera dataset, where objects experiences intersections. Row 1-3: from *Tracker-3* [33] (white box) for views 1-3 (Column 1-3). Row 4-6: from the proposed tracker (magenta box) for views 1-3 (Column 1-3). The video frames (# 510, 1089, 1481) are chosen the same as those given in *Tracker-3*.

criterion, better performance is obtained by the proposed tracker, indicated by small error values. It is worth mentioning that, since tracking in *Tracker-3* is based on detection and learning classifiers that is rather different from the proposed scheme, performance comparisons only give some rough indications.

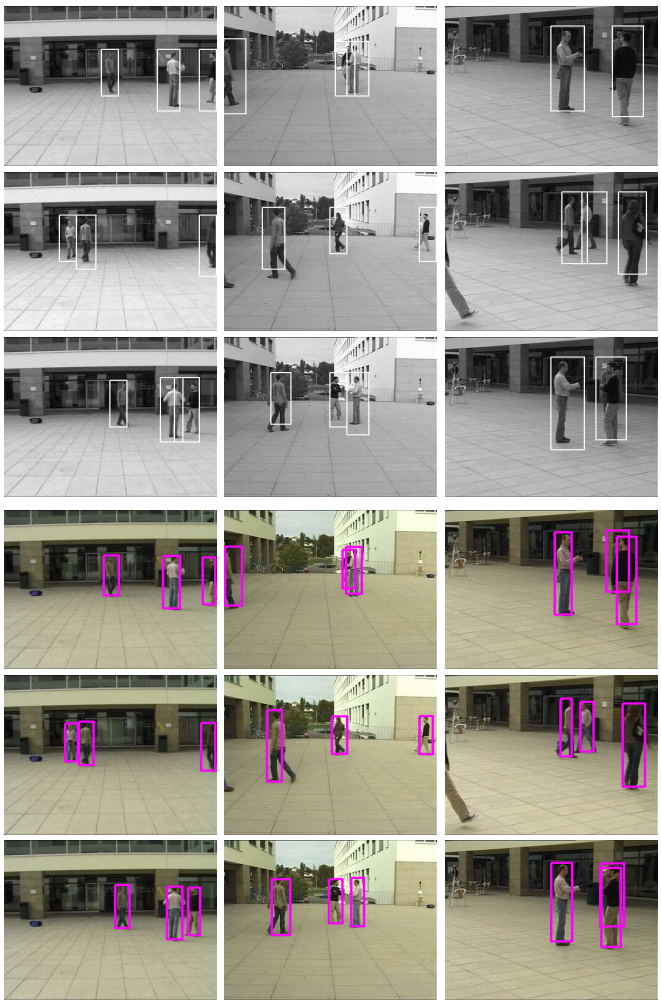


**Figure 21:** Comparison: Tracking results from proposed tracker and *Tracker-3* on 3-view videos in Case-i. Videos are obtained from views 1-3 of EPFL Multi-Camera dataset, where objects experiences intersections. Row 1-3: from *Tracker-3* [33] (white box) for view 1-3 (Column 1-3). Row 4-6: from the proposed tracker (magenta box) for view 1-3 (Column 1-3). The video frames ( # 511, 692, 741) are chosen the same as those given in *Tracker-3*.

# 7 Conclusion

The proposed multi-view tracker, through mapping tracked object between camera views, maximum likelihood estimation based on geodesics, and online learning on the Riemannian manifold, is tested on videos containing long-term full occlusion. Test results have shown the effectiveness and ro-

bustness of the proposed tracking and online learning scheme, in terms of tracking drifts and bounding box accuracy especially for long-term full occlusion scenarios. Performance of the proposed tracker is evaluated using three criteria, and qualitative and quantitative comparisons with three existing multi-view tracking methods have been made which provided further support to the proposed scheme. Future work will be conducted on extensive tests and evaluation on videos where occluding and target objects have similar appearances.

# References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, article 13, Dec. 2006.

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

[3] G. Chao, S. Jeng, and S. Lee, "An improved occlusion handling for appearance-based tracking," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 465–468, Brussels, Belgium, Sept. 11 - 14, 2011.

[4] S. Kwak *et al.*, "Learning occlusion with likelihoods for visual tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 1551–1558, Barcelona, Spain, Nov. 6 - 13, 2011.

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

[6] A. Mittal and L.S. Davis, "$M_2$Tracker: 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.

[7] C. Chu *et al.*, "Tracking across multiple cameras with overlapping views based on brightness and tangent transfer functions," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–6, Ghent, Belgium, Aug. 22 -25, 2011.

[8] W. Qu, D. Schonfeld, and M. Mohamed, "Decentralized multiple camera multiple object tracking," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pp. 245–248, Toronto, Canada, Jul. 9 - 12, 2006.

[9] Z. Yue, S.K. Zhou, and R. Chellappa, "Robust two-camera tracking using homography," in *Proceedings of IEEE International Conference*

*on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1–4, Montreal, Canada, May 17 - 21, 2004.

[10] B. Kwolek, "Multi camera-based person tracking using region covariance and homography constraint," in *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 294–299, Boston, USA, Aug. 29 - Sept. 1, 2010.

[11] Y. Yun, I.Y.H. Gu, H. Aghajan, "Maximum-likelihood object tracking from multi-view video by combining homography and epipolar constraints," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–6, Hong Kong, Oct. 30 - Nov. 2, 2012.

[12] W.M. Boothby, "An Introduction to Differentiable Manifolds and Riemannian Geometry," *Academic Press*, edition 2, 2002.

[13] J.M. Lee, "Introduction to Smooth Manifolds," *Springer*, edition 1, 2002.

[14] P.A. Absil, R. Mahony, and R. Sepulchre, "Optimization Algorithms on Matrix Manifolds," *Princeton University Press*, 2008.

[15] J. Gallier, "Notes on differential geometry and Lie groups," *Technical report, Department of Computer and Information Science, University of Pennsylvania*, USA, 2010.

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

[17] Y. Wu *et al.*, "Real-time visual tracking via incremental covariance tensor learning," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 1631–1638, Kyoto, Japan, Sept. 29 - Oct. 2, 2009.

[18] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *Proceedings of European Conference on Computer Vision (ECCV)*, vol. 2, pp. 589–699, Graz, Austria, May 7 - 13, 2006.

[19] Z.H. Khan and I.Y.H. Gu, "Bayesian online learning on Riemannian manifolds using a dual model with applications to video object tracking," in *Proceedings of IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition (in conjunction with ICCV)*, pp. 1402–1409, Barcelona, Spain, Nov. 13, 2011.

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

[21] A. Arsigny *et al.*, "Geometric means in a novel vector space structure on symmetric-positive definite matrices," *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, vol. 29, no. 1, pp. 328–347, 2007.

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

[23] S. Calderara, A. Prati, and 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.

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

[25] M.S. Arulampalam *et al.*, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[26] Z.H. Khan, I.Y.H. Gu, and A.G. Backhouse, "Robust visual object tracking using multi-mode anisotropic mean shift and particle filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, issue 1, pp. 74–87, 2011.

[27] "PETS 2001 benchmark data," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (in conjunction with CVPR)*, Kauai, Hawaii, USA, Dec. 9, 2001. [Online]. Available: http://www.cvg.cs.rdg.ac.uk/PETS2001/

[28] "PETS 2006 benchmark data," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (in conjunction with CVPR)*, New York, USA, Jun. 18, 2006. [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2006/

[29] "PETS 2007 benchmark data," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (in conjunction with ICCV)*, Rio de Janeiro, Brazil, Oct. 14, 2007. [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2007/

[30] "PETS 2009 benchmark data," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (in conjunction with CVPR)*, Miami, USA, Jun. 25, 2009. [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2009/

[31] "Multi-camera datasets," *Graz University of Technology*, Austria. [Online]. Available: http://lrs.icg.tugraz.at/download.php

[32] "Multi-camera pedestrian video," *Swiss Federal Institute of Technology in Lausanne (EPFL)*, Switzerland. [Online]. Available: http://cvlab.epfl.ch/data/pom

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

# Paper B

## Multi-View Face Pose Classification by Boosting with Weak Hypothesis Fusion Using Visual and Infrared Images

Yixiao Yun, and Irene Yu-Hua Gu

Published in
*Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*
pp. 1949–1952,
Kyoto, Japan, Mar. 25 - 30, 2012

*The layout has been revised.*

**Abstract**

This paper proposes a novel method for multi-view face pose classification through sequential learning and sensor fusion. The basic idea is to use face images observed in visual and thermal infrared (IR) bands, with the same sampling weight in a multi-class boosting structure. The main contribution of this paper is a multi-class AdaBoost classification framework where information obtained from visual and infrared bands interactively complement each other. This is achieved by learning weak hypothesis for visual and IR band independently and then fusing the optimized hypothesis sub-ensembles. In addition, an effective feature descriptor is introduced to thermal IR images. Experiments are conducted on a visual and thermal IR image dataset containing 4844 face images in 5 different poses. Results have shown significant increase in classification rate as compared with an existing multi-class AdaBoost algorithm SAMME trained on visual or infrared images alone, as well as a simple baseline classification-fusion algorithm.

# 1 Introduction

Multi-view face pose classification has drawn increasing research interest in recent years, largely driven by many applications such as robotic surveillance [1], monitoring of driver attentiveness [2] or automating camera management [3].

Several face pose classification methods have been proposed and developed recently. Guo *et al.* [4] use PCA-based face features and soft margin AdaBoost to detect the frontal views. Baluja *et al.* [5] extract features inspired by [6] and build five separate AdaBoost classifiers for face images in each class. Huang *et al.* [7] present a nested cascade detector for face poses in 5 classes using confidence-rated AdaBoost [8] based on Haar features. Yang *et al.* [9] introduce a tree-structured classifier for face poses in 7 classes, and each node is a three-class classifier trained by AdaBoost.MH. Islam *et al.* [10] suggest a subspace learning approach for feature extraction and classify five different face poses by k-NN technique. Good results have been achieved, however, these methods mainly adopt one-against-all or one-against-one strategies for multi-class problems, so model complexities may be increased.

To improve the classification of objects, approaches are proposed on fusion of visual and infrared information. Hanif and Ali [11], Ulusoy and Yuruk [12], Wang and Li [13] each present a fusion method at the sensor level. Neagoe *et al.* [14] use decision fusion of neural classifiers for real time face recognition. Apatean *et al.* [15] introduce fusion scheme at different levels for SVM-based obstacle classification. These methods usually combine multiple individual features or decisions in a one-off manner, however, the interactive relations between visual and infrared observations are seldom considered. Despite these efforts, classifying face poses using both visual and infrared observations remains an open issue.

To tackle these problems, we propose a novel method fusing visual and infrared information interactively within a boosting framework for multi-view face pose classification. Different from one-against-all or one-against-one strategies, our model is similar to SAMME [16] in true solution to multi-class problems, however, a new part of sensor fusion is introduced. The main contributions of this paper include using sub-ensemble learning for fused hypothesis optimization and suggesting effective feature for thermal IR image. Improved classification results are demonstrated by empirical evaluation compared with SAMME using visual or infrared images alone, as well as a simple baseline classification-fusion algorithm.

The rest of this paper is organized as follows: Section 2 gives a big picture of the proposed framework; Section 3 makes some review of AdaBoost algorithms; Section 4 describes our fusion strategy; Section 5 describes feature extraction for thermal IR images; Section 6 shows experiment results on a visual and thermal IR image dataset and comparisons with most relevant

existing method; finally Section 7 concludes the paper.

## 2 Problem Formulation: The Big Picture

As shown in Fig. 1, the proposed framework consists of three major parts: (a) independent weak hypothesis learning using visual and infrared features with the same sampling weight; (b) fusion by optimizing hypothesis sub-ensemble; (c) adding sub-ensemble to a final strong classifier and updating sampling weight distribution with a scale factor.

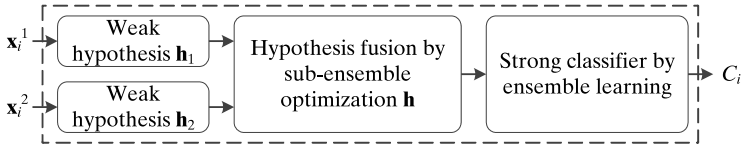



**Figure 1:** Block diagram of proposed scheme. The dashed box represents boosting structure. The notations $\mathbf{x}_i^1$, $\mathbf{x}_i^2$, $C_i$ denote visual features, infrared features, and predicted class labels of $i$-th object, respectively.

The essence for using the same sampling weight is to force weak classifiers for both visual and infrared bands to focus on the same objects, therefore weak hypotheses independently learned from visual and infrared features match each other. The basic idea for hypothesis optimization is to add hypotheses for both bands to the sub-ensemble, with sub-ensemble weights according to their accuracies, so that hypothesis sub-ensemble may have enhanced performance based on fusion of visual and infrared information. In this way, the final strong ensemble may have further improved accuracy. The main motivation for using a scale factor to update sampling weights is to make weak classifiers focus on those difficult objects misclassified in both visual and infrared bands. The main novelty lies in two-stage ensemble learning within multi-class boosting framework, by using visual and infrared information in this interactive manner, which may lead to better classification results.

## 3 AdaBoost: Review

This section briefly reviews AdaBoost algorithms, with emphasis on SAMME, which our proposed classification method is built upon.

AdaBoost is an ensemble learning method originally intended only for binary problems. Many extensions of AdaBoost for multi-class problems exist, and most of them have been restricted to using one-against-all or one-against-one strategies [17]. SAMME, one of the true multi-class AdaBoost

algorithms, is a true multi-class classifier that solves multi-class problems without reducing them to multiple binary subproblems.

Let $\mathbf{X} = \{\boldsymbol{x}_i\}$, $i = 1, 2, \cdots, N$ be the entire training set containing feature vectors of objects. Let the class label (denoted by $c$) be represented as a $K$-dimensional vector $\mathbf{y} = (y_1, y_2, \cdots, y_K)^T$, where $y_k = 1$ if $c = k$, otherwise $y_k = -1/(K-1)$, $k \in \{1, 2, \cdots, K\}$ and $K \geq 3$ is the number of classes. In such a way, $\mathbf{Y} = \{\mathbf{y}_i\}$ is an equivalent set of class labels corresponding to $\mathbf{X}$. The output of weak classifier for each feature vector is encoded in the same way as the weak hypothesis $\mathbf{h} = (h_1, h_2, \cdots, h_K)^T$.

The goal is to minimize the objective function as exponential loss function $L(\mathbf{Y}, \mathbf{H}) = \sum_{i=1}^{N} \exp\left(-\frac{1}{K}\mathbf{y}_i^T \mathbf{H}(\boldsymbol{x}_i)\right)$ by learning a strong ensemble

$$\mathbf{H}^{(t)}(\boldsymbol{x}_i) = \mathbf{H}^{(t-1)}(\boldsymbol{x}_i) + \alpha^{(t)}\mathbf{h}^{(t)}(\boldsymbol{x}_i) \tag{1}$$

subject to the constraint $\sum_{k=1}^{K} H_k(\boldsymbol{x}_i) = 0$. Several boosting rounds $t = 1, \cdots, T$ is applied. In each boosting round, the sampling weight $D_i^{(t)}$ for each feature vector of objects, weighted errors $\epsilon^{(t)}$ for the weak classifier and the ensemble weight $\alpha^{(t)}$ for each hypothesis that is added to the ensemble are updated as follows:

$$D_i^{(t)} = \exp\left(-\frac{1}{K}\mathbf{y}_i^T \mathbf{H}^{(t-1)}(\boldsymbol{x}_i)\right) \tag{2}$$

$$\epsilon^{(t)} = \sum_{i=1}^{N} D_i^{(t-1)} \mathbb{I}(\mathbf{y}_i^T \mathbf{h}^{(t)}(\boldsymbol{x}_i) \leq 0) / \sum_{i=1}^{N} D_i^{(t-1)} \tag{3}$$

$$\alpha^{(t)} = \frac{(K-1)^2}{K}\left(\log\frac{1-\epsilon^{(t)}}{\epsilon^{(t)}} + \log(K-1)\right) \tag{4}$$

where $\mathbb{I}(A)$ is an indicator function which equals 1 if event $A$ is true, and 0 otherwise.

# 4 Multi-Class Boosting with Weak Hypothesis Fusion

A sub-ensemble learning method fusing weak hypotheses learned from visual and infrared features under multi-class AdaBoost framework is introduced in this section. Each object feature vector $\boldsymbol{x}_i$ contains two component feature vectors $\{\mathbf{x}_i^1, \mathbf{x}_i^2\}$, corresponding to visual and infrared bands, respectively.

In the proposed method, we enforce a same set of sampling weights to the weak classifiers for both visual and infrared bands on the same objects, therefore weak hypotheses independently learned from visual and infrared features match each other, yielding $\mathbf{h}_m(\mathbf{x}_i^m)$, $m = 1, 2$. Different from multiple AdaBoost classifiers trained on single-band features with independent

sampling weights, the interaction between visual and infrared information in our case is conducted at each boosting round inside the boosting structure.

The objective criterion of the proposed scheme is to minimize the exponential loss function

$$L(\mathbf{Y}, \mathbf{h}) = \sum_{i=1}^{N} \exp\left(-\frac{1}{K}\mathbf{y}_i^T \mathbf{h}^{(t)}(\boldsymbol{x}_i)\right) \tag{5}$$

through learning a sub-ensemble of weak hypotheses

$$\mathbf{h}^{(t)}(\boldsymbol{x}_i) = \sum_{m=1}^{M} \beta_m^{(t)} \mathbf{h}_m^{(t)}(\mathbf{x}_i^m) \tag{6}$$

subject to the constraints $\sum_{k=1}^{K} h_k(\boldsymbol{x}_i) = 0$ and $\sum_{m=1}^{M} \beta_m^{(t)} = 1$, $M = 2$. The solution is shown to be:

$$\beta_m^{(t)} = \frac{\log\left(\frac{1-\epsilon_m^{(t)}}{\epsilon_m^{(t)}}(K-1)\right)}{\log\left((K-1)^M \prod_{m=1}^{M} \frac{1-\epsilon_m^{(t)}}{\epsilon_m^{(t)}}\right)} \tag{7}$$

where

$$\epsilon_m^{(t)} = \sum_{i=1}^{N} \mathbb{I}(\mathbf{y}_i^T \mathbf{h}_m^{(t)}(\mathbf{x}_i^m) \leq 0)/N \tag{8}$$

$\beta_m$ is the sub-ensemble weight for each single-band weak hypothesis that is added to the sub-ensemble and $\epsilon_m$ is the error rate for each single-band weak hypothesis.

A scale factor $\gamma_i^{(t)}$ is then introduced for $\boldsymbol{x}_i$, which is exponentially proportional to the count of misclassification by the two weak classifiers

$$\gamma_i^{(t)} = 2^{\eta_i} \tag{9}$$

where $\eta_i = \sum_{m=1}^{M} \mathbb{I}(\mathbf{h}_m^{(t)}(\mathbf{x}_i^m) \neq \mathbf{y}_i)$. In such a way, objects correctly classified by weak classifiers in both visual and infrared bands lose more weights, and objects misclassified by both weak classifiers are treated as difficult objects by gaining more weights:

$$D_i^{(t)} = \gamma_i^{(t)} D_i^{(t-1)} \exp\left(-\frac{1}{K}\beta^{(t)} \mathbf{y}_i^T \mathbf{h}^{(t)}(\boldsymbol{x}_i)\right) \tag{10}$$

Table 1 summarizes the pseudo code of the proposed scheme.

**Table 1:** Pseudo code of multi-class boosting with weak hypothesis fusion.

**Training process:**
**Input:** training set $\mathbf{X}$, label set $\mathbf{Y}$ and #iteration $T$.

1. **Initialization:** sampling weights $D_i^{(0)} = 1/N, i = 1, 2, \cdots, N$ and ensemble $\mathbf{H}^{(0)}(\boldsymbol{x}_i) = \mathbf{0} \in \mathbb{R}^K$.
2. **For** $t = 1$ to $T$ (boosting round):
   (a) Learn single-band weak hypothesis $\mathbf{h}_m^{(t)}(\mathbf{x}_i^m)$ with $D_i^{(t-1)}$;
   (b) Compute error rate $\epsilon_m^{(t)}$ for each single-band weak hypothesis by (8);
   (c) Set sub-ensemble weight $\beta_m^{(t)}$ by (7);
   (d) Fuse optimized hypothesis sub-ensemble $\mathbf{h}^{(t)}(\boldsymbol{x}_i)$ by (6);
   (e) Compute weighted error for fused hypothesis $\epsilon^{(t)}$ by (3);
   (f) Set ensemble weight for fused hypothesis $\alpha^{(t)}$ by (4);
   (g) Add to ensemble $\mathbf{H}^{(t)}$ according to (1);
   (h) Update sampling weights $D_i^{(t)}$ by (10) and re-normalize, where the scale factor $\gamma_i^{(t)}$ is obtained from (9);

   **End**

**Output**: parameters of trained classifier $\mathbf{h}_m^{(t)}$, $\beta_m^{(t)}$, $\alpha^{(t)}$.

**Testing process:**
**Input:** a new pair of test images $\boldsymbol{x}_j$ in the test set
**Repeat:** the boosting round in Step 2 above, using fixed classifier's parameters obtained from the training process;
**Output**: class label $k^* = \arg\max_k H_k(\boldsymbol{x}_j)$ for $\boldsymbol{x}_j$.

# 5 Feature Descriptor for IR Image

Thermal infrared images present different characteristics from those in visual band images, e.g. blurred edges and lack of texture information (as shown in Fig. 3).

**Figure 2:** A bank of Gabor wavelets in (a) frequency domain and (b) real parts in spatial domain.

Viewing the special nature of thermal IR images, special feature descriptors that are effective should be explored. We propose in this paper to use

Gabor wavelet features. The idea here is that a bank of Gabor wavelets with appropriately specified frequency bands and orientations is used to characterize an IR image, which may extract salient features in thermal IR images due to the spatial locality, frequency selectivity and orientation selectivity [18]. DC component is added as a feature component covering the lower frequency band. Fig. 2 shows the Gabor wavelets in frequency domain and real part in spatial domain. To further reducing the feature dimension, we then apply PCA (principal component analysis) to Gabor features from each IR image.

# 6 Experimental Results

**Dataset:** A total of 2422 visual and 2422 thermal infrared images are used. Detail about the dataset split to each class is given in Table 2. Fig. 3 shows some example images.

**Table 2:** Visual and thermal IR face image dataset containing five poses.

| Face pose | #Visual images | #IR images |
|---|---|---|
| Front | 506 | 506 |
| Left | 500 | 500 |
| Right | 500 | 500 |
| Up | 456 | 456 |
| Down | 460 | 460 |

**Figure 3:** Example face images of visual and thermal IR bands with five poses.

**Setup:** All face images are manually cropped and normalized to $32 \times 32$ pixels in gray-scale images. Gabor wavelets with 3 frequency bands (1.5 octave bandwidth) are used for extracting visual and infrared features. The number of orientations is 8 for each image. The down-sampling rate is 4

in each (horizontal/vertical) direction. PCA is applied to Gabor feature vectors retaining average of 95% energy. Images in the dataset are partitioned into 2 sets, i.e. 60% of images in each class are used for training, the remaining 40% are used for testing.

**Results and comparisons:** Table 3 and 4 show the classification results from the proposed scheme on the testing set by using visual and thermal IR images as compared with (a) SAMME using visual images only; (b) SAMME using infrared images only; (c) a simple baseline classification-fusion algorithm. The simple baseline classification-fusion algorithm is obtained by training sub-classifiers $\mathbf{H}_m(\mathbf{x}_i^m)$ with independent sampling weights for visual features $\mathbf{x}_i^1$ and IR features $\mathbf{x}_i^2$. The class label is then determined according to $k^* = \arg\max_{m,k} H_{m,k}(\mathbf{x}_i^m)$, where $H_{m,k}(\mathbf{x}_i^m)$ is the $k$-th element of sub-classifier $\mathbf{H}_m(\mathbf{x}_i^m)$. Fig. 3 shows the curves of the corresponding classification error as a function of boosting rounds for all these four cases on the testing set.

**Table 3:** Comparison of different methods: average classification rate on the testing set (V: Visual).

| Method | Dataset | Classification rate (%) |
|---|---|---|
| SAMME(V) | Visual | 87.31 |
| SAMME(IR) | IR | 92.44 |
| Baseline classification-fusion | Visual+IR | 93.90 |
| Proposed | Visual+IR | **96.20** |

**Table 4:** Comparison of different methods: false positive rate and false negative rate for each class on the testing set.

| False positive rate (%) | | | | | |
|---|---|---|---|---|---|
| Method | Front | Left | Right | Up | Down |
| SAMME(V) | 14.01 | 10.70 | 9.55 | 13.35 | 16.14 |
| SAMME(IR) | 12.18 | 5.30 | 3.70 | 8.30 | 8.42 |
| Baseline classification-fusion | 12.38 | 3.00 | 5.50 | 4.95 | 4.35 |
| Proposed | **6.09** | **2.20** | **1.90** | **4.62** | **4.29** |
| False negative rate (%) | | | | | |
| Method | Front | Left | Right | Up | Down |
| SAMME(V) | 15.10 | 7.03 | 7.28 | 11.35 | 22.19 |
| SAMME(IR) | 13.15 | 3.02 | 1.78 | 9.64 | 9.80 |
| Baseline classification-fusion | 9.23 | 1.52 | 3.08 | 8.95 | 7.85 |
| Proposed | **7.33** | **1.66** | **0.71** | **2.85** | **6.38** |

Results from Table 3 and 4 show that the proposed classifier improves the average classification rate as comparing with SAMME(V), SAMME(IR) and the baseline fusion-classifier. Observing Fig. 4 shows that the proposed classifier has a fast convergence speed with the lowest classification errors. Further, Fig. 5 shows that using the Gabor feature descriptor for IR images is very efficient in the proposed classifier. It allows very low dimensional features for IR images without significantly reducing the final classification rate.

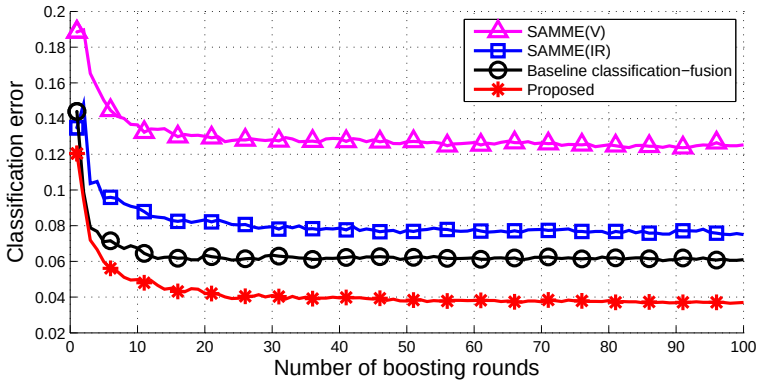



**Figure 4:** Classification errors vs. boosting round for the proposed classifier and 3 other classifiers on the testing set.

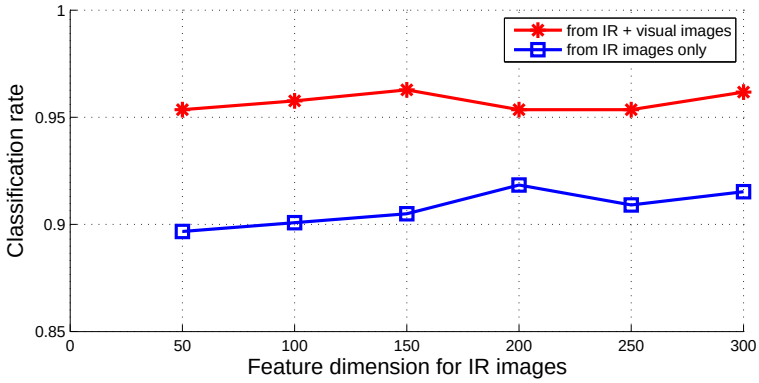



**Figure 5:** Dimension of IR image features vs. the average classification rate. Red curve: final classification rate from proposed scheme when the feature dimension of IR images changes meanwhile the feature dimension of visual band (386 in our tests) is fixed; Blue curve: the classification rate when the classifier only uses IR images with specified feature dimension.

# 7 Conclusion

The proposed multi-class classification method, using fused hypotheses from visual and IR information in a unified multi-class AdaBoost, is shown to be effective in obtaining high classification rate with low false alarm in our experiments. Our results have also shown that the proposed feature descriptor for IR images is very effective. Comparison with an existing and most relevant AdaBoost algorithm SAMME on visual or IR face image dataset alone as well as a baseline classification-fusion algorithm has provided fur-

ther evidence on the effectiveness of the proposed method. Future work will be conducted on testing on more datasets.

## References

[1] K.B.J. Axnick and R. Jarvis, "Face and pose recognition for robotic surveillance," in *Proceedings of Australian Conference on Robotics and Automation*, pp. 1–9, Sydney, Australia, Dec. 5 - 7, 2005.

[2] X. Liu, Y. Zhu, and K. Fujimura, "Real time pose classification for driver monitoring," in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, pp. 174–178, Singapore, Sept. 3 - 6, 2002.

[3] Q. Liu, Y. Rui, A. Gupta, and J.J. Cadiz, "Automating camera management for lecture room environment," in *Proceedings of ACM CHI International Conference on Computer Human Interaction*, pp. 442–449, Hague, Netherlands, Apr. 1 - 6, 2000.

[4] Y. Guo *et al.*, "Soft margin adaboost for face pose classification," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 221–224, Hong Kong, Apr. 6 - 10, 2003.

[5] S. Baluja, M. Sahami, and H.A. Rowley, "Efficient face orientation discrimination," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 589–592, Singarpore, Oct. 24 - 27, 2004.

[6] P. Viola and M. Jones, "Robust real-time object detection," *International Journal on Computer Vision (IJCV)*, vol. 57, no. 2, pp. 137–154, 2001.

[7] C. Huang *et al.*, "Boosting nested cascade detector for multi-view face detection," in *Proceedings of IAPR International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 415–418, Aug. 23 - 26, Cambridge, UK, 2004.

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

[9] Z. Yang *et al.*, "Multi-view face pose classification by tree-structured classifier," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 358–361, Genoa, Italy, Sept. 11 - 14, 2005.

[10] E. Islam, A. Khan, and I. Kim, "Effective face pose classification method," in *Proceedings of IEEE International Conference on Computer, Control and Communication (IC4)*, pp. 1–6, Karachi, Pakistan, Feb. 17 - 18, 2009.

[11] M. Hanif and U. Ali, "Optimized visual and thermal image fusion for efficient face recognition," in *Proceedings of IEEE International Conference on Information Fusion*, pp. 1–6, Florence, Italy, Jul. 10 - 13, 2006.

[12] I. Ulusoy and H. Yuruk, "New method for the fusion of complementary information from infrared and visual images for object detection," *IET Image Processing*, vol. 5, no. 1, pp. 36–48, 2011.

[13] X. Wang and G. Li, "Fusion algorithm for infrared-visual image sequences," in *Proceedings of IEEE International Conference on Image and Graphics (ICIG)*, pp. 244–248, Hefei, China, Aug. 12 - 15, 2011.

[14] V.E. Neagoe, A.D. Ropot, and A.C. Mugioiu, "Real time face recognition using decision fusion of neural classifiers in the visible and thermal infrared spectrum," in *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 301–306, London, UK, Sept. 5 - 7, 2007.

[15] A. Apatean *et al.*, "Visible-infrared fusion in the frame of an obstacle recognition system," in *Proceedings of IEEE International Conference on Automation Quality and Testing Robotics (AQTR)*, vol. 1, pp. 1–6, Cluj-Napoca, Romania, May 28 - 30, 2010.

[16] J. Zhu *et al.*, "Multi-class AdaBoost," *Statistics and its Interface*, vol. 2, pp. 349–360, 2009.

[17] E. Allwein, R. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research (JMLR)*, vol. 1, pp. 113–141, 2000.

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