NORTHWESTERN UNIVERSITY

Context-aware and Attentional Visual Object Tracking

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical and Computer Engineering

By

Ming Yang

EVANSTON, ILLINOIS

June 2008

ABSTRACT

Context-aware and Attentional Visual Object Tracking

Ming Yang

Visual object tracking, *i.e.* consistently inferring the motion of a desired target from image sequences, is a must-have component to bridge low-level image processing techniques and high-level video content analysis. This has been an active and fruitful research topic in the computer vision community for decades due to both its versatile applications in practice, *e.g.* in human-computer interaction, security surveillance, robotics, medical imaging and multimedia applications, and diverse impacts in theory, *e.g.* Bayesian inference on graphical models, particle filtering, kernel density estimation, and machine learning algorithms.

However, long-term robust tracking in unconstrained environments remains a very challenging task, and the difficulties in reality are far from being conquered. The two core challenges of the visual object tracking task are the computational efficiency constraint and the enormous unpredictable variations in targets due to lighting changes, deformations, partial occlusions, camouflage, quick motion and imperfect image qualities, *etc.* More critical, the tracking algorithms have to deal with these variations in an unsupervised manner. All the target variations in on-line applications are unpredictable, thus it is extremely hard, if not impossible, to design universal target specific or non-specific observation models in advance. Therefore, these challenges call for non-stationary

target observation models and agile motion estimation paradigms that are intelligent and adaptive to different scenarios.

In the thesis, we mainly focus on how to enhance the generality and reliability of object-level visual tracking, which strives to handle enormous variations and takes the computational efficiency constraint into consideration as well. We first present an in-depth analysis of the chicken-and-egg nature of on-line adaptation of target observation models directly using the previous tracking results. Then, we propose two novel ideas to combat unpredictable variations: context-aware tracking and attentional tracking. In context-aware tracking, the tracker automatically discovers some auxiliary objects that have short-term motion correlation with the target. These auxiliary objects are regarded as the spatial contexts to enhance the target observation model and verify the tracking results. The attentional tracking algorithms enhance the robustness of the observation models by selectively focusing on some discriminative regions inside the targets, or adaptively tuning the feature granularity and model elasticity. Context-aware tracking aims to search for external informative contexts of targets, in contrast, attentional tracking tries to identify internal discriminative characteristics of targets, thus they are complementary to each other in some sense. The proposed approaches can tolerate many typical difficult variations, thus greatly enhancing the robustness of the region-based object trackers. Besides single object tracking, we also introduce a new view to multiple target tracking from a game-theoretic perspective which bridges the joint motion estimation and the Nash Equilibrium of a particular game and has linear complexity with respect to the number of targets. Extensive experiments on challenging real-world test video sequences demonstrate excellent and promising results of the proposed object-level visual tracking algorithms.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Ying Wu for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Aggelos K. Katsaggelos, Prof. Thrasyvoulos N. Pappas, and Dr. James Crenshaw, for their encouragement, insightful comments, and hard questions.

My sincere thanks also goes to Dr. James Crenshaw, Dr. Senthil Periaswamy, and Dr. Qiong Liu, for offering me the summer internship opportunities in their groups and leading me working on diverse exciting projects.

I thank my fellow labmates in Northwestern Vision Group: Gang Hua, Ting Yu, Zhimin Fan, Shengyang Dai, Junsong Yuan, Jiang Xu, Jialue Fan, and Neelabh Gupta, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years. Also I thank my friends in Tsinghua University: Chen Fan, Wensheng Wang, Bo Qin, Xiwei Wang, Yonggang Zhao, Hua Chen, Junlin Li, and Rui Zhou. In particular, I am grateful to Dr. Chen Fan for enlightening me the first glance of research.

Last but not the least, I would like to thank my family: my parents Siyu Yang and Xiaolan Ding, for giving birth to me at the first place and supporting me spiritually throughout my life.

Table of Contents

CHAPTER 1

Introduction

Visual tracking in the computer vision community refers to the efforts of consistently inferring the motion of the desired targets, *e.g.* feature points, contours, regions of interest, and articulated objects *etc.*, from image sequences captured by single or multiple cameras, which is a fundamental component to bridge low-level image processing techniques and high-level video content analysis. In particular, the task is often referred to as visual object tracking to emphasize the cases where targets bear some semantic meaning. Visual tracking has gained extensive research interest since the early 1970s. In the new century, the rapid growth of computing power and the sharp drop of storage cost, especially as video cameras become pervasive, boost more research efforts on visual tracking. The popularity and significance of visual tracking in vision originates from its numerous applications in practice and diverse impacts in theory.

Inference of the motion parameters of some targets from video, *e.g.* the trajectories, scale and orientation, and joint pose configuration of the targets, is an indispensable component in many applications. To list a few typical applications in different areas: human-computer interaction, *e.g.* hand [**105**] and face tracking [**99**] for gaming, and eye gaze tracking [**120**] for disability assistance; security surveillance [**33**], *e.g.* airport surveillance, door access control, and home monitoring; medical image processing, *e.g.* tracking cardiac borders in MRI images [**75**] or in echocardiography [**119**]; multimedia applications, *e.g.* face and people tracking for video conferencing [**71**], lip tracking in audio-visual analysis [**53**]; activity and event analysis, *e.g.* gesture tracking [**108**],

facial expression tracking [39]; and robotics *e.g.* autonomous vehicle and intelligent traffic control [76, 88, 38]. Most of the aforementioned applications heavily rely on long-duration, efficient and robust tracking in an unconstrained environment, which is the ultimate goal of visual tracking research efforts.

Towards this end, numerous novel algorithms as well as a lot of classical algorithms were developed and applied to visual tracking, *e.g.* the Kalman filter [51], probabilistic data association filtering (PDAF) [6, 78], multiple hypothesis tracking (MHT) [79, 18], Bayesian inference on graphical models, particle filtering or sequential Mote Carlo [45, 20, 47, 106] (also known as CONDENSATION in vision literature [46]), subspace analysis [9, 36, 61, 110], kernel-based density estimation [15, 32, 22, 23, 21], variational analysis [39], and various machine learning algorithms, such as exemplar-based pattern learning [93], support vector machine (SVM) [4], relevance vector machine (RVM) [101] and on-line boosting [69, 5, 28]. In addition, tracking has greatly benefited from and interacted with many related research tasks in vision, *e.g.* local feature descriptor [84, 62, 68], object detection [96] and recognition [24], image segmentation [83], and background modeling [87].

Although recent years have witnessed remarkable advance in both theory and practice, visual tracking remains a challenging task and the diverse difficulties in reality are far from being conquered. In fact, it is the set of difficulties that a task faces that shape its identity and scope. So the foremost question is what are the core challenges that characterize the identity of visual tracking and distinguish it from other similar tasks? Most of the visual tracking algorithms are confronted by two slightly contradictory challenges: the demands for computational efficiency and the capability to handle the unpredictable variations of the targets. Computational efficiency is the inherent constraint for tracking since real-time processing is vital for the successes of most online applications and even for off-line video analysis applications due to the vast video data. Especially, when

the motion parameters are in high dimensional space, it is time consuming to explore the large solution space. Without this computational constraint, tracking is no longer a stand-alone problem from detection and recognition tasks. The other fundamental challenge is the dynamic nature of the targets due to enormous and unforeseeable variations in real-world scenarios. In unconstrained environments, there are too many factors that may affect the image evidence of the targets, *e.g.* background may be cluttered or even contain some camouflage objects as distractions. Illumination conditions may change evenly or unevenly so as to affect the target appearance, moreover, partial occlusion, out-of-plane rotation, target deformation, and quick motion all may present severe threats to long-term robust tracking. All these variations are unpredictable, and therefore it is extremely hard, if not impossible, for a tracker to consider all the potential variations and identify target specific or non-specific image invariants in advance. Adding further complexity, the visual tracking algorithms have to deal with these variations in an unsupervised and incremental manner. After initialization, the trackers will have no supervision to verify the tracking results and can hardly discern whether the appearance of the target is changing or partial occlusion is happening, so the estimation error could be accumulated. Besides, the trackers are expected to be insensitive to inaccurate target initialization and low image resolution or poor quality. In summary, the demand for computational efficiency and the dynamic nature of the tracking scenario are the two core challenges that tracking algorithms have to address, which are distinguishable from object detection and recognition tasks where the variations are expected to be covered by the training samples and the computation is not the topmost concern.

In the thesis, we mainly focus on how to enhance the generality and reliability of object-level visual tracking given no prior knowledge about the targets. On-line adaptation of target models to follow the dynamic changes is a natural and straightforward choice to handle the unpredictable

variations. But analyzed in an appearance-based subspace tracking framework, we find that directly updating target models with previous tracking results is a chicken-and-egg problem due to the unsupervised nature of tracking. Therefore, we investigate several novel approaches adapting region-based target models for object-level tracking to combat a target's dynamic appearance. We propose two novel ideas: context-aware tracking and attentional tracking. In context-aware tracking, the tracker automatically discovers some auxiliary objects that have short-term motion correlation with the target as the spatial contexts which can enhance the target observation model and provide additional verification. Inspired by psychological findings, the attentional tracking algorithms augment the robustness of the observation models by selectively attending some discriminative regions inside the targets, or adaptively tuning the feature granularity and model elasticity. Context-aware tracking aims to leverage external informative contexts of targets to verify the tracking results, in contrast, attentional tracking tries to identify internal discriminative characteristics of targets to enhance the robustness of tracking, thus they are complementary to each other in some sense. These approaches are robust to quite a few difficult cases, *e.g.* out-of-plane rotation, complex partial occlusions, and inaccurate initialization, so as to achieve promising results on challenging real-world test sequences. Besides single object tracking, we also introduce a new view to multiple target tracking from a game-theoretic perspective which bridges the joint motion estimation and the Nash Equilibrium of a particular game. The advantage of this multiple target tracking algorithm is that it is decentralized and has linear complexity with respect to the number of targets.

The thesis is organized as follows. Related work are reviewed in Chapter 2 as well as the Bayesian inference formulation of visual tracking. The chicken-and-egg nature of on-line adaptation is analyzed in Chapter 3 in a subspace tracking algorithm. Mining auxiliary objects for context-aware tracking and attentional tracking in terms of spatial selection, feature granularity

and model elasticity adaptation are present in Chapter 4 and Chapter 5, respectively. At the end, Chapter 6 introduces the game-theoretic multiple target tracking. Concluding remarks are given in Chapter 7.

CHAPTER 2

Related Work

Countless tracking algorithms have been proposed in past decades and can be reviewed from different perspectives and categorized with different criteria, for instance, in terms of the particular targets of interests, *e.g.* head [8], pedestrian [33], or vehicles [38], *etc.*; in terms of the modalities and cues used, *e.g.* appearance-based [31] or shape-based [45]; in terms of the target representations, *e.g.* subspace based [9] or density based [15]; in terms of the difficulties that are focused on, *e.g.* robust to scale changes [13], lighting changes [31], or occlusions [12]; or in terms of the theoretical basis, *e.g.* manifold [55, 56] or variational analysis [39], *etc.*. We refer readers to [113] for a fairly comprehensive literature survey. In this chapter, we will first introduce the popular technical tasks in the tracking area and then concentrate on the probabilistic inference framework and study the related tracking algorithms in terms of their motion estimation strategies and the likelihood model's design philosophy.

2.1. Bayesian Inference Framework for Tracking

Historically, visual tracking is regarded as a generalization of target tracking in radar applications where the estimation of the target's 2D motion trajectory in a cluttered environment is the primary goal. This goal has led to many illuminative and seminal works in the 1960s and 1970s, *e.g.* Kalman filters [51] where tracking was formulated as recursively estimating hidden states in discrete-time linear dynamical systems, probabilistic data association filtering (PDAF) [7] and multiple hypothesis tracking (MHT) [79] that estimates the data association probabilistically in

noisy environment. These algorithms were quite successful in radar target tracking and inspired many visual tracking algorithms. Images can provide much richer observations which lead to more versatile tracking tasks rather than point tracking in a radar signal. Thus, the goals of visual tracking evolve from tracking the translation motion of sparse feature points [**84**, **18**] and estimating the dense image point correspondences from the motion of the brightness patterns in optical flow [**37**, **63**, **10**, **81**] to inferring the affine motions of the contours and shapes [**45**, **119**, **39**] and the regions of interests [**9**, **8**, **15**]. Besides tracking a single target, tracking multiple independent targets simultaneously [**79**, **18**, **64**, **78**, **47**, **69**, **115**, **34**, **117**, **115**, **54**], which is complicated by the coalescence phenomenon that multiple trackers are trapped by single image evidence, is further developed to the more ambitious task of estimation of the joint motion configuration of articulated objects [**52**, **20**, **107**, **44**, **102**, **12**, **108**, **77**].

Despite the versatile formulations, the majority of tracking algorithms fall into the match-and-search framework where searching a set of hypotheses leads to the one bearing the highest similarity to the target model, and that one is selected as the tracking result. Usually, the target is abstracted to certain concise representations which may be based on different modalities *e.g.* appearance, color, and texture *etc.* Given the historical tracking results, some predictions for the current frame can be obtained with temporal correlation. Then a set of hypotheses are selected either by analyzing the bottom-up image evidence or choosing the top-down model parameters. The hypothesis yielding the best matching measurement to the target model is picked up as the tracking result, and so on and so forth.

In a more principled view, this procedure can be formulated in a probabilistic Bayesian inference framework, where the hidden states of the target and image observations are denoted by $\bar{\mathbf{X}}_t = \{\mathbf{x}_1, \cdots, \mathbf{x}_t\}$ and $\bar{\mathbf{Z}}_t = \{\mathbf{z}_1, \cdots, \mathbf{z}_t\}$, respectively, and the tracking result is determined by

the maximum *a posteriori* (MAP) estimation,

$$\bar{\mathbf{X}}_t^* = \underset{\bar{\mathbf{x}}_t}{\operatorname{argmax}}\ p(\bar{\mathbf{X}}_t|\bar{\mathbf{Z}}_\mathbf{t}). \tag{2.1}$$

Since real-time processing is preferred in tracking, most algorithms maximize the *a posteriori* $\mathbf{x}_t^* = \operatorname{argmax}_{\mathbf{x}_t}\ p(\mathbf{x}_t|\bar{\mathbf{Z}}_t)$ in a recursive manner rather than optimizing the whole sequence, though maximizing $p(\bar{\mathbf{X}}_t|\bar{\mathbf{Z}}_t)$ is feasible at the cost of some latency. Further, by assuming the Markov properties in the time axis, tracking is well formulated as an inference problem on a hidden Markov chain as represented by the graphical model in Fig. 2.1, where we have $p(\mathbf{z}_t|\mathbf{x}_t, \bar{\mathbf{Z}}_{t-1}) = p(\mathbf{z}_t|\mathbf{x}_t)$ and $p(\mathbf{x}_t|\bar{\mathbf{X}}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ due to the chain structure and Markov property. Then,

$$\begin{aligned} p(\mathbf{x}_t|\bar{\mathbf{Z}}_t) &= \frac{p(\mathbf{z}_t|\mathbf{x}_t, \bar{\mathbf{Z}}_{t-1})p(\mathbf{x}_t|\bar{\mathbf{Z}}_{t-1})}{p(\mathbf{z}_t|\bar{\mathbf{Z}}_{t-1})} \\ &= \frac{p(\mathbf{z}_t|\mathbf{x}_t)\int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\bar{\mathbf{Z}}_{t-1})d\mathbf{x}_{t-1}}{p(\mathbf{z}_t|\bar{\mathbf{Z}}_{t-1})}. \end{aligned} \tag{2.2}$$

Therefore, $p(\mathbf{x}_t|\bar{\mathbf{Z}}_t)$ can be derived from $p(\mathbf{x}_{t-1}|\bar{\mathbf{Z}}_{t-1})$ recursively with the help of $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{z}_t|\mathbf{x}_t)$ which are called as *dynamic model* and *observation or likelihood model*, respectively. $p(\mathbf{z}_t|\bar{\mathbf{Z}}_{t-1})$ is regarded as a normalization factor unrelated to $\mathbf{x}_t$.

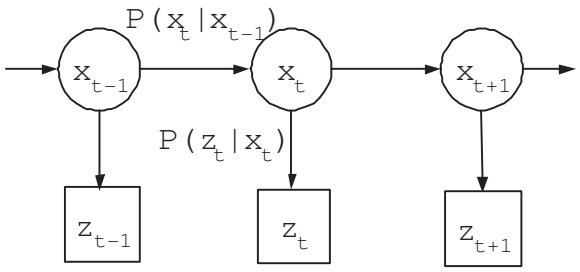


Figure 2.1. Markov chain representation of visual tracking.

Under this probabilistic formulation, the target and observation representations are encoded in $\mathbf{x}_t$ and $\mathbf{z}_t$, respectively, which are closely coupled with the observation model $p(\mathbf{z}_t|\mathbf{x}_t)$ that

determines the similarity measurement of a hypothesis against the target model. The dynamic model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ delineates the temporal correlation of the target states in successive frames. It is often assumed to be a constant velocity model or a simple smoothness model to indicate that the target states will not change dramatically in consecutive frames. The dynamic model can help to generate the hypotheses set and save some computations but may not have fundamental impact on the tracking performance.

Within this inference framework, the two essential and core issues in tracking are 1) the observation or likelihood model that encloses the target representation and similarity measurements, and 2) the motion estimation strategy that determines how to optimize or search $\mathbf{x}_t^*$ that maximizes $p(\mathbf{x}_t|\bar{\mathbf{Z}}_t)$. These two issues are not independent but highly correlated, which correspond to the two core challenges of tracking, *i.e.* enormous unpredictable variations and the computational efficiency constraint, as mentioned in Chapter. 1. Therefore, we can review the existing approaches from two threads by studying how they handle these two issues.

2.2. Motion Estimation and Observation Model

2.2.1. Motion Estimation strategy

The motion estimation strategy to search for the optimal or locally optimal hidden motion parameters $\mathbf{x}_t^*$ to maximize the *a posteriori* $p(\mathbf{x}_t|\bar{\mathbf{Z}}_t)$, together with the similarity measurement which is usually the basic computational unit, mainly determines the computational efficiency of a tracking algorithm. A straightforward and effective method is to perform local exhaustive search around the prediction given by the dynamic model based on the previous tracking results. This can be done by searching a predefined range of motion parameters [8], or finding the mode in a matching confidence map or occupance map [5, 28, 1, 114, 94] in coarse-to-fine or hierarchical ways [4]. This scheme implicitly assumes the target state's transition is smooth and only utilizes one mode

of $p(\mathbf{x}_{t-1}|\bar{\mathbf{Z}}_{t-1})$. However, evenly distributing the computation power in the discrete neighborhood around the prediction is not efficient and the *a posteriori* $p(\mathbf{x}_{t-1}|\bar{\mathbf{Z}}_{t-1})$ seldom has only one mode in reality. Therefore, the milestone CONDENSATION method [45, 46] proposed to represent the *a posteriori* density by weighted samples and propagate the conditional density by sequential Monte Carlo which is also well-known as particle filtering. In addition, a proposal density similar as the *a posteriori* density can well guide the importance sampling. The particle filtering method is capable of reserving multimodal densities and solving hard optimization problems by Monte Carlo simulation that has greatly boosted visual tracking research [20, 78, 41, 12, 118, 39]. However, when the target state $\mathbf{x}_t$ involves many motion parameters to be optimized, *e.g.* 6 affine motion parameters or joint motion configuration of articulated objects, the high dimensional solution space needs a large amount of samples to be covered, which induces the curse of dimensionality problem, and it is hard to know how many samples are sufficient to approximate a high dimensional density with multiple modes. Then, it would be efficient and ideal if the optimum or local optimum of the maximization of $p(\mathbf{x}_t|\bar{\mathbf{Z}}_t)$ could be solved analytically, by gradient descent search [31, 15, 16, 13, 22] or by Expectation-maximization (EM) estimation [103, 49, 90]. This requires the similarity measurements to be differentiable with respect to the motion parameters, but some parameters, *e.g.* scale and rotation angle, can hardly satisfy this requirement and need to resort to local exhaustive search or sampling method. Another new scheme to accelerate the motion estimation in searching the solution space is to borrow the indexing algorithms from database field, *e.g.* the KD-tree [42, 11] and locality sensitive search (LSH) [43, 3], to pre-hash the feature vectors for multiple queries/matching [29, 11, 112].

2.2.2. Target observation model design

Observation models are critical in tracking and are responsible for capturing the essential of targets and combating against variations. They define what the targets are that the trackers are chasing in feature space and contribute to the objective functions that the motion estimators need to optimize. Firstly, ideally some visual invariants are expected to be identified for the targets by extracting efficient features based on single modality, *e.g.* pixel intensities [37, 63], appearance template [31], skin color [103], edge-detection along curve normals [45], steerable filter responses, kernel-weighted feature density [15] *etc.*, a set of feature points or interest regions [5, 1, 116, 112, 114, 94], or by combining multiple cues [8, 104, 78, 106]. Although some features are robust to certain variations, such as gradient orientations and color histograms that are insensitive to illumination and in-plane rotation respectively, general invariant features against all possible variations are extremely hard to find, if not impossible.

As a further step, observation models try to cover the target's variations by involving exemplars [93] or off-line training [4, 100]. Off-line training can learn more complex visual invariants but it requires collecting sufficient training samples for a specific target and some variations such as partial occlusion can hardly be covered by limited training samples. These restrictions limit the application scenarios of pre-learned observation models.

An intuitive question to ask is, as the target appearances are inevitably dynamic, why do the trackers use stationary and fixed observation models? Therefore, recently more research efforts have examined to how to adapt the observation model to follow the target variations. There are mainly three ways to extend to non-stationary observation models, *i.e.*, 1) on-line adaptation of the observation model which means the parameters of observation models are adaptive, including on-line appearance models [49, 118], adaptive Gaussian color mixture [65], incremental subspace

update [36, 80, 61, 56], transductive co-inference [103, 104, 106]; 2) dynamic feature selection [13, 98]; 3) online learning [69, 5, 28] which means a classifier is trained with the training samples collected on-line. On-line adaptation is capable of handling unexpected target variations but tends to suffer from the drift problem. The robustness of the observation model also can be enhanced by incorporating multiple trackers and inferring the motion parameters in a collaborative way, as in multiple kernel tracking [32, 22, 23, 21], part-based tracking [116], and fragment tracking [1].

Another issue in observation model design is that most features used in observation models can only focus on certain characteristics of targets, for example, the existences of certain local visual patterns or coherence with certain overall feature statistics. Consequently, successful tracking methods for certain type of targets may not adapt to other targets easily. Therefore, for generally applicable trackers, matching need to be not only adaptive with respect to target variations but also flexible for distinctive targets. Specifically, for object-level tracking, two key aspects in designing observation models shall be flexible, *i.e.* the abstraction level of features, and the way to take into account the geometrical structures of targets.

In the thesis, we will mainly concentrate on the line of research on how to enhance the robustness of the observation model for dynamic targets while at the same time taking the computational efficiency constraint into consideration. By analyzing the nature of on-line adaptation in tracking, we propose two novel approaches, context-aware tracking and attentional tracking, by taking the spatial context information of targets into account, and selectively attending different discriminative characteristics of targets. These new approaches are quite different from the conventional observation model adaptation in that no new features are incorporated or training samples are collected for on-line learning but some correlated auxiliary objects are identified or a subset of more discriminative characteristics are selected from a rich target model during tracking.

CHAPTER 3

On-line Appearance Model Adaptation

Without any prior knowledge about the target, the appearance is usually the only cue available in visual tracking. However, in general the appearances are often non-stationary which may ruin the pre-defined visual measurements and often lead to tracking failure in practice. Thus, a natural solution is to adapt the observation model to the non-stationary appearances or equivalently to dynamically select the discriminant features. However, this idea is threatened by the risk of adaptation drift that originates in its ill-posed nature, unless good constraints are imposed. In this chapter, we present an in-depth analysis of on-line adaptation for appearance-based observation models and show that it is a chicken-egg problem in nature if directly using the latest previous tracking results to update the models. To alleviate the risk of drift in on-line appearance model adaptation, we propose to enforce three novel constraints for the optimal adaptation: (1) negative data, (2) bottom-up pair-wise data constraints, and (3) adaptation dynamics, which are different from most existing adaptation schemes. The general adaptation problem is substantialized as a subspace adaptation problem which can be solved in a closed-form. Further, to avoid solving eigenvalue decomposition for large matrices on-line, a practical iterative algorithm for subspace tracking is proposed and applied to test sequences in a variety of non-stationary scenes.

3.1. The Nature of On-line Adaptation

Visual appearance is critical for tracking, since the target is tracked or detected based on the matching between the observed visual evidence (or measurements) and the visual appearance

model. The visual appearances of an object may bear a manifold in the image space. Depending on the features used to describe the target and on the variances of the appearances, such a manifold can be quite nonlinear and complex. Therefore, the complexity in the appearances largely determines the degree of difficulty of the tracking task.

In the observation models with fixed appearance templates, the motion parameters to be estimated (denoted by $\mathbf{x}$) are the only variables that affect the appearance observations (denoted as $\mathbf{z}$). We denote the hypothesized image observations given $\mathbf{x}$ by $\hat{\mathbf{z}}(\mathbf{x})$. Then the observation model needs to measure the similarity of $\mathbf{z}$ and $\hat{\mathbf{z}}(\mathbf{x})$, or the likelihood $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\hat{\mathbf{z}}(\mathbf{x}))$. If $\mathbf{z}$ is a vector, *i.e.*, $\mathbf{z} \in \mathbb{R}^m$, this class of observation models is concerned with the distance between two vectors. Most tracking algorithms employ this type of observation models. There are cases where the motion parameters of interest are not the only contribution to the appearance changes, but there can be many other factors. We denote it by $\hat{\mathbf{z}}(\mathbf{x}, \theta)$. For example, the illumination also affects the appearance [31] (*e.g.*, in tracking a face), or the non-rigidity of the target changes the appearances (*e.g.*, in tracking a walking person), but we may not be interested in recovering too many delicate non-rigid motion parameters. Thus, there are uncertainties in the appearances model itself, and the observation model needs to integrate all these uncertainties, *i.e.*,

$$p(\mathbf{z}|\mathbf{x}) = \int_{\theta} p(\mathbf{z}|\mathbf{x}, \theta) p(\theta|\mathbf{x}) d\theta = \int_{\theta} p(\mathbf{z}|\hat{\mathbf{z}}(\mathbf{x}, \theta)) p(\theta|\mathbf{x}) d\theta.$$

In other words, given a motion hypothesis $\mathbf{x}$, its hypothesized observation $\hat{\mathbf{z}}(\mathbf{x})$ is no longer a vector, but a manifold in $\mathbb{R}^m$, and the observation model needs to calculate the distance of the evidence $\mathbf{z}$ to this manifold. Depending on the free parameters θ, such a manifold can be as simple as a linear subspace [9, 31], or as complex as a highly non-linear one [4, 93].

Although the appearance manifolds exist, in most cases, they are quite complex. Learning the manifold off-line is a good choice, but, in real applications, we may not have the luxury of being able to learn the manifolds of arbitrary objects for two reasons: we may not be able to collect enough training data, and the applications may not allow the off-line processing. Thus, we need to recover and update the appearance manifolds online [**49**, **95**, **104**, **61**] during the tracking. In general, we make a reasonable assumption that the manifold at a short time interval is linear [**36**, **80**]. The non-linear manifold is approximated by piece-wise linear subspace [**55**] or mapped to low dimensional manifold using non-linear mapping [**60**]. The learned general subspace could be updated to a specific one during the tracking [**56**]. The method of online feature selection, *e.g.*, in [**14**, **98**], can also be categorized to on-line adaptation, since the selected features span a subspace. In these methods, model drift is one of the common and fundamental challenges.

Although the appearance manifold of a target can be quite complex and nonlinear, it is reasonable to assume the linearity over a short time interval. In this chapter, we assume the appearances (or visual features) $\mathbf{z} \in \mathbb{R}^m$ lie in a linear subspace $\mathcal{L}$ spanned by r linearly independent columns of a linear transform $\mathbf{A} \in \mathbb{R}^{m \times r}$, *i.e.*, $\mathbf{z}$ is a linear combination of the columns of $\mathbf{A}$. We write $\mathbf{z} = \mathbf{A}\mathbf{y}$. The projection of $\mathbf{z}$ to the subspace $\mathbb{R}^r$ is given by the least square solution of $\mathbf{z} = \mathbf{A}\mathbf{y}$, *i.e.*,

$$\mathbf{y} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{z} = \mathbf{A}^\dagger\mathbf{z},$$

where $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the pseudo-inverse of $\mathbf{A}$. The reconstruction of the projection in $\mathbb{R}^m$ is given by:

$$\bar{\mathbf{z}} = \mathbf{A}\mathbf{A}^\dagger\mathbf{z} = \mathbf{P}\mathbf{z},$$

where $\mathbf{P} = \mathbf{A}\mathbf{A}^\dagger \in \mathbb{R}^{m \times m}$ is called the *projection matrix*. Unlike the orthonormal basis, the projection matrix is unique for a subspace. We can decompose the Hilbert space $\mathbb{R}^m$ into two

orthogonal subspaces: a r-dimensional subspace characterized by $\mathbf{P}$ and its $(m-r)$-dimensional orthogonal complement characterized by $\mathbf{P}^{\perp} = \mathbf{I} - \mathbf{P}$.

Therefore, the subspace $\mathcal{L}$ delineated by a random vector process $\{\mathbf{z}\}$ is given by the following optimization problem:

$$\mathbf{P}^* = \arg\min_P \mathtt{E}(||\mathbf{z} - \mathbf{P}\mathbf{z}||^2) = \arg\min_P \mathtt{E}(||\mathbf{P}^{\perp}\mathbf{z}||^2).$$

It is easy to prove that the optimal subspace is spanned by the r principal components of the data covariance matrix. This problem is well-posed since the samples from $\{\mathbf{z}\}$ are given, thus the covariance matrix is known.

However, in the tracking scenario, the problem becomes:

$$\{\mathbf{P}_t^*, \mathbf{x}_t^*\} = \arg\min_{P_t, x_t} \mathtt{E}(||\mathbf{P}_t^{\perp}\mathbf{z}(\mathbf{x}_t)||^2), \tag{3.1}$$

where $\mathbf{x}_t$ is the motion parameters to be tracked. In this setting, we are facing a dilemma: if $\{\mathbf{x}\}$ can not be determined, then neither can $\mathbf{P}$, and vice versa. Namely, given any tracking result, good or bad, we can always find an optimal subspace that can best explain this particular result. Thus, this is a chicken-and-egg problem, and this problem is even worse since no constraints on either $\mathbf{P}$ or $\{\mathbf{x}\}$ are imposed. Therefore, this problem is ill-posed and the formulation allows arbitrary subspace adaptations.

From the analysis above, it is clear that constraints need to be added to make this problem well-posed. A commonly used constraint is the "smoothness" of the adaptation, *i.e.*, the updated model should not deviate much from the previous one, and most existing methods [**14**, **36**, **49**, **80**]

solve this dilemma in the following manner:

$$\begin{cases} \mathbf{x}_t^* &= \arg\min_{x_t} \mathtt{E}(||\mathbf{P}_{t-1}^{\perp}\mathbf{z}(\mathbf{x}_t)||^2) \\ \mathbf{P}_t^* &= \arg\min_{P_t} \mathtt{E}(||\mathbf{P}_t^{\perp}\mathbf{z}(\mathbf{x}_t^*)||^2). \end{cases} \tag{3.2}$$

In this adaptation scheme, at time t, the data that are the closest to the subspace at the previous time instant are found first, and then are used to update the subspace. This approach is valid only if the following assumption holds: the optimal subspace at $t-1$ is also optimal for time t. In reality, this assumption may not necessarily be true, since a data point that is the closest to the subspace $\mathcal{L}_t$ may not be the closest to $\mathcal{L}_{t-1}$. Thus, we often observe that the model adaptation can not keep up with the real changes and the model gradually drifts away. When the data found based on $\mathbf{P}_{t-1}$ in fact deviate from $\mathbf{P}_t$ significantly, the adaptation is catastrophic. Although this approach makes the original ill-posed problem in Eq. 3.1 well-posed, it is prone to drift and thus not robust.

3.2. Appearance Adaptation with Bottom-up Constraints

From the analysis in Sec. 3.1, it is clear that we need more constraints than the adaptation dynamics constraint alone. In the tracking problem, at time t before the target is detected, all the observation data are unlabelled data, *i.e.*, we can not tell whether or not a certain observation should be associated (or classified) to the target appearance subspace. The adaptation dynamics constraint is a top-down constraint, which does not provide much supervised information to the data at time t. Therefore, to make the adaptation more robust, we need to also identify and employ bottom-up data-driven constraints, besides the smoothness constraint.

In this chapter, we propose to integrate the following three constraints:

- *Adaptation smoothness constraints.* The smoothness constraints are essential for the tracking process, since the process of the data at time t should take advantage of the

subspace at time $t-1$. There are many ways to represent and use this type of constraints. The most common scheme as indicated in Sec. 3.1 enforces a very strong smoothness constraint. In our approach, we treat the constraint as a penalty which can be balanced with other types of constraints. The penalty is proportional to the distance of two subspaces, *i.e.*, the Frobenius norm of the difference of the two projection matrices $||\mathbf{P}_t - \mathbf{P}_{t-1}||_F^2$;

- *Negative data constraints.* At the current time t, although it is difficult to obtain the positive data (*i.e.*, the visual observations that are truly produced by the target), negative data are everywhere. In fact, positive data are very rare in all the set of possible observation data. The negative data may help to constrain the target appearance subspaces. We denote the positive data at time t by $\mathbf{z}_t^+$, and the negative data by $\mathbf{z}_t^-$;
- *Pair-wise data constraints.* Given a pair of data points, it is relatively easier to determine whether or not they belong to the same class. Such pair-wise data constraints are also widely available. A large number of pair-wise constraints may lead to a rough clustering of the data. Based on the smoothness constraints, we can determine a set of *possible positive* data to constrain the subspace updating. The detail is in Sec. 3.2.4.

3.2.1. Formulation

When processing the current frame t, the following are assumed to be known: (1) the projection matrix of the previous appearance subspace $\mathbf{P}_{t-1}$, (2) a set of negative data collected from the current image frame, $\{\mathbf{z}_t^-\}$, (3) a set of possible positive data identified based on the pair-wise constraints, $\{\mathbf{z}_t^+\}$, (4) previous negative covariance matrix $\mathbf{C}_{t-1}^-$ and positive covariance matrix $\mathbf{C}_{t-1}^+$.

An optimal subspace should have the following properties. The negative data should be far from their projections onto this subspace; the positive data should be close to their projections, and

this subspace should be close to the previous one. Therefore, we form an optimization problem to solve for the optimal subspace at current time t:

$$\underset{A_t}{\operatorname{argmin}} J_0(\mathbf{A}_t) = \underset{A_t}{\operatorname{argmin}} \{\mathtt{E}(||\mathbf{z}_t^+ - \mathbf{P}_t\mathbf{z}_t^+||^2) + \mathtt{E}(||\mathbf{P}_t\mathbf{z}_t^-||^2) + \alpha||\mathbf{P}_t - \mathbf{P}_{t-1}||_F^2\}, \tag{3.3}$$

where $\mathbf{P}_t = \mathbf{A}_t\mathbf{A}_t^\dagger$ is the projection matrix and $\alpha > 0$ is a weighting factor. We denote by $\mathbf{C}_t^+ = \mathtt{E}(\mathbf{z}_t^+\mathbf{z}_t^{+T})$, and $\mathbf{C}_t^- = \mathtt{E}(\mathbf{z}_t^-\mathbf{z}_t^{-T})$. It is easy to show Eq. 3.3 is equivalent to the following:

$$\underset{A_t}{\operatorname{argmin}} J_1(\mathbf{A}_t) = \underset{A_t}{\operatorname{argmin}} \{\mathtt{tr}(\mathbf{P}_t\mathbf{C}_t^-) - \mathtt{tr}(\mathbf{P}_t\mathbf{C}_t^+) + \alpha||\mathbf{P}_t - \mathbf{P}_{t-1}||_F^2\}, \tag{3.4}$$

where $\mathtt{tr}(\cdot)$ denotes the trace of a matrix.

3.2.2. An closed-form solution

Theorem 1. *The solution to the problem in Eq. 3.4 is given by* $\mathbf{P}_t = \mathbf{U}\mathbf{U}^T$, *where* $\mathbf{U}$ *is constituted by the* r *eigenvectors that corresponds to the* r *smallest eigenvalues of a symmetric matrix*

$$\hat{\mathbf{C}} = \mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1}.$$

The proof of this theorem is given in the Appendix A. Please note that the solution to $\mathbf{A}_t$ is not unique, but the projection matrix $\mathbf{P}_t$ is. If we require that $\mathbf{A}_t$ is spanned by r orthogonal vectors, then $\mathbf{A}_t = \mathbf{U}$. Please also note the eigenvalues of $\hat{\mathbf{C}}$ may be negative.

By considering the data in previous time instants, we can use a forgetting factor $\beta < 1$, which can down-weight the influence of the data from previous times. This is equivalent to the use of exponentially-weighted sliding window over time. Thus, we can write:

$$\mathbf{C}_t = \sum_{k=1}^{t} \beta^{t-k}\mathtt{E}(\mathbf{z}_k\mathbf{z}_k^T) = \beta\mathbf{C}_{t-1} + \mathtt{E}(\mathbf{z}_t\mathbf{z}_t^T).$$

This way, we can update both $\mathbf{C}_t^+$ and $\mathbf{C}_t^-$.

3.2.3. An iterative algorithm

Sec. 3.2.2 gives a closed-form solution to the subspace, but this solution involves the eigenvalue decomposition of a $m \times m$ matrix $\hat{\mathbf{C}}$, where m is the dimension of the visual observation vectors and thus can be quite large. To achieve a less demanding computation, we develop an iterative algorithm in this section, by formulating another optimization problem as:

$$\underset{U}{\operatorname{argmin}}\, J_2(\mathbf{U}) = \underset{U}{\operatorname{argmin}} \{\mathtt{E}(||\mathbf{z}_t^+ - \mathbf{U}\mathbf{U}^T\mathbf{z}_t^+||^2) + \mathtt{E}(||\mathbf{U}\mathbf{U}^T\mathbf{z}_t^-||^2) + \alpha||\mathbf{U}\mathbf{U}^T - \mathbf{P}_{t-1}||_F^2\}\ s.t.\ \mathbf{U}^T\mathbf{U} = \mathbf{I}, \tag{3.5}$$

where $\mathbf{U} \in \mathbb{R}^{m \times r}$ is constituted by r orthonormal columns. The gradient of J_2 is given by:

$$\nabla J_2(\mathbf{U}) = \frac{\partial J_2(\mathbf{U})}{\partial \mathbf{U}} \propto (\mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1})\mathbf{U}. \tag{3.6}$$

To find the optimal solution of $\mathbf{U}$, we can use the gradient descent iterations:

$$\mathbf{U}^k \longleftarrow \mathbf{U}^{k-1} - \mu\nabla J_2(\mathbf{U}^{k-1}), \tag{3.7}$$

during which the columns of $\mathbf{U}^k$ need to be orthogonalized after each update.

To speed up the iteration, we can also perform an approximation. When the subspace is to be updated by the positive data $\mathbf{z}_t^+$, the PAST algorithm [109] can be adopted for fast updating. When the updating is directed by the negative data $\mathbf{z}_t^-$, we can use the gradient-based method in Eq. 3.6.

3.2.4. Pair-wise constraints

Although the target can not be detected directly, the low level image features which distinguish the target object from its neighborhood may give some hints about the target. Here we employ a graph

cut algorithm [82] to roughly cluster some sample appearances collected within the predicted target regions. Then we may be able to find possible positive data and negative data from bottom-up.

Suppose the predicted region for the target is a rectangular region centered at (u, v) with width w and height h. We draw uniform samples (*i.e.*, 15×15 image patches) to cover a rectangle region $(u \pm w, v \pm h)$. For each sample patch, the kernel-weighted [15] hue histogram $\mathbf{h}$ with 64 bins is calculated. The affinity matrix, obtained based on the similarity of all pairs of these histograms, is:

$$\mathbf{S} = [S_{ij}], \quad \text{where } S_{ij} = \exp\left\{\frac{(\rho(\mathbf{h}_i, \mathbf{h}_j) - \mu)^2}{2\sigma^2}\right\}, \tag{3.8}$$

where $\rho(\cdot)$ is the Bhattacharya coefficient, μ is the mean of all coefficients, σ is their standard deviation. These sample patches can be grouped into $3-5$ clusters by the eigenvalue decomposition of the affinity matrix.

It is not necessary to have a perfect clustering, as observed in our experiments. The image region delineated by the cluster with the minimum mean L^2 distance to the previous target subspace indicates the possible locations that the target may present. In practice, we can simply treat its geometric centroid as the possible location of the target and the corresponding appearance vector as the possible positive data $\mathbf{z}_t^+$.

3.2.5. Selecting negative data

The negative data should be selected carefully. Because if the negative data are too far from the target, the data point may already lie in the orthogonal complement of the target subspace, then minimizing the projections of the negative data may not help. In addition, if the negative data are too close to the target, they may lie partly in the target subspace such that the estimated target subspace is pushed away from its true place. Our selection of negative data is heuristic based

on the clustering in Sec. 3.2.4: in the image regions spanned by all the negative clusters, we find the locations whose appearances (or features) are close to the previous target manifold, and treat these appearance data as negative data $\mathbf{z}_t^-$ in order to distinguish the target from the negative clusters. This heuristic works well in our experiments, but a more principled selection deserves more investigation.

3.3. Experiments and Discussions

3.3.1. Setup and comparison baseline

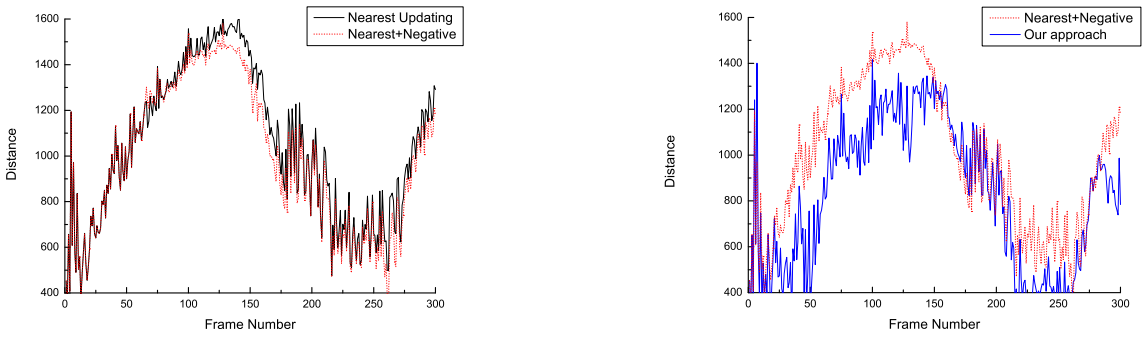
In our experiments, we aim to recover the motion parameter $\mathbf{x} = \{u, v, s\}$, where (u, v) is the location of the target and s is its scale. The corresponding appearance region is normalized to a 20×20 window and rasterized to a feature vector $\mathbf{z} \in \mathbb{R}^{400}$. Since the target appearances during tracking may become totally different from the first frame, the remedy of always including the initial appearance in the model [**14**, **36**] does not apply.

For comparison, we implemented a subspace updating tracker similar to the method in [**36**], where the nearest appearance observation $\mathbf{z}_i$ to the previous target subspace $\mathbf{P}_{t-1}$ is used to update the orthonormal basis of the subspace by using Gram-Schmidt and dropping the oldest basis. We refer to this method as *Nearest Updating*. The method is referred to as *Nearest+Negative* when the positive data are collected by the nearest scheme and the negative data are used in updating the same way as in our approach. In all these methods, the adaptation applies every 4 frames.

3.3.2. Impact of the positive and negative Data

In this quantitative study, we show that the use of negative and possible positive data do help. We have manually annotated a video with 300 frames, in which a head presents a 180^o out-of-plane rotation, and collect the ground truth appearance data for each frame (denoted by $\mathbf{z}_t^*$). The

comparison is based on the L^2 distance of the ground truth data $\mathbf{z}_t^*$ to the subspaces estimated by various methods. A smaller distance implies a better method.



(a) *Nearest Updating* vs. *Nearest+Negative Updating*

(b) *Nearest+Negative Updating* vs. Our approach

Figure 3.1. Comparison of the distances of the ground truth data to the updated subspaces given by three schemes.

As shown in Fig. 3.1(a), the distance curve for the *Nearest+Negative* scheme is slightly lower than that for *Nearest Updating*, showing negative data can help to keep the adaptation away from the wrong subspaces. We also observed in our experiments that the negative data themselves may not be able to precisely drive the adaptation to the right places. We compare the proposed method with *Nearest+Negative* in Fig. 3.1(b), in which the curve of our approach is apparently lower than that of *Nearest Updating*. This verifies that the possible positive data from bottom-up do help.

These two comparisons validate that the proposed approach are more capable of following the changes of the non-stationary appearances. Some sample frames are shown in Fig. 3.2, where the top row is the results of the proposed method, the middle row shows the location of the possible positive cluster and possible positive data is shown at the top-left corner of each frame, and the bottom row shows the results of *Nearest Updating* and the nearest data is shown at the top-left corner as well.

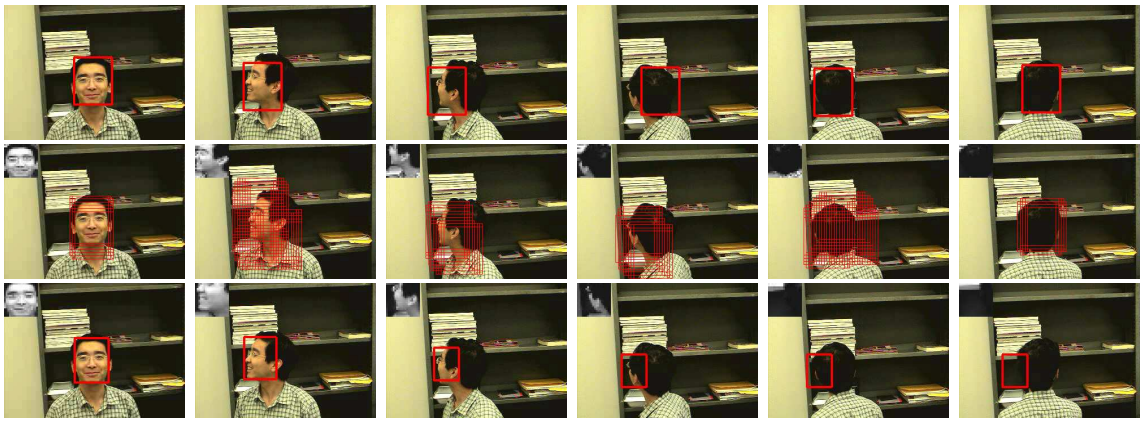


Figure 3.2. Tracking a head with 180^o rotation [`head180.avi`]. (top) our method, (middle) clustering, (bottom) *Nearest Updating*.

3.3.3. Impact of the clustering procedure

In this experiment, we compare our method with *Nearest Updating* in the situation of partial occlusion. We need to track a face, but the partial occlusion makes it difficult when the person drinks and the face moves behind a computer.

Figure 3.3. Clustering performance in `face.avi`: top row shows the drift process of *Nearest Updating* around frame 272; middle row lists 6 positive data at frame 272; bottom row lists 6 positive data at frame 284.

When the face moves slowly behind the computer, *Nearest Updating* drifts and erroneously adapts to a more stable appearance, *i.e.*, a back portion of the computer. In Fig. 3.3, the top row illustrates this drift process in detail. The middle row in Fig. 3.3 presents 6 appearance samples from the possible positive cluster in our method at the 272-th frame. Obviously, some of them are not faces, since the clustering is quite rough. But our heuristic of selecting the centroid of the

cluster does help and leads to a correct adaptation. Similarly, the bottom row shows the situation of our method at the 284-th frame. As the person moves upward, our method correctly follows the face.

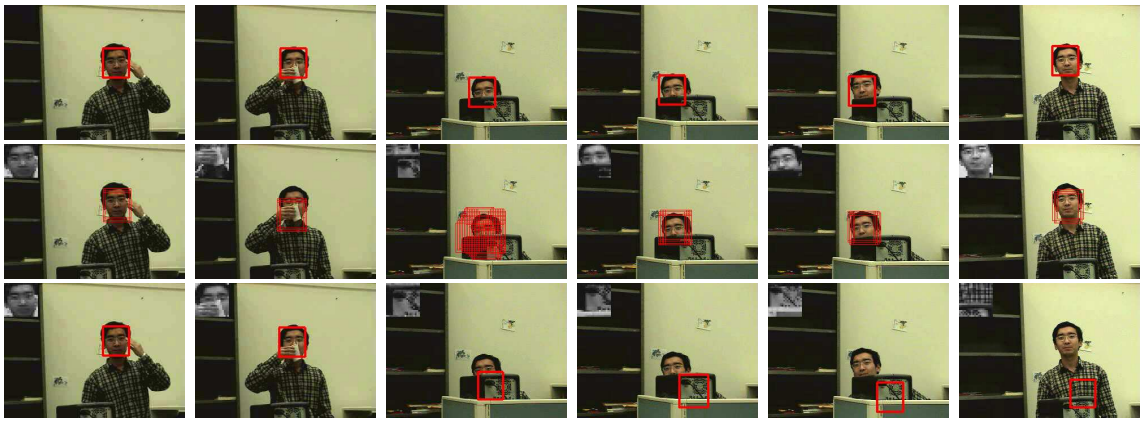


Figure 3.4. Tracking partial occlusion target [`face.avi`]. (top) our method, (middle) clustering, (bottom) *Nearest Updating*.

This also illustrates that a rough clustering is sufficient for our method which is more robust than *Nearest Updating*. Some sample frames are shown in Fig. 3.4, where the top row is our method and the bottom row is that of *Nearest Updating*.

3.3.4. More test sequences with rotations and illumination changes

Fig. 3.5 shows the results of tracking a head presenting 360^o out-of-plane rotation. The appearances of different views of the head are significantly different, which makes the tracking difficult and also challenges the adaptation. Our experiment shows that *Nearest Updating* tends to stick to the past appearances and thus reducing the likelihood of including new appearances. For example, when the front face gradually disappears, this scheme is unable to adapt to the hairs to track the back head. In all of our experiments, this scheme loses track when the face fades away. In contrast, since the bottom-up information (*i.e.*, the negative and possible positive data) hints the emerging

appearances, our method can successfully track the head, although the bottom-up processing is quite rough.

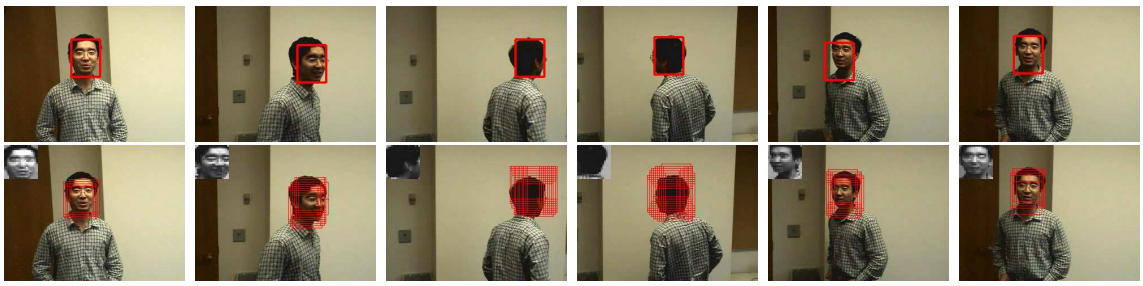


Figure 3.5. Tracking a head with 360^o out-of-plane rotation [`head360.avi`]. (top) our method, (bottom) clustering.

In general, 2D in-plane rotation also induces significant changes to the target appearance. In Fig. 3.6, the black background is similar to the panel of the watch such that the adaptation in *Nearest Updating* deviates from the true subspace and it drifts rapidly. On the contrary, although the proposed method is also distracted at frame 444, it is able to recover quickly thanks to the help from the pair-wise constraints.

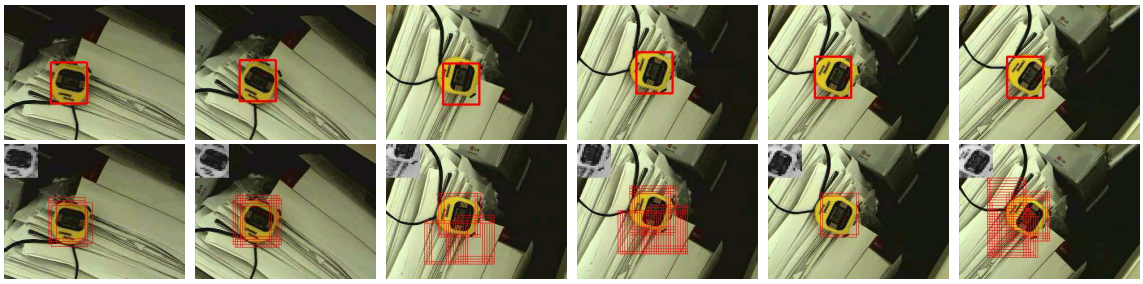


Figure 3.6. Tracking a watch with in-plane rotations [`watch.avi`]. (top) our method, (bottom) clustering.

In Fig. 3.7, we demonstrate the performance of our algorithm for large illumination changes. *Nearest Updating* will soon lose the face after the sudden lighting change, since all observations are far from the target subspace thus the samples used in *Nearest Updating* to update the subspace

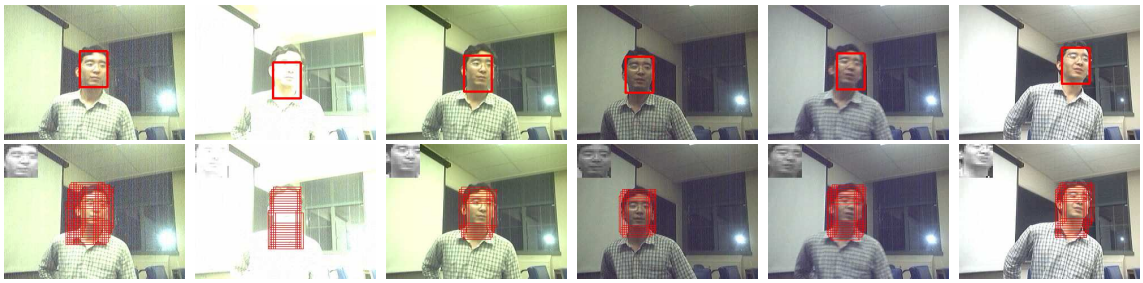


Figure 3.7. Tracking a face with large illumination changes. [`light.avi`]. (top) our method, (bottom) clustering.

are kind of random. While, in our method, selecting the centroid of the positive clustering to update the model ensures the the samples used are consistent.

Fig. 3.8 shows the results of tracking the head of a person walking in a real environment. The appearance of the head undergoes large changes, and there are also scale changes. Our result shows that *Nearest Updating* drifts to the background when the appearances of the black hair that the subspace has initially learned almost disappears. This happens when the person moves towards the camera. On the other hand, the proposed method can work comfortably and stably in the case.

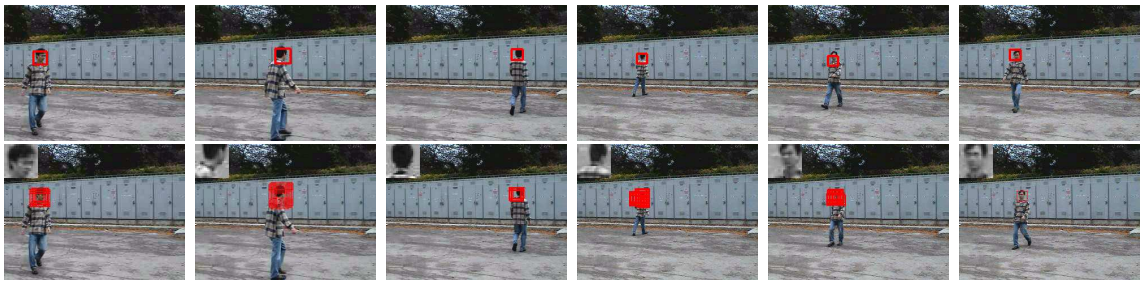


Figure 3.8. Tracking a head [`walking.avi`]. (top) our method, (bottom) clustering.

3.3.5. Discussions

All the above experiments have validated the proposed approach. When the target model experiences dramatic changes, we can explain the reason why the methods sharing the same nature as *Nearest Updating* deteriorate in two aspects. First, these methods tend to adhere to the old model

as much as possible and are reluctant to include the changes. When the model changes completely or the original features disappear, the updated model will drift away from the true one eventually. Second, when the drift starts, there is no mechanism in these methods to force them back, thus the drift is unstable and catastrophic. In contrast, since our method utilizes the information from bottom-up, it can be thought as feedbacks that makes our method stable and avoids catastrophic drift to a large extent. As a result, the proposed method can be more robust and stable to cope with the adaptation drift.

In this chapter, we have investigated the on-line adaptation of appearance-based observation models. If no constraints are imposed, this problem is ill-posed. Instead of the commonly used nearest updating scheme, we propose to impose both top-down smoothness constraints and the bottom-up data-driven constraints from current observances. Our method balances three factors: (1) distance of positive data to the subspace, (2) the projections of the negative data, and (3) the smoothness of two consecutive subspaces. The proposed method can largely alleviate the risk of adaptation drift and thus achieving better tracking performance.

CHAPTER 4

Context-aware Visual Tracking

Many tracking methods face a fundamental dilemma in practice: tracking has to be computationally efficient but verifying whether or not the tracker is following the true target tends to be demanding, especially when the background is cluttered and/or when occlusion occurs. This dilemma originates from the opposite requirements for the image likelihood models: on one hand, the likelihood model should be simple for efficient motion estimation and tracking; on the other hand, it has to be sophisticated for comprehensive verification of the target. Due to the lack of a good solution to this problem, many existing methods tend to be either computationally intensive by using sophisticated image observation models, or efficient but vulnerable to false alarms. This greatly threatens long-duration robust tracking. As an alternative to the on-line adaptation idea, this chapter presents a novel solution to this dilemma by considering the context of the tracking scene. Specifically, we integrate into the tracking process a set of auxiliary objects that are automatically discovered in the video on the fly by data mining. Auxiliary objects have three properties at least in a short time interval: (1) persistent co-occurrence with the target; (2) consistent motion correlation with the target; and (3) easy to track. Regarding these auxiliary objects as the contexts of the target, the collaborative tracking of these auxiliary objects leads to an efficient computation as well as a strong verification. Our extensive experiments have exhibited exciting performance in challenging real-world testing cases.

In all the existing methods, the dynamic environment is taken for granted as the adverse party for the tracker, as it generates false positives, and most computation has to be spent in separating

the true target from the environment. However, the environment can also be advantageous to the tracker if it contains objects that are correlated to the target. For example, if we need to track a face in a crowd, it is almost impossible to learn a discriminative model to distinguish the face of interest from the rest of the crowd. Why do we have to focus our attention only on the target? If the person (with that face) is wearing a quite unique shirt (or a hat), then including the shirt (or the hat) in matching will surely make the tracking much easier and more robust. By the same token, if another face is always accompanying the target face, treating them as a geometric structure and tracking them as a group will be much easier than tracking either of them. We call this new approach *context-aware tracking* (CAT) as it takes into consideration the context of the target, as shown in Fig. 4.1.

A target is seldom isolated and independent to the entire scene, therefore there may exist some objects that have short-term or long-term motion correlations to the targets (but are unknown to the tracker beforehand). Thus, taking the advantage of this context information in an efficient way can improve the robustness of the tracker as the spatial context provides additional verification. We represent the context of a target by a set of *auxiliary objects* that are automatically discovered

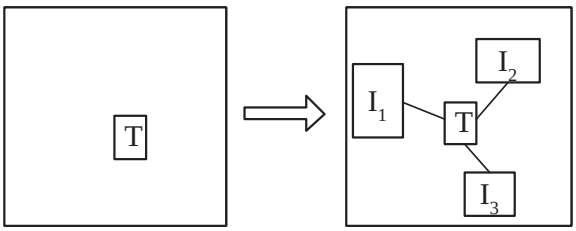


Figure 4.1. Illustration of context-aware tracking. T indicates the target and I_k means the spatial context of the target. Traditional tracking methods focus their attention on the target only, while context-aware tracking considers the target and its spatial context within a network.

on the fly in an unsupervised fashion by using data mining techniques. A context-aware tracker is able to discover a set of auxiliary objects and track them simultaneously.

Auxiliary objects can be in various forms, *e.g.* solid semantic objects which bear intrinsic relations to the target, or certain image regions that happen to have motion correlation with the target for a short period of time. They may reliably associate to the target for a long duration, or only for a short time interval, or may not exist at all. Thus, it is impossible to determine auxiliary objects off-line in advance. They have to be discovered on the fly. For example, in Fig. 4.2, the targets of interest are the heads in solid-yellow boxes, and the image regions in dash-red boxes are the auxiliary objects discovered automatically. We resort to data mining techniques for discovering auxiliary objects by learning their co-occurrence associations and estimating affine motion models to the target. Data mining methods originated from text information processing and relational databases [2], and have found their uses in extracting video objects [**85**, **86**, **57**]. To the best of our knowledge, the proposed approach presents an original attempt of combining visual tracking and data mining in a collaborative tracking framework.

Figure 4.2. Some sample auxiliary objects to the target head.

This new approach has the following advantages. Firstly, it is computationally efficient. Because the auxiliary objects by definition are those easy to track (*e.g.* color regions), tracking them does not incur significant computational costs. Secondly, it outputs more accurate tracking results.

The new method tracks the target and the set of auxiliary objects as a random field in a collaborative manner. It is provably correct that the uncertainty of the motion estimation is reduced. Thirdly, it also provides an effective verification, because the learned motion and/or geometric correlations among the target and the auxiliary objects serve as a strong cue for verification. Last but not the least, it is intelligent and robust. All the auxiliary objects and the motion correlation (*i.e.*, the random field) are automatically discovered on the fly.

In contrast to the previous on-line adaptation methods, rather than changing the target observation, we propose to enhance the observation model by on-line discovery of some auxiliary objects [**111**] to help verify the target tracking results in a collaborative way. The new approach, called *context-aware visual tracking*, or CVT, addresses the following three important issues (the entire procedure of CVT algorithm is summarized in Fig. 4.3):

- **Mining auxiliary objects** (in Sec. 4.1): the methods of extracting the candidates of auxiliary objects and mining the associations will be discussed. For auxiliary object candidates, multibody grouping is employed to discover the potential multibody structure from motion and to estimate the affine motion models through subspace analysis. This step not only identifies a set of auxiliary objects, but also learns a random field among them;
- **Collaborative tracking** (in Sec. 4.2): both the target and the set of auxiliary objects need to be tracked in CAT. Because they are not independent, the tracking is formulated based on a random field and is achieved efficiently by the collaborations among all the individual trackers in the network where an individual tracker influences other trackers as well as receiving influence from others;
- **Robust fusion** (in Sec: 4.3): for an individual tracker, there may exist inconsistency among the influences it receives and its own image measurements. Handling inconsistency is fundamental and critical to fuse auxiliary object trackers and the target tracker.

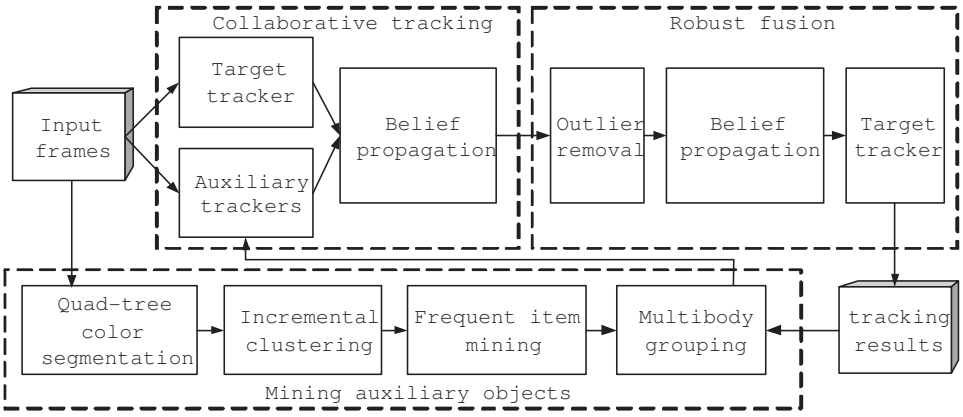


Figure 4.3. Block diagram of CAT algorithm. The sub-modules of auxiliary object mining, collaborative tracking, and robust fusion are enclosed in dash rectangles.

4.1. Mining Auxiliary Objects

4.1.1. Auxiliary objects

Auxiliary objects (AOs) are the spatial context that can help the target tracker. We abuse a little bit the term "object". In fact, it is not necessary for an AO to be a semantic object. In the tracking scenario, it refers to an informative image region or an image feature that satisfies the following three properties:

(1) frequent co-occurrence with the target;

(2) consistent motion correlation to the target;

(3) suitable for tracking.

Although this definition may cover a large variety of image regions or features, not all of them are appropriate for balancing the complexity and generality. Since the prior knowledge about the target and the environments are in general not accessible, it is preferable to choose simple, generic and low-level auxiliary objects, such as image regions or feature points. Feature points are geometrically significant and provide the most localized information. There are some outstanding

work on invariant feature points, *e.g.* [84, 62, 66, 25]. Although feature points may be salient and therefore suitable for object recognition, they are in general prone to occlusion, lighting and local geometry changes. Thus they are not always stable and reliable in video. In addition, extracting invariant features needs a good amount of computation, which makes it hard to achieve real-time performance. Therefore, although the tracking of feature points can be quite efficient, we generally do not use feature points as auxiliary objects.

Instead, we choose to use significant image regions. Different from localized image feature points, image regions reflect the visual property of a neighborhood, and they tolerate more occlusions and local geometry changes. More importantly, image regions, if selected properly, can be reliably and efficiently tracked, for example, by the mean-shift algorithm [16]. Although texture regions may have invariants and can be very significant, our current implementation does not use them because it takes more computation to spot them than color regions. Therefore, our current treatment for data mining is to discover a set of color regions that are temporally stable and spatially correlated to the target in a video sequence in an unsupervised way.

4.1.2. Item candidate generation

To follow data mining conventions make our discussion clear. We define the following terms for our video data mining task.

Definition 1. *We denote an* item candidate *by s which is a particular image feature obtained by low-level image processing; an* item *by I which is a quantized item candidate in a* vocabulary *$\mathcal{V} = \{I_1, \ldots, I_N\}$ which is learned by clustering all item candidates; an* itemset *by $\mathbf{I} \subset \mathcal{V}$, set of items; and a transaction by τ, the itemset within a neighborhood R.*

In our implementation, an item candidate is a rough color segment with its motion parameters, and an item is defined by $I = \{H(I), \mathbf{x}_I\}$, where $H(I)$ is the average color histogram of the item and $\mathbf{x}_I$ is the motion parameters and respective covariances. The set of candidate AOs, denoted by F, is a subset of $\mathcal{V}$, which are frequently co-occurrent with the target. The candidate AOs that have strong motion correlations to the target are identified as auxiliary objects.

The item candidates s, *i.e.*, the color segments in our case, are the inputs for mining. In the tracking scenario, efficient segmentation is more preferred than a delicate but expensive one since exact boundaries of the segments are not necessary for mining and tracking. In our current implementation, we employ the classical split-merge quad-tree color segmentation [48]. The image is recursively split into the smallest possible homogenous color regions, and then the adjacent regions with similar appearances are merged gradually. The most prominent advantage of this method is computational efficiency. Some segments are not appropriate for tracking, so we employ some heuristics to prune them, *e.g.* segments that are too large (the area over $1/2$ of the entire image) or too small (the area less than 64 pixels), and concave segments (the area less than $1/2$ of the bounding box) are excluded. These kinds of item candidates are suitable for tracking. Fig. 4.4 shows some typical segmentation results.

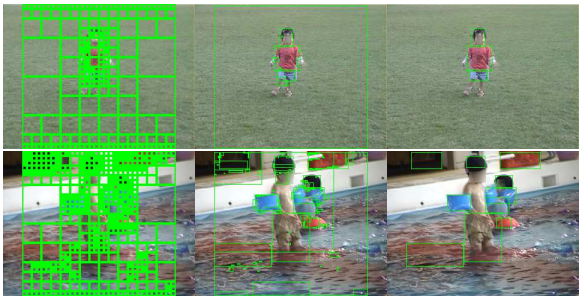


Figure 4.4. Illustration of the quad-tree color segmentation. (left) input frame, (middle) over-segmentation, (right) pruned segmentation.

4.1.3. Frequent item mining

Candidate auxiliary objects are the items that are frequently co-occurrent with the target. To build the vocabulary $\mathcal{V}$ so as to construct the transactions for mining, we need to quantize the item candidates. In conventional mining applications, usually item candidates can be collected and quantized off-line by k-means or kNN clustering methods. But in this tracking scenario, we have to do this in an incremental way. The procedure is the following. The color segments in each incoming frame are matched to the items in the current vocabulary by the Bhattacharyya coefficient [16] of the histograms of the segments as the similarity measurement. Then, each color segment (*i.e.* item candidate) can be quantized and given a label, *e*.g. I_A to I_G are items as shown in Fig. 4.5. Afterwards, for each item, we form a transaction that consists of the item itself and the items within its neighborhood. There are different choices of the neighborhood. For example, we can use the item itself (*i.e.* use a 0 neighbor). The items inside the region of interest in each frame construct a transaction τ, and a transaction database is built based on M consecutive frames.

Given the transaction database, the items which have a high co-occurrent frequency will be chosen as candidate auxiliary objects. Since the mining is performed online, we need to take into account the importance of the historical images. We maintain an M-frame sliding window ($M = 100$ in the experiments) and count the item frequency $f(I_n) = \sum_{i=t-M+1}^{t} \beta^{t-i} B_i(I_n)$ with the forgetting factor $\beta = 0.9$ where $B_i(I_n)$ is a binary function and 1 indicates I_n appears in frame i. If image segmentation does not end up with too many small segments, the frequent items are good enough for identifying candidate auxiliary objects. If the segmentation tends to over-segment and produces too many small segments, we cannot use the 0 neighbor for constructing transactions, but use the nearby items to form transactions to identify co-occurrent patterns that merge the adjacent small segments. This is another reason that it is fine for the image segmentation

step to be imperfect. As illustrated in Fig. 4.5, though there are quite many color segments in each frame, by counting their co-occurrent frequencies, only $F = \{I_A, I_B\}$ are identified as frequent items, *i.e.* candidates of auxiliary objects. The rest of the problem is to determine whether a candidate really bears a motion correlation to the target. The issue will be discussed next.

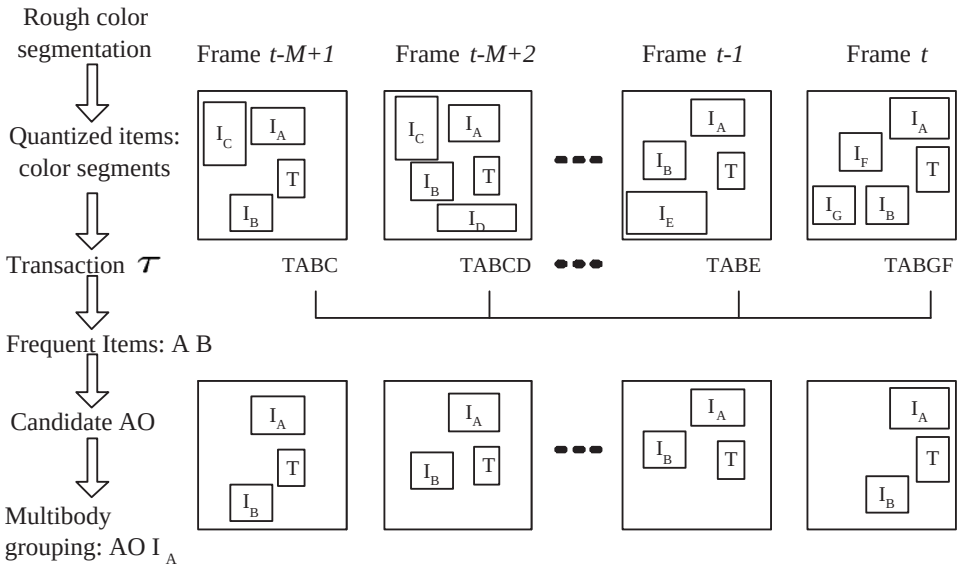


Figure 4.5. Illustration of mining auxiliary objects. The target is denoted as T and I_A to I_G represent the items (*i.e.* the color segments). I_A and I_B are selected as candidate auxiliary objects as they are frequently co-occurrent with the target. I_A is identified as one auxiliary object by multibody grouping since it has strong motion correlation to T.

4.1.4. Mining by subspace analysis

Finding the frequent items only spots the candidate auxiliary objects that are frequently co-occurrent with the target, but they do not necessarily exhibit strong motion correlations to the target. For example, in Fig. 4.5, I_B is less correlated to the target T than I_A. We need to check if these candidates satisfy the motion correlation requirement of an auxiliary object. For each candidate, we can initialize a mean-shift tracker to find its correspondences in the successive image frames. If this tracker loses track for 4 frames in a row, we assert that this candidate is not suitable for tracking

and remove it. Otherwise, we can form the motion trajectories over the frames for a set of candidate auxiliary objects. Then, we employ a noise subspace analysis method to discover the potential multibody structure from motion and estimate the affine motion models between the object pairs.

The motion correlation between two moving objects can be very complicated and non-linear, but generally, linear motion models can be used as a good approximation. We extend the simple translational model in [111] to a more general affine motion model. When the points on two objects have an affine motion relation, they must reside in a linear subspace. Thus, identifying this subspace will lead to the estimation of the affine motion model.

At time t, one candidate auxiliary object $I_O \in F$ is represented as $\mathbf{x}_t = \{u_t^x, v_t^x\}^\top$ and $\{s_t^u, s_t^v\}$ where (u_t^x, v_t^x) are the coordinates of the center of I_O and s_t^u and s_t^v are the scales, respectively. Similarly the target T can be represented as $\mathbf{y}_t = \{u_t^y, v_t^y\}^\top$ and $\{s_t^u, s_t^v\}$. If I_O and T co-occur and have stable motion correlation, then I_O can be claimed as an auxiliary object. So the goal is to evaluate whether I_O and T have strong motion correlation in time window $[t - M + 1, t]$ given the trajectories of $\mathbf{y}_t$ and $\mathbf{x}_t$ within this time window.

Assume an affine motion model between candidate auxiliary object I_O and the target T for the period of frame $t - M + 1$ to frame t, which is specified by a 2×2 matrix $\mathbf{A}_t$ and a translation vector $\mathbf{b}_t = \{u_t^b, v_t^b\}^\top$, as

$$\mathbf{y}_t = \mathbf{A}_t \mathbf{x}_t + \mathbf{b}_t. \tag{4.1}$$

Subtract the mean $\bar{\mathbf{y}}_t$ of $\mathbf{y}_t$ and $\bar{\mathbf{x}}_t$ of $\mathbf{x}_t$ in the time window $[t - M + 1, t]$ and take the noise into consideration, the relation between I_O and T can be expressed with $\tilde{\mathbf{y}}_t = \mathbf{y}_t - \bar{\mathbf{y}}_t$ and $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \bar{\mathbf{x}}_t$, as

$$\tilde{\mathbf{y}}_t = \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{n}, \tag{4.2}$$

where $\mathbf{n}$ is a zero mean white noise with $E[\mathbf{n}\mathbf{n}^\top] = \sigma^2 \mathbf{I}$.

If we stack $\tilde{\mathbf{y}}_t$ and $\tilde{\mathbf{x}}_t$, the covariance matrix $\mathbf{C}$ can be expressed as

$$\mathbf{C} = E[\begin{pmatrix} \tilde{\mathbf{y}}_t \\ \tilde{\mathbf{x}}_t \end{pmatrix} (\tilde{\mathbf{y}}_t^\top, \tilde{\mathbf{x}}_t^\top)]. \tag{4.3}$$

It is clear that $rank(\mathbf{C}) \leq 2$ if there is no noise (*i.e.* $\mathbf{n} = 0$). This rank deficiency property is important in detecting the subspace due to motion correlation. In reality, because $\mathbf{n} \neq 0$, $\mathbf{C}$ is likely to have a full rank. Since the noise is additive, it is easy to prove that the 4D space spanned by $\left(\tilde{\mathbf{y}}_t^\top, \tilde{\mathbf{x}}_t^\top\right)$ is a direct sum of a signal subspace and a noise subspace. The signal subspace is up to rank 2 and corresponds to the large eigenvalues of $\mathbf{C}$, and the noise subspace corresponds to the smallest eigenvalues (*i.e.* σ). Therefore, we can check and threshold the eigenvalues to identify those subspaces.

Denote the estimated covariance matrix by $\hat{\mathbf{C}}$ and the covariance matrix of $\tilde{\mathbf{x}}$ by $\hat{\mathbf{C}}^x$, and we have

$$\hat{\mathbf{C}} = \sum_{i=0}^{M-1} \begin{pmatrix} \tilde{\mathbf{y}}_{t-i} \\ \tilde{\mathbf{x}}_{t-i} \end{pmatrix} (\tilde{\mathbf{y}}_{t-i}^\top, \tilde{\mathbf{x}}_{t-i}^\top) = \begin{pmatrix} \mathbf{A}_t \hat{\mathbf{C}}^x \mathbf{A}_t^\top + \sigma^2 & \mathbf{A}_t \hat{\mathbf{C}}^x \\ \hat{\mathbf{C}}^x \mathbf{A}_t^\top & \hat{\mathbf{C}}^x \end{pmatrix}. \tag{4.4}$$

Performing eigenvalue decomposition on $\hat{\mathbf{C}}$,

$$\hat{\mathbf{C}} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}, \tag{4.5}$$

we obtain the sorted eigenvalues $\{\lambda_1, \cdots, \lambda_4\}$ and orthonormal basis $\mathbf{Q}$. If there are more than 2 eigenvalues $\lambda_j^2 \gg \sigma^2$, this candidate is not an auxiliary object since its motion and the target's are not in one subspace.

$$\#\text{ of } \{\lambda_j^2 \gg \sigma^2\} \begin{cases} > 2, & \text{the candidate is not an AO} \\ <= 2, & \text{otherwise} \end{cases}. \tag{4.6}$$

If the candidate is an auxiliary object, we can estimate its affine matrix $\mathbf{A}_t$ with the property that the noise subspace is orthogonal to the signal subspace. The last two eigenvectors correspond to the noise subspace of $\hat{\mathbf{C}}$ are denoted as

$$\begin{pmatrix} q_{31} & q_{41} \\ q_{32} & q_{42} \\ q_{33} & q_{43} \\ q_{34} & q_{44} \end{pmatrix},$$

which are orthogonal to arbitrary vector $(\tilde{\mathbf{x}}_t^\top \mathbf{A}_t^\top, \tilde{\mathbf{x}}_t^\top)$ in the signal subspace. Substitute them back to $\hat{\mathbf{C}}$, and the 2×2 matrix $\mathbf{A}_t$ can be solved by

$$\mathbf{A}_t^\top \begin{pmatrix} q_{31} & q_{41} \\ q_{32} & q_{42} \end{pmatrix} + \begin{pmatrix} q_{33} & q_{43} \\ q_{34} & q_{44} \end{pmatrix} = 0. \tag{4.7}$$

Then, the translation vector $\mathbf{b}_t$ is obtained with $\bar{\mathbf{y}}$, $\bar{\mathbf{x}}$, and $\mathbf{A}_t$. This method gives an effective detection of auxiliary objects and efficient estimation of their affine motion models.

Such a mining process is meaningful, because it has learned a random field. We denote the motion of the target T by $\mathbf{y}$ and those of the auxiliary objects by $\mathbf{x}_k, k = 1, \ldots, K$, where K is the number of auxiliary objects. They constitute a random field. The pair-wise potentials $\psi_{k0}(\mathbf{x}_k, \mathbf{y})$ are actually learned as a by-product of this mining process, as

$$\psi_{k0}(\mathbf{x}_k, \mathbf{y}) \propto e^{-\frac{(\mathbf{y}-\mathbf{A}_k\mathbf{x}_k-\mathbf{b}_k)^\top(\mathbf{y}-\mathbf{A}_k\mathbf{x}_k-\mathbf{b}_k)}{2\sigma^2}}, \tag{4.8}$$

where σ^2 is derived from the small eigenvalues of $\mathbf{C}$ in Eq. 4.3. In many cases, auxiliary objects share almost the same motion as the target, *e.g.*, the torso and the target head. Therefore, we can use a Gaussian distribution to characterize those potentials. The mean of the Gaussian is given by

$\mathbf{A}_k$ and $\mathbf{b}_k$, which is the affine motion model estimated for the kth auxiliary object. Note from now on, the subscript indicates the index of an auxiliary object instead of the time step.

4.2. Collaborative Tracking

It is clear that CAT is not tracking a single target, but a random field. This random field among auxiliary objects and the target is hidden and needs to be inferred from image evidence. We formulate this problem under a Markov network with a special topology, as shown in Fig. 4.6, where we only assume pair-wise connections between the target $\mathbf{y}$ and the auxiliary object $\mathbf{x}_k$ and there are no connections among auxiliary objects. Each of them is associated with its image evidence $\mathbf{z}_k$. We denote $\mathbf{Z} = \{\mathbf{z}_k, k = 0, \ldots, K\}$, where K is the number of AOs and $\mathbf{z}_0$ is the observation of $\mathbf{y}$ (*i.e.* the target). The core of tracking is to estimate the posteriors $p(\mathbf{y}|\mathbf{Z})$ of the target and $p(\mathbf{x}_k|\mathbf{Z}), k = 1, \ldots, K$, for the auxiliary objects.

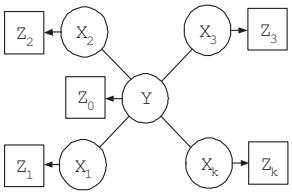


Figure 4.6. The star topology of a random field. The hidden motion parameter of the target is denoted as $\mathbf{y}$ with the image observation $\mathbf{z}_0$. The motion parameters of the auxiliary objects are denoted as $\mathbf{x}_k$ with their respective observations $\mathbf{z}_k$.

For such a graph with a star topology, a belief propagation algorithm with 2-step message passing gives the exact estimates of the posteriors. Denote by $p(\mathbf{z}_i|\mathbf{x}_i)$ the local likelihood and by $\phi_k(\mathbf{x}_k)$ the local prior such as the dynamics prediction prior for $\mathbf{x}_k$. Each pair of the target and an auxiliary object $\mathbf{x}_k$ bears a pair-wise potential $\psi_{k0}(\mathbf{x}_k, \mathbf{y})$ learned in the subspace-based mining process, as described in Sec. 4.1.D. $m_{k0}(\mathbf{y})$ represents the message passed from the kth auxiliary object to the target and $m_{0k}(\mathbf{x}_k)$ is the message from the target to the kth auxiliary object.

At the first iteration step, the target $\mathbf{y}$ receives all the messages m_{k0} from every auxiliary object $\mathbf{x}_k$, then propagates the message back to them at the second iteration. This message passing mechanism implies a collaborative way of tracking. Notice that if the target and the auxiliary objects are independent, their independent motion estimates are $\hat{p}_k(\mathbf{x}_k|\mathbf{Z}) \propto \phi_k(\mathbf{x}_k)p(\mathbf{z}_k|\mathbf{x}_k), k = 1, \ldots, K$. The relation between the true estimates and independent estimates is simply captured by a fixed-point equation of the messages:

$$p(\mathbf{y}|\mathbf{Z}) \propto \hat{p}_0(\mathbf{y}|\mathbf{Z}) \prod_k m_{k0}(\mathbf{y}), \tag{4.9}$$

$$m_{k0}(\mathbf{y}) = \int_{x_k} \hat{p}_k(\mathbf{x}_k|\mathbf{Z})\psi_{k0}(\mathbf{x}_k, \mathbf{y})d\mathbf{x}_k, \tag{4.10}$$

$$p(\mathbf{x}_k|\mathbf{Z}) \propto \hat{p}_k(\mathbf{x}_k|\mathbf{Z})m_{0k}(\mathbf{x}_k) \quad k = 1, \ldots, K, \tag{4.11}$$

$$m_{0k}(\mathbf{x}_k) = \int_y \hat{p}_0(\mathbf{y}|\mathbf{Z}) \prod_{\mathbf{x}_i \setminus \mathbf{x}_k} m_{i0}(\mathbf{y})d\mathbf{y}. \tag{4.12}$$

This suggests that we can use individual trackers for the target and auxiliary objects. But these sets of individual trackers are not independent, as they need to combine their local estimates and the messages from others, and iterate. Such a collaborative mechanism leads to a very efficient solution to tracking the random field. Thus, even if our new approach involves the tracking of a set of auxiliary objects (*e.g.* by mean-shift), the computation is manageable because of the efficiency of the collaborative way.

Compared with a single tracker for the target, the involvement of auxiliary objects can reduce the uncertainty of the motion estimation of the target and thus make the tracking more confident. We can prove this in a special case when setting both the potential $\psi_{k0}(|\mathbf{x}_k - \mathbf{y}|)$ to be a Gaussian $N(\mu_{k0}, \mathbf{\Sigma}_{k0})$ and the local likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ to be a Gaussian $N(\hat{\mu}_k, \hat{\mathbf{\Sigma}}_k)$ (we ignore the local

prior without losing generality). Under this setting, the closed-form belief propagation gives:

$$\Sigma_0{}^{-1} = \hat{\Sigma}_{\mathbf{0}}^{-1} + \sum_{k=1}^{K} (\hat{\Sigma}_k + \Sigma_{k0})^{-1}, \tag{4.13}$$

$$\mu_0 = \Sigma_0 (\hat{\Sigma}_{\mathbf{0}}^{-1} \hat{\mu}_0 + \sum_{k=1}^{K} (\hat{\Sigma}_k + \Sigma_{k0})^{-1} (\hat{\mu}_k + \mu_{k0})), \tag{4.14}$$

where (μ_0, Σ_0) is the target's posterior when tracking the random field. If we assume the local priors to be Gaussian, this result still holds but now $(\hat{\mu}_k, \hat{\Sigma}_k)$ refers to the local posterior.

Eq. 4.13 makes it clear that Σ_0 is always less than $\hat{\Sigma}_0$ since these covariance matrices are positive definite and different motion parameters are uncorrelated. Therefore, the confidence of the collaborative estimate of the target is higher than that produced by a single target tracker.

4.3. Inconsistency and Robust Fusion

The closed form analysis for the collaborative tracking can be explained in the view of information fusion. When the connection potentials between the target and the auxiliary objects are set to be extremely tight, *i.e.*, the covariance of Σ_{k0} is a zero matrix $\mathbf{0}$, this belief propagation is equivalent to the best linear unbiased estimator (BLUE) for $\mathbf{y}$; if they are extremely loose, *i.e.* Σ_{k0} approaches infinity, it becomes an independent estimation; otherwise, it is similar to covariance intersection [50].

However, there is a hidden assumption for this conclusion, *i.e.*, the information from all the sources must be consistent. In simple terms, they must more or less agree with each other. But in reality, this assumption may not be valid, when the estimates from the individual trackers may be completely different or inconsistent for many reasons. If we use the above mentioned method to fuse these inconsistent estimates, we may end up with an estimate that is completely wrong but of a very high confidence. Such an adverse estimation makes no sense and should be avoided. Thus,

it is desirable to have a mechanism to detect the inconsistency and identify the outliers for a robust fusion.

We define two Gaussian sources as *consistent* if the variance in the compatible function of these two Gaussian sources approaches zero using EM estimation. Gang *et. al.* [40] gave a new theorem to measure the inconsistency. We employ the following two criteria that are very useful for detecting the pair-wise inconsistency. The proofs are presented in Appendix B.

Theorem 2. *Considering two Gaussian sources* $N(\mu_1, \Sigma_1)$ *and* $N(\mu_2, \Sigma_2)$*, where* $\mu_1, \mu_2 \in R^n$*, the two sources are inconsistent if:*

$$\frac{1}{n}(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2) \geq 2 + \sqrt{C_p} + \frac{1}{\sqrt{C_p}}, \tag{4.15}$$

where C_p *is the* 2*-norm conditional number of* $\Sigma_1 + \Sigma_2$*, and they are consistent if:*

$$\frac{1}{n}(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2) < 4. \tag{4.16}$$

Although these are sufficient conditions in general cases, they are actually also necessary conditions when $n = 1$. These criteria enable simple and quick detection of pair-wise inconsistency. Then, the estimation that is inconsistent with all the others will be regarded as an outlier. The outlier can be the target or the AOs. If the target is an outlier, we assert that the target is experiencing occlusion or drift, and suspend the mining process temporarily. In this case, we can give an estimation of the target purely based on the predictions from the auxiliary objects, and search for the image evidence. If the outlier is an auxiliary object, we simply exclude this auxiliary object for fusion. After excluding the outliers, we perform belief propagation again on the rest of the network and employ the target tracker to locate the target precisely. When the majority are not consistent which means the target estimate can not be verified, a tracking failure is asserted.

4.4. Experiments and Discussions

4.4.1. Experiment settings

We substantialized and implemented the proposed CVT algorithm in a head tracking system, where the head tracker is a contour-based elliptical tracker similar to [8] and initialized by the frontal face detector [59, 58], and the auxiliary trackers are mean-shift trackers. Since a fixed number of edge points along the ellipse are matched, the single head tracker is quite computationally efficient and runs at over 50 fps. Although the single head tracker is relatively robust to illumination and view changes, it is vulnerable to the clutter background, motion blur and occlusions. In our experiments, we compare the proposed CVT algorithm with the single head tracker in a large number of real-world sequences captured in unconstrained environments including both indoor and outdoor scenes. These extensive experiments and exciting results have demonstrated the advantages of the CVT algorithm. Furthermore, we apply the same CAT algorithm to people tracking based on an appearance-based torso tracker to exhibit the applicability of the proposed idea to different types of targets.

The motion parameter $\mathbf{y} = \{u, v, s^u, s^v\}$ to be recovered includes the location (u, v) and the scales s^u and s^v. The color segmentation and the mean-shift tracker work in the normalized R-G color space with 32×32 bins. Without code optimization, our C++ implementation of CAT runs comfortably at around 10 fps on average on a Pentium 3GHz desktop for 320×240 images depending on the number of auxiliary objects discovered.

4.4.2. Quantitative experiments

For a quantitative evaluation, we manually labelled the ground truth of the sequences `kid in yellow`, `dancing girl` and `birthday kid` for 1200, 1600 and 1460 frames respectively.

The evaluation criteria of tracking error are based on the relative position errors between the center of the tracking result and that of the ground truth, and the relative scale normalized by the ground truth scale. Ideally, the position differences should be around 0, and the relative scales 1.

As shown in Fig. 4.7, Fig. 4.8 and Fig. 4.9, the position differences of the results in the CAT are much smaller than that of the single head tracker and the relative scales have much less fluctuations around 1. It demonstrates the advantages of the CAT, *i.e.* reducing the false alarm rate and the estimation covariance. Note that at the end of the sequence `kid in yellow`, the single tracker happens to track the head by chance after the drift. Although the CAT tracker loses track at around frame 1100 for several frames, it is able to recover promptly because of the auxiliary objects.

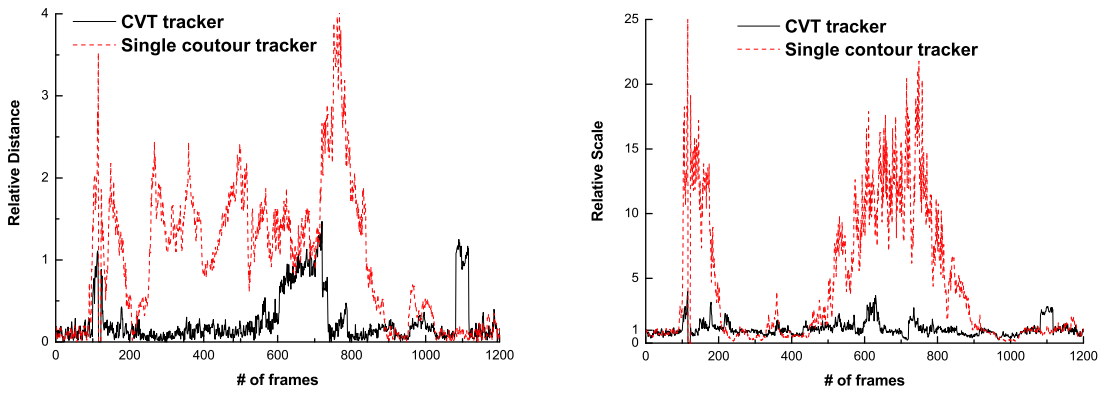


Figure 4.7. Quantitative comparison: (left) position errors, (right) scale errors, [`kid in yellow,1200 frames`].

Some key frames are shown in Fig. 4.10 [1]. The first row shows the results of the single head tracker where the highlighted solid-yellow box indicates the location of the head. The second row shows the segmentation and mining results, where each green rectangle indicates an item in the current frame. The numbers in blue at the corner show the item labels of the candidate auxiliary objects. The third row illustrates the fusion results. Each blue box is the estimate of

[1] All the faces shown in this chapter were mosaicked afterwards for privacy protection.

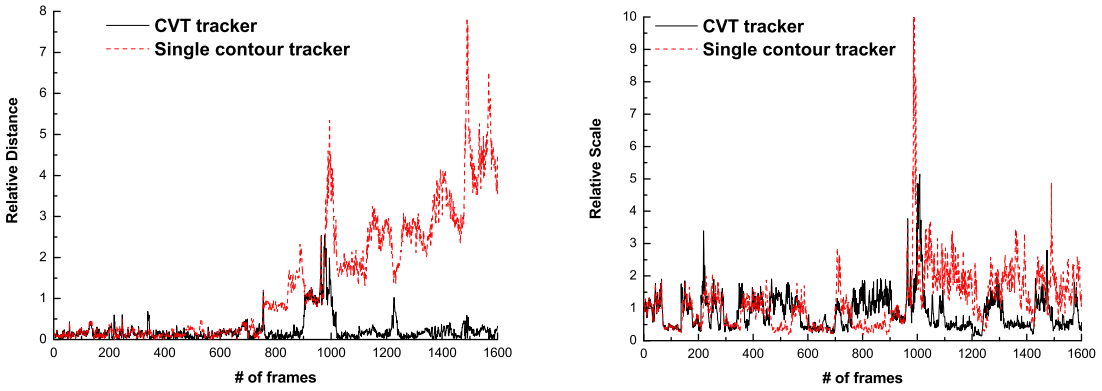


Figure 4.8. Quantitative comparison: (left) position errors, (right) scale errors, [dancing girl,1600 frames].

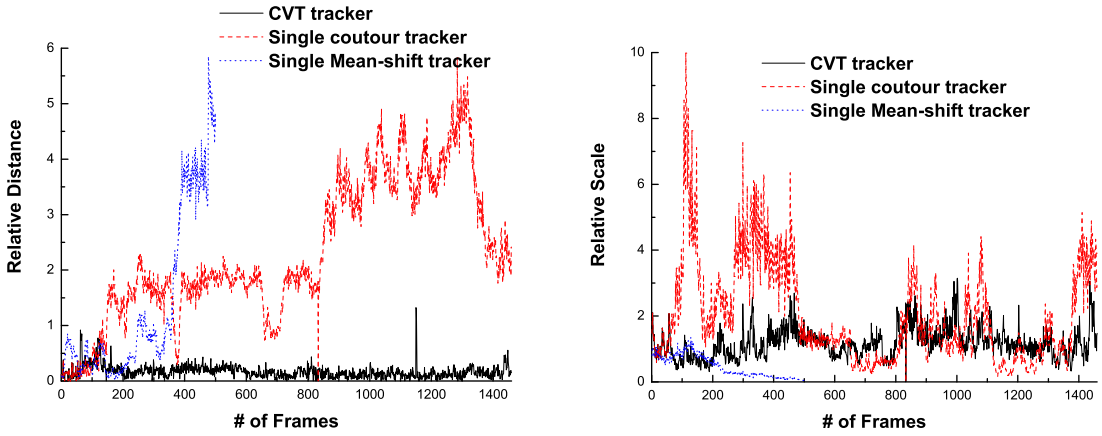


Figure 4.9. Quantitative comparison: (left) position errors, (right) scale errors, [birthday kid,1460 frames].

the head from different sources (*i.e.* the target or the auxiliary objects trackers). The white box indicates that estimate is regarded as an outlier. The dark red box is the final result of the fusion. The corresponding labels of the auxiliary objects are shown at the bottom-right corner. The final tracking results of CAT are shown in the 4-th row as highlighted solid-yellow box, and the dash-red boxes are the auxiliary object trackers.

4.4.3. Occlusion and drift

Fig. 4.10 samples the results on the sequence `kid in yellow` which is very challenging due to a serious occlusion, target out-of-range and clutter. When the head moves outside the upper boundary at frame 113, the single head tracker drifts to a false positive in the cluttered background and is unable to recover. In contrast, the CAT tracker asserts the occlusion and keeps tracking correctly. It freezes the head tracker temporarily and re-initializes it based on the predictions provided by the auxiliary objects. When the kid is walking in front of the bush, the background is so cluttered that it causes big troubles to the edge-based tracker. On the other hand, CAT discovers several auxiliary objects, *i.e.* the shirt and short pant, which are quite stable and provide roughly correct estimates of the head location and rescue the head tracker from the drift at frame 736.

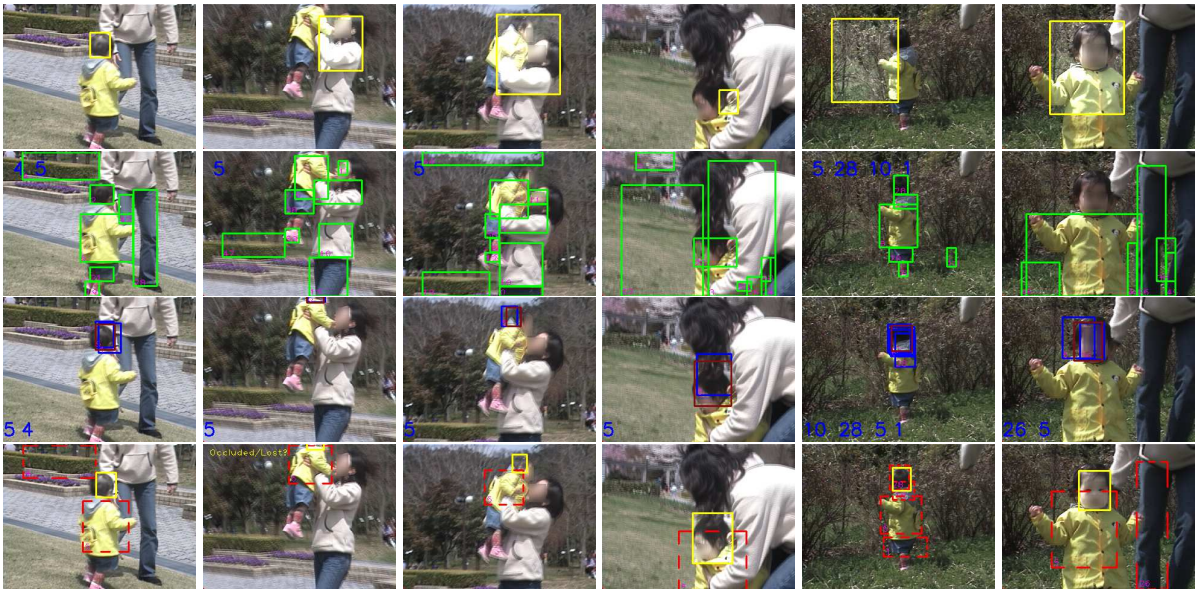


Figure 4.10. Frame # 50, 113, 124, 229, 736 and 866 of `kid in yellow,1200 frames`. (1st row) the head tracker, (2nd row) the mining results, (3rd row) the fusion results,(4th row) the CVT tracker.

4.4.4. Quick movement and camouflage

As shown in Fig. 4.11, the sequence `dancing girl` presents quick movements and camouflage. All the girls are similar in terms of their appearances. This is extremely difficult for a single head tracker to work, but CAT comfortably handles such a challenge. During the dancing, CAT gradually discovers the spatial relations between the target (the girl of interest) and the adjacent context *e.g.* other girls' shirts, although such relations are only valid in a short time interval. At frame 757, the single head tracker is trapped by the shoulder of the girl and unable to recover. At frame 758, the CAT tracker identifies this false alarm and pulls back the head tracker with the help of the predictions of the AOs that are close to the true target. At frame 1234, the girl of interest suddenly bows down, CAT detects the tracking failure and resumes tracking quickly. CAT can comfortably track over 1600 frames for this highly dynamic sequence until the target moves outside the left boundary for several seconds.

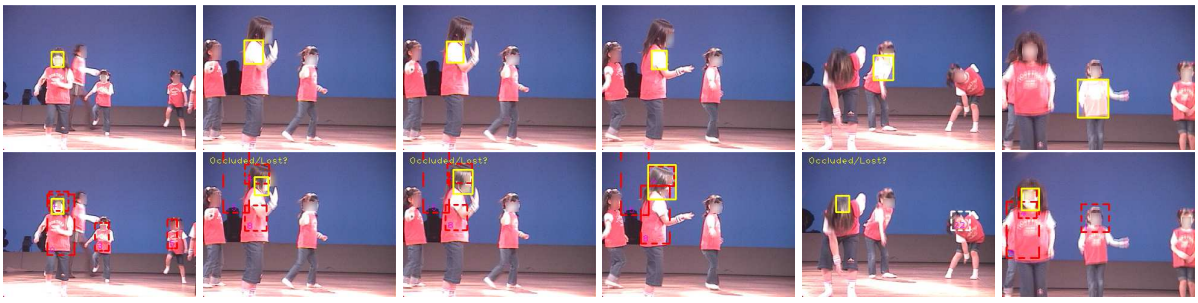


Figure 4.11. Frame # 67, 757, 758, 764, 1234 and 1372 of `dancing girl,1600 frames`. (top) the head tracker, (bottom) the CVT tracker.

4.4.5. Scale and view changes

We show the tracking performance when the target undergoes large scale and view changes and demonstrate the transition of the auxiliary objects in the sequence `kid&dad` (Fig. 4.12). For the single head tracker, when the scale of the head becomes very small, it drifts to the torso of the kid

from frame 69 and fails the tracker. During the first 300 frames, the dad walks with the kid with quite stable motion correlation. This is discovered by CAT and the region of dad's shirt is mined as the auxiliary object to help track the kid's head. When they move close to the camera, the scale and the view change dramatically so that the learned relation between dad's shirt and the kid's head no longer holds. Fortunately, CAT spots that the hat is a good auxiliary object at large scale and guides the tracking. At the end of the sequence, the head is completely occluded by the hat for several seconds. Although this is impossible to recover, CAT detects and reports the tracking failure, while the single head tracker tends to drift to a false positive without notice.

Figure 4.12. Frame # 52, 69, 70, 313, 555 and 616 of `kid&dad,617 frames`. (top) the head tracker, (bottom) the CVT tracker.

4.4.6. Cluttered background

In sequence `birthday kid`, the target head experiences large out-of-plane rotation and the appearances change greatly, as shown in Fig. 4.13. For the contour tracker, when the rear head is in the dark background, no good observation is available around the head so the contour tracker drifts to the torso and other elliptical regions, and is unable to recover. For the CAT tracker, with the help of the auxiliary objects, the tracker either keeps tracking in the tough situations or recovers from drifting in several frames. Note the auxiliary objects discovered can be some objects with inherent

relations with the target, such as the hat and short pant, or just something that happens to have temporary relations, such as the refrigerator or the gift box. This real-world sequence demonstrates the advantages of the auxiliary objects for long-duration tracking.

Figure 4.13. Frame # 0, 72, 93, 170, 578, and 1455 of `birthday kid,1460 frames`. (top) the head tracker, (bottom) the CAT tracker.

As shown in Fig. 4.14 (`swimming boy`), the background is quite cluttered due to the texture of water and other people, which makes the single head tracker hopeless. The single head tracker is easily distracted by the edges in the background and drifts away. On the other hand, CAT discovers the two blue life buoys and the swimming hat and uses them as the auxiliary objects. When the boy jumps towards his mother's arms, CAT uses the life buoys as well as the orange box on the bank to help locate his head accurately, which is difficult for the single head tracker. Note that at the end of this sequence, the kid's head is occluded by his mom's head and CAT switches to the mom. This is reasonable because the auxiliary objects can not differentiate the two heads at the same location.

4.4.7. More people tracking results

To demonstrate the generalization ability of the proposed method, we apply the context-aware tracking algorithm to people tracking based on an appearance-based torso tracker. As shown in Fig. 4.15 [26], when the person to track is occluded by his friends around frame 56, the single

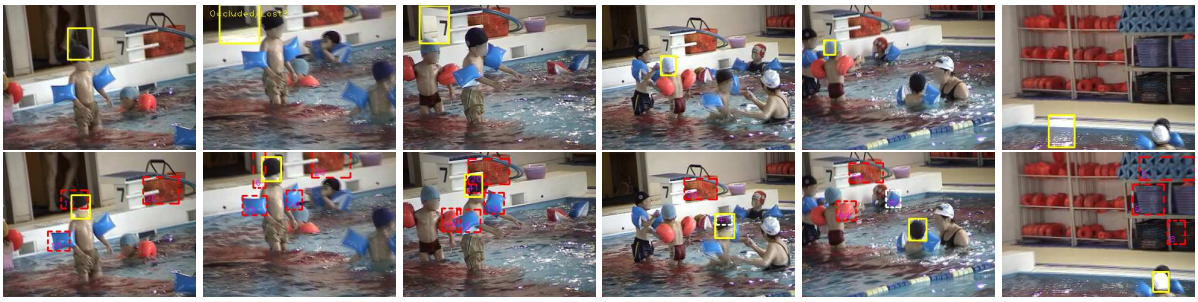


Figure 4.14. Frame # 87, 131, 334, 526, 578 and 848 of `swimming boy,900 frames`. (top) the head tracker, (bottom) the CVT tracker.

torso tracker loses the target and drifts away. In contrast, since the other pedestrians serve as the temporary contexts, they can help the CAT tracker keep following the target. In addition, after frame 135 the context information helps to prevent the tracker from drifting to the person next to the target though both persons have very similar appearances. Another example sequence is shown in Fig. 4.16 where an athlete in a marathon is tracked with natural lighting changes and view changes are present.

Figure 4.15. Frame # 0, 40, 56, 68, 135, and 425 of `three past shop,425 frames`. (top) the torso tracker, (bottom) the CAT tracker.

4.4.8. Discussions

As demonstrated in a large number of challenging sequences, there are two primary scenarios when the auxiliary objects greatly help the tracking: 1) some auxiliary objects have persistent relations to the target and present fairly accurate estimates although these relations may not be foreseen; 2)

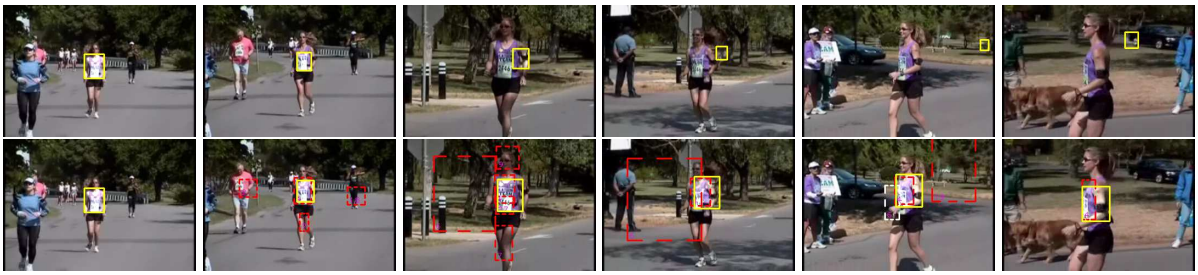


Figure 4.16. Frame #1, 72, 468, 504, 582, and 625 of `marathon, 625 frames`. (top) the torso tracker, (bottom) the CAT tracker.

a number of auxiliary objects have transitional relations to the target and the majority of them can give rough correct estimates in a short time interval. In the cases of occlusion or drift, it is not likely that all the auxiliary objects are occluded or all auxiliary trackers lose track at the same time, since the auxiliary objects may not be located in a close vicinity of the target. The mechanism of robust fusion can identify the inconsistency induced by occlusions or drifts. There are some extremely difficult cases, *e.g.* the target is occluded for long time, and CAT fails reasonably because on-line data mining may not be invoked at all. Or only a couple of auxiliary objects discovered and they do not agree with each other about the target motion, which implies insufficient context information to verify the tracking results. For these cases, the advantage of CAT is the ability to detect and report the failure, and leave the system to other means of re-initialization, while the single tracker has no reliable mechanism to report the failure but keeps tracking aimlessly and regardlessly.

We have proposed a novel solution to robust long-duration tracking by considering the context of the target. By integrating an unsupervised data mining procedure, a set of auxiliary objects are discovered on the fly which provide extra measurements to the target and reduce the uncertainty of the estimation. In addition, the learned motion correlations among the auxiliary objects and the target serve as a strong cue to verify the tracking results to handle short-term occlusion or tracking loss. The auxiliary objects are automatically discovered without supervision and do not incur much

extra computation, which makes the approach generally applicable to a wide spectrum of tracking scenarios.

For future directions, we will study the relation between the number of auxiliary objects discovered and the confidence level of the verification. Another important issue to investigate is how to compromise the need for a quicker initial mining procedure within a shorter time window which may find more auxiliary objects and a longer time window which may find less auxiliary objects but with a high reliability.

CHAPTER 5

Attentional Visual Tracking

Long-duration tracking of general targets is quite challenging for computer vision due to the large uncertainties in a target's visual appearance and the unconstrained environments which may be cluttered and distractive. However, tracking has never been a challenge to the human visual system. In contrast to the tremendous challenges encountered in developing tracking algorithms, being able to persistently follow moving objects seems to be a very basic functionality in human visual perception. It is so natural and intuitive that we may not be aware of how complex it is. Although the details in human perception on visual dynamics are still largely mysterious, the studies in psychology, neuroscience and cognitive sciences have obtained substantial evidence and interesting findings, based on which several hypothetical theories have been proposed [70]. For example, evidence shows that human visual perception is inherently selective. Perceiving realistic scenes requires a sequence of many different fixations through the saccadic movements of the eye. Even when the eye is fixated on a particular location, the act of *visual attention* (like the movements of an internal eye or the so-called "mind's eye") selects and determines what subset of information of the retinal image gets full processing [72]. This ability to engage in some flexible and selective strategies for processing different aspects of visual field is generally referred to as *visual attention*.

In human visual attention, *spatial selection* is one important aspect in which human visual perception system selectively samples the retinal image and concentrates the resources to only process a restricted portion of information. Another important aspect is *property selection* in which people sequentially perceive different properties or features of the same object, *e.g.* its color,

its shape, its texture, and its structures. Psychological and cognitive findings suggest that these selective attention mechanisms are necessary and critical for human visual tracking. An interesting question is how we can take advantage of these studies to develop more powerful visual tracking algorithms.

This chapter presents new visual tracking approaches that reflects some findings of selective visual attention in human perception. Recent studies from the 90s have indicated that *selective attention* may act in both early and late stages of visual processing but under different conditions of perceptual load [**70**]. *Early selection* may be based on innate principles obtained through evolution, while *late selection* is learned through experiences. By integrating these two selection stages, we develop and implement two attentional visual tracking (AVT) algorithms. One mainly reflects the spatial selection mechanism by representing targets with a pool of salient image patches and dynamically attending to the discriminative subset of patches. The other represents targets with Markov random fields (MRF) of interest features and tunes the matching criteria to adjust the emphases on the properties of local appearances and structures automatically. These attentional visual tracking algorithms adaptively focus on the discriminative characteristics of the targets and achieve fairly prominent performances in tracking diversified targets without any prior knowledge.

5.1. Spatial Selection for Attentional Tracking

In this section, we propose a new visual tracking approach reflecting some aspects of spatial selective attention by connecting the low-level matching to the early attentional selection and the high-level process to the late selection. Specifically, the early selection process extracts a pool of *attentional regions* (ARs) that are defined as the salient image regions that have good localization properties, and the late selection process dynamically identifies a subset of discriminative attentional regions (D-ARs) through a discriminative learning on the historical data on the fly. The computationally demanding process of matching of the AR pool is done in an efficient and innovative way by using the idea in the locality-sensitive hashing (LSH) [**43**, **19**, **3**] technique.

The proposed spatially selective attentional visual tracking (SS-AVT) algorithm is general, robust and computationally efficient. Representing the target by a pool of attentional regions makes SS-AVT robust to appearance variations due to lighting changes, partial occlusions and small deformation. Spatial attentional selection of ARs allows SS-AVT to focus its computational resources on more informative regions to handle distractive environments and targets with complex shapes. Pre-indexing the features of ARs based on LSH enables fast matching in order to search a large motion parameter space. In addition, SS-AVT can be used as a region tracking tool for tracking general objects without any prior knowledge. These merits have been shown in extensive results on a variety of real-world sequences.

This work is different from some recent work on on-line selection of discriminative features [**14**] and other adaptive methods [**5**, **28**, **49**], in that SS-AVT does not select global features but spatially-distributes local attentional regions so as to enable a broader and a more robust selection. In addition, SS-AVT is also quite different from the fragment-tracking [**1**] where the target is evenly divided into a fixed number of fragments in a pre-defined way with no selection.

5.1.1. Overview of spatial selection for attentional tracking

Selective attention is crucial to visual perception, because the amount of information contained in visual scenes is far more than what we can process at one time and thus the visual system has to sample visual information over time by some inherently selective perceptual acts, including spatial selection that directs the attention to a restricted region of the visual field. Selective attention may be made possible by two kinds of heuristics. One is based on innate principles obtained through evolution, and could be performed in the early stage of visual processing. The other one is learned through experience and might happen later in visual processing. Both are important in the human visual system.

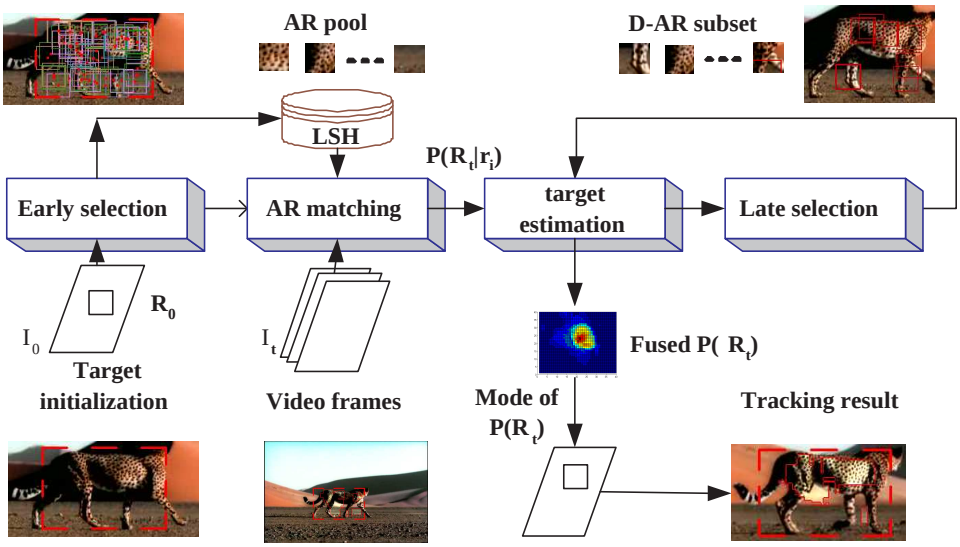


Figure 5.1. Spatial selection for attentional visual tracking.

As summarized in Fig. 5.1, the proposed attentional visual tracking reflects these perceptual findings of spatial selection in visual attention. SS-AVT has 4 important processes:

- **Early attentional selection**. As the first step, it extracts informative and salient image regions called *attentional regions* (ARs) from images. This is a low-level process, as it is

only concerned on local visual fields. In this paper, we treat those image regions that have good localization properties as ARs, and the AR is characterized by its color histogram;

- **Attentional region matching**. Once a pool of ARs is extracted by the early selection process, they will be used to process an incoming image to localize their matches. An innovative method is proposed to conquer the large computational demands, by pre-indexing the features of ARs. For each frame, the matching set of each AR is obtained and used to estimate a belief of the target location;
- **Attentional fusion and target estimation**. The beliefs of all the ARs are fused to determine the target location. A subset of ARs have larger weights in the fusion process, because they are more discriminative. This subset of ARs is obtained by the late selection process in the previous time frame;
- **Late attentional selection**. This process reflects some higher level processing to learn and adapt to the dynamic environments. Based on the collected history tracks of ARs, a discriminative selection is performed to identify a subset of most discriminative ARs (or D-ARs) that exhibit the distinctive features of the target from the environments. They will have larger weights in the attentional fusion process at the next frame.

5.1.2. Components in spatially selective attentional tracking

5.1.2.1. Early attentional selection. Visual information is so rich that the human visual system has a selective attention mechanism to sample the information over time in processing. Early attentional selection that is believed to act in the very early stage of visual perception performs the initial pre-filtering task, which should not involve much higher level processing such as object recognition. Early selective attention is likely to be based on innate principles of human perception,

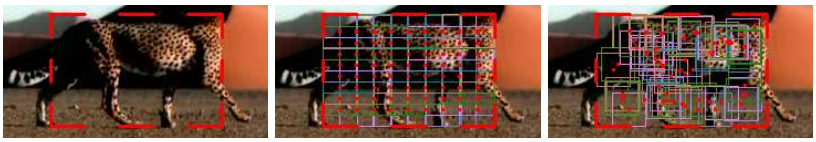
e.g., to attend certain information that is evolutionarily advantageous. For example, moving objects are generally important for survival and appear to play an important role in early attention.

We describe a spatial selection method for this early attentional process. We call the selected image region as *attentional regions* (ARs). As discussed before, motion detection appears to play an important role in early attention. Therefore, the selection of attentional regions should be sensitive to motion (*i.e.*, informative) but insensitive to noise (*i.e.*, stable). Mathematically, any change in the appearance of such an AR should correspond to a unique motion estimation, and the small differences between two appearance changes should not lead to dramatically different motion estimates (*i.e.*, well-behaved).

In view of this, we choose to use the criterion and the region extraction method described in [23] that views the stability of an image region in motion estimation from a system theory perspective. The appearance change of an image region is treated as measurement of the motion and is viewed as the system states. For some image regions, *e.g.*, homogeneous regions, the system states (motions) are *unobservable* from the measurements, *i.e.*, the motions of these regions are not fully recoverable from their appearance changes. Thus, they should not be attentional regions. In addition, image regions that lead to unstable systems, *i.e.*, small appearance changes that result in dramatically different motion estimates, should not be attentional regions. Therefore, attentional regions can be selected by finding those regions that generate observable and stable systems. It was proved [23] that because an image region is characterized by its feature histogram, the stability of the linear motion estimation system can be evaluated by checking the condition number of a matrix that is only related to the properties of the corresponding image region. A more stable system has a lower condition number. Thus, in the proposed AVT algorithm, we select the pool of ARs by locating and extracting those salient image regions.

Specifically, at the first frame I_0, given the target initialization rectangle $\mathbf{R_0}$, we evenly initialize $N_{max} = 100$ tentative ARs inside the target. With an efficient gradient descent search algorithm [23], the tentative ARs converge to positions where the corresponding condition numbers are local minima. By removing the duplicated tentative ARs that have converged to the same location and those that have large condition numbers, the selected AR pool is obtained and denoted by $\{\mathbf{r}_1, \cdots, \mathbf{r}_N\}$. Their relations to the target are recorded for future target estimation in subsequent tracking. The number N of ARs is automatically determined by the early selection process itself, depending on targets, *e.g.*, we have observed $N = 60 \sim 70$ for large and complex objects and $N = 30 \sim 40$ for small and simple objects in our experiments. Then, the color histograms of $\{\mathbf{r}_1, \cdots, \mathbf{r}_N\}$ are obtained as the feature vectors $\{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ with D bins, *i.e.*, $\mathbf{p}_i = \{p_{i1}, \cdots, p_{iD}\}$.

As the color histograms on various image regions need to be calculated, the integral histogram technique [74] can be applied to save computation. In AVT, we implement a modified version of an integral histogram that is able to retrieve histograms at arbitrary locations in constant time, but also consumes moderate memory when using high resolution color histograms. Although the sizes and shapes of different ARs are not necessarily identical, to be able to process ARs in a uniform way, we impose that all ARs be the same size and shape, *i.e.*, 30×30 squares initially. An example of the early selection of attentional region pool is shown in Fig. 5.2.



(a) initialization. (b) initial search positions of ARs (c) the pool of ARs.

Figure 5.2. Early selection of the attentional region pool.

5.1.2.2. Attentional region matching. For each frame I_t at time t, to locate the correct target position, all hypotheses in motion parameter space have to be evaluated to find the best matches to the ARs in the AR pool. Because the prior knowledge of the dynamics of the ARs is generally unavailable, exhaustively searching the motion parameter space can provide optimal performance. Although this is computationally demanding, we have an innovative solution that significantly reduces the computation to allow close to real-time performance. This solution is based on the idea of the locality-sensitive hashing (LSH) [19], a powerful database retrieval algorithm.

Each AR needs to examine a large number of motion hypotheses. In this thesis, the motion parameters include location (u, v) and scale s. Each motion hypothesis corresponds to a candidate image region. For all target hypotheses, all image patches $\mathbf{r}_c$ with the same size as ARs within the searching range of one AR constitute the *candidate region set* whose D dimensional color histograms are denoted as $\{\mathbf{q}_1, \cdots, \mathbf{q}_M\}$, where M is the size of the set. Generally the candidate region set has thousands of entries. We employ the Bhattacharya coefficient to measure the similarity of two histograms $\mathbf{p}$ and $\mathbf{q}$, which is equivalent to Matusita metric [32] in L_2 distance form

$$d(\mathbf{p}, \mathbf{q}) = \sum_{j}^{D} \|\sqrt{p_j} - \sqrt{q_j}\|^2. \tag{5.1}$$

Matching a feature vector can be translated to querying a database for the nearest neighbor points in the feature space. The worst case complexity is obviously linear, but this is not good enough. A significant speed-up can be achieved if the database can be pre-indexed. Locality-sensitive hashing (LSH) proposed by Indyk and Motwani [43] in 1998 and further developed in [19] aims to solve the approximate Nearest Neighbor (NN) problem in high dimensional Euclidean space. LSH provides a probabilistic approximation to this problem by randomly hashing the database with L locality-sensitive hashing functions, and only the points in the union of the

hashing cells that the query point falling in are checked for nearest neighbors. This will lead to computational savings comparing with checking all the entries in the database. The idea is illustrated in Fig. 5.3. We refer readers to [43, 19] for details.

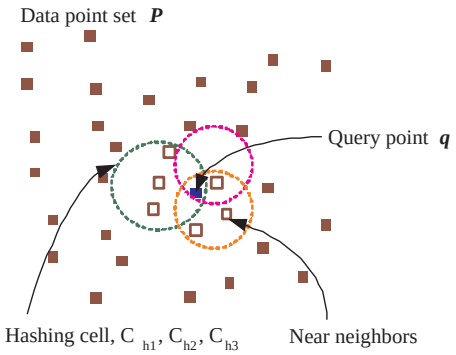


Figure 5.3. Illustration of query with LSH.

LSH has been applied in texture analysis [27] and fast contour matching [29]. To the best of our knowledge, LSH has not been used for on-line tracking before, although another database technique (K-D Trees) has been used for off-line (non-causal) tracking [11] by hashing the whole video sequence. When incorporating LSH into on-line visual tracking, there is a fundamental difference from database applications. In database applications, the indexing is done off-line and thus the computational overhead of indexing is not a critical issue. In our on-line tracking scenario, on the contrary, the indexing overhead cannot be ignored because both indexing the database and retrieving the database are performed during the tracking process. So computational costs of both indexing and querying are critical. This turns out to be very important in the SS-AVT implementation.

Now we have two data sets: one for the AR pool with size N and the other for the *candidate region set* with size M. Typically, N is within one hundred and M is several thousands. The worst case of complexity in matching is $O(N \times M)$. As discussed before, this complexity can be further

reduced by applying LSH. Because the overhead of indexing needs to be considered, which data set should be chosen to be the database for LSH? If choosing the candidate set as the database, we find that the indexing overhead is not worth the gain for a limited number of queries from the AR pool. When we treat the AR pool as the LSH database, the computational gain is significant. The detailed complexity analysis will be present in a later section. After querying all candidate regions $\mathbf{r}_c$ with feature vectors $\mathbf{q}_c$ using LSH, the near neighbors within d_t in Matusita distance of each AR $\mathbf{r}_i$ are obtained and denoted as matching set $S_{\mathbf{r}_i} = \{\mathbf{r}_c | d(\mathbf{p}_i, \mathbf{q}_c) \leq d_t\}$.

5.1.2.3. Attentional fusion and target estimation. As described in the previous subsection, for each AR $\mathbf{r}_i$, the attentional region matching process outputs a matching set. Based on the recorded geometrical relation between this AR and the target (relative translation and scale in our implementation), the belief of this AR is the probability distribution of target location (u_t, v_t) of $\mathbf{R}_t$ given $\mathbf{r}_i$'s matching set, denoted by $P(\mathbf{R}_t|\mathbf{r}_i)$, which is approximated based on the set of matched candidate $\mathbf{r}_c \in S_{\mathbf{r}_i}$.

To estimate the target location and scale, the beliefs of all the ARs need to be fused. Because some ARs may have a substantial spatial overlap in images, their beliefs may be correlated. This dependency may complicate the exact fusion process. But we can approximate it by clustering the significantly overlapped ARs and treating them as one, so as to reduce the dependency. By doing this, we approximate the estimated distribution of target location $\hat{P}(\mathbf{R}_t)$ by

$$\hat{P}(\mathbf{R}_t) \approx \sum_{i}^{\hat{N}} P(\mathbf{R}_t|\mathbf{r}_i) P(\mathbf{r}_i), \tag{5.2}$$

where $\hat{N}$ is the number of AR clusters, and $P(\mathbf{r}_i)$ represents the prior distribution of $\mathbf{r}_i$ in I_t which is regarded as uniform. The mode of the $\hat{P}(\mathbf{R}_t)$ determines the tracking result of $\mathbf{R}_t$. This is a voting process, as shown in Fig. 5.4.

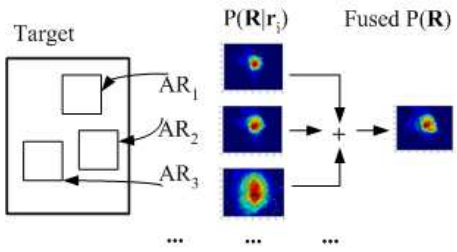


Figure 5.4. Estimation of target location.

It can be proved that this approximation only holds when $\hat{N}$ is large, because in this case the matching likelihoods of the ARs tend to dominate while the spatial correlations tend to be less critical. But this approximation is questionable when $\hat{N}$ is actually small. This is the limitation of our current implementation, as it is not quite suitable for tracking very small targets when only very few ARs are available and are largely correlated. Study of partially correlated information fusion is out of the scope of the thesis.

5.1.2.4. Late attentional selection. As described in previous sections, attentional selection is indispensable to the human perception of visual dynamics. For long duration tracking, the human visual tracking system is able to adapt to changing environments and to discriminate the small differences of the target from the distractions. Tremendous psychological evidence [72] indicates that visual tracking involves both early selection and late selection. Late selection may be a series of focused attention processes that are more proactive and involve higher level processing. For instance, the camouflage objects in the background around the target may have similar appearances, *e.g.*, people in a crowd as shown in Fig. 5.9. When tracking objects with non-convex shapes, it is inevitable to include some background regions in target initialization as shown in Fig. 5.11 and 5.12.

Some ARs may be more distinctive and have a large discriminative power, so that they should play a more important role in tracking. Thus, during the tracking, a subset of discriminative attentional regions (or D-ARs) are selected through ranking their abilities of discerning target motion from the background motion. We select the subset of D-ARs based on the Principle of Minimum Cross-Entropy (MCE) [17], also called Maximum Discrimination Information (MDI). This is tantamount to measuring discrimination information between the case of using $P(\mathbf{R}_t|\mathbf{r}_i)$ to approximate $\hat{P}(\mathbf{R}_t)$, and the case of using it to approximate the distribution of background motion:

$$KL(P(\mathbf{R}_t|\mathbf{r}_i)||\hat{P}(\mathbf{R}_t)) - KL(P(\mathbf{R}_t|\mathbf{r}_i)||P(B)), \tag{5.3}$$

where $P(B)$ is the distribution of nearby background motion. Assume $P(B)$ to be uniform, this reduces to cross-entropy between $P(\mathbf{R}_t|\mathbf{r}_i)$ and $\hat{P}(\mathbf{R}_t)$:

$$\begin{aligned} H(\mathbf{r}_i, \mathbf{R}_t) &= H(P(\mathbf{R}_t|\mathbf{r}_i), \hat{P}(\mathbf{R}_t)) \\ &= H(P(\mathbf{R}_t|\mathbf{r}_i)) + KL(P(\mathbf{R}_t|\mathbf{r}_i)||\hat{P}(\mathbf{R}_t)) \\ &= E_{P(\mathbf{R}_t|\mathbf{r}_i)}(-\log(\hat{P}(\mathbf{R}_t))), \end{aligned} \tag{5.4}$$

where $H(\cdot,\cdot)$ stands for the cross-entropy of two distributions and $H(\cdot)$ is the entropy.When occlusion happens, for those $S_{\mathbf{r}_i} = \emptyset$, the cross-entropy is set to ∞.

For each AR, the cross-entropy in a sliding temporal window of $\Delta t = 10$ frames is averaged with forgetting factor $\beta = 0.95$. The average cross-entropy $\tilde{H}(\mathbf{r}_i, \mathbf{R}_t)$ of all ARs are sorted to rank

their discriminative abilities:

$$\tilde{H}(\mathbf{r}_i, \mathbf{R}_t) = \sum_{j=0}^{\Delta t} \beta^j H(P(\mathbf{R}_{t-j}|\mathbf{r}_i), \hat{P}(\mathbf{R}_{t-j})). \tag{5.5}$$

The top-ranked ARs are identified as D-ARs and have larger weights in fusion. In our implementation, we choose the top 75%. They will be used to estimate $\hat{P}(\mathbf{R}_{t+1})$ in the next frame. The D-ARs are not fixed but dynamically changing with respect to the changes of the environment. Fig. 5.5 shows the top 10 D-ARs (as red rectangles) for two sequences at 3 different frames.

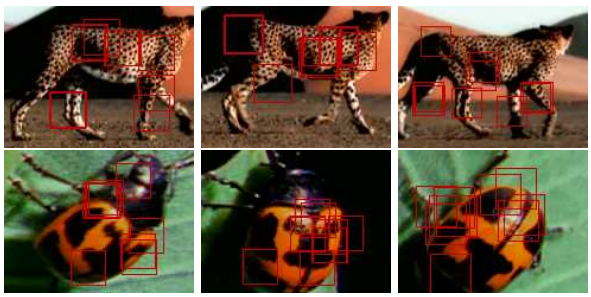


Figure 5.5. Examples of late selection of discriminative ARs.

5.1.2.5. Complexity analysis. In the SS-AVT algorithm, the computation costs for integral histogram calculation, fusion of $P(\mathbf{R}_t|\mathbf{r}_i)$, mode seek of $\hat{P}(\mathbf{R}_t)$ are constant and relatively inexpensive. The most computationally intensive module is attentional region matching. Exhaustive matching will involve $O(MN)$ times of D-dimensional vector comparison which is the basic computational unit in our analysis.

When the data set is hashed by LSH with L hashing functions, consider both indexing and query costs, the complexity is $O(ML + NL)$, where one hashing function is a D dimensional inner product calculation [19]. Therefore, the complexity ratio is approximately

$$\tau \approx \frac{O(ML + NL)}{O(MN)} \approx \frac{ML + NL}{MN}. \tag{5.6}$$

In the tracking scenario, the number of entries M in candidate set is much larger than the number of ARs N. Usually, M is several thousands and N is less than a hundred. Then, if we choose to hash the candidate set, L could be larger than N which means no speedup since we need to do indexing for every frame. So we hash AR pool with N elements, the complexity ratio $\tau \approx (L/N + L/M) \approx L/N$. Suppose there are $N = 100$ ARs, empirically $L = 20$ hashing functions are sufficient for querying the near neighbors within $d_t = 0.1$ at 0.9 probability. The computation reduces to approximately $\tau = 1/5$, if $N = 36$ and $L = 10$, $\tau = 0.28$. With this efficient matching, we can search a larger portion of the motion parameter space, *e.g.*, in our implementation, $[-20, +20]$ for (u, v) respectively and 3 scales ranging from 0.95, 1.0, and 1.05. For large targets, we down-sample the candidate region set to ensure $M \leq 3000$. The algorithm is implemented in C++ and tested on a Pentium-IV 3GHz PC. With moderate code optimization, the program runs at $10 - 15$ fps on average for 352×240 image sequences.

5.2. Experiments of Spatial Selection

5.2.1. Settings

We test the proposed SS-AVT algorithm for a variety of challenging real-world sequences including 3 primary types: quick motion with occlusion, camouflage environments, and objects with complex shapes. Note that in these tests, there are also scale and lighting changes. The targets include pedestrian, people in crowd, wild animals, bicycle and boat *etc.* The SS-AVT tracker is compared with the Mean-shift tracker [15] in the same enhanced YCbCr space with 1040 bins (32×32 for Cb and Cr and 16 bins for Y when the pixel is too dark or too bright). Since histograms of many rectangular regions need to be calculated, integral histogram technique [74] is a good implementation method to save some computations. The histograms with a large number of bins could well delineate the feature distributions, but also induce memory and computational

costs. For instance, 1040 additional images need to be stored for integral histogram calculation, this memory consumption is too large to afford. Thus, at the first frame we sort the 1040 bins and keep the top 128 bins and employ the 128 dimensional vector to represent one attentional region. This also saves considerable amount of computations in attentional region matching with LSH. Most of the video clips are downloaded from *Google Video*.

5.2.2. Quantitative comparison

For the quantitative comparison, the evaluation criteria of tracking error are based on the relative position error between the center of the tracking result and that of the ground truth, and the relative scale normalized by the ground truth scale. A perfect tracking expects the position differences to be around 0 and the relative scales close to 1.

We manually labeled the ground truth of the sequence `Walking` for 650 frames. The walking person, as shown in Fig. 5.7, is subjected to irregular severe occlusion when passing behind the bush. As indicated in quantitative comparison in Fig. 5.6, SS-AVT performs extremely well, but mean-shift loses track at frame 164 and never recovers.

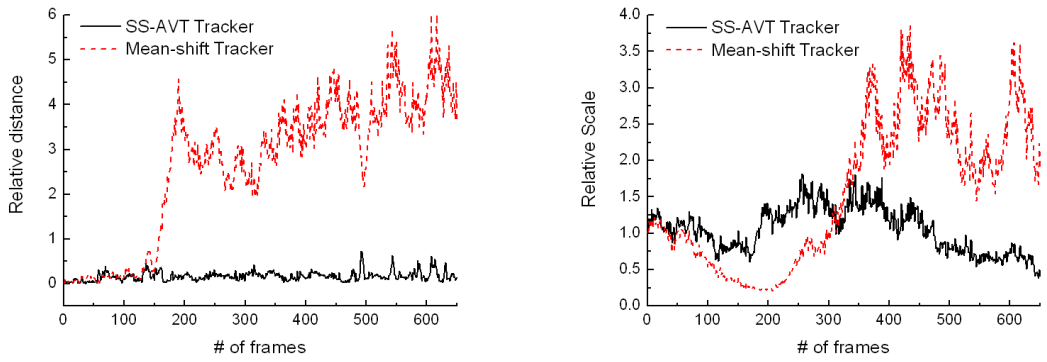


Figure 5.6. Quantitative comparison of relative position error and relative scale for tracking results of sequence [`Walking`].

5.2.3. More tracking results

As shown in Fig. 5.8, sequence `Horse Ride` involves very quick motion with occasional severe occlusions. The top row shows SS-AVT tracking results where the first frame displays the attentional region pool. The second row shows Mean-shift tracker's results. For SS-AVT tracker, the target is displayed as red dash rectangle, and the pixels covered by more than one D-AR are highlighted by increasing the luminance and the D-AR regions are surrounded by solid red lines. When there are too few matches for ARs, occlusion is detected and displayed with a white dash bounding box. Mean-shift tracker drifts after a serious occlusion is present at frame 54, while SS-AVT tracker is able to keep the track by a few attentional regions.

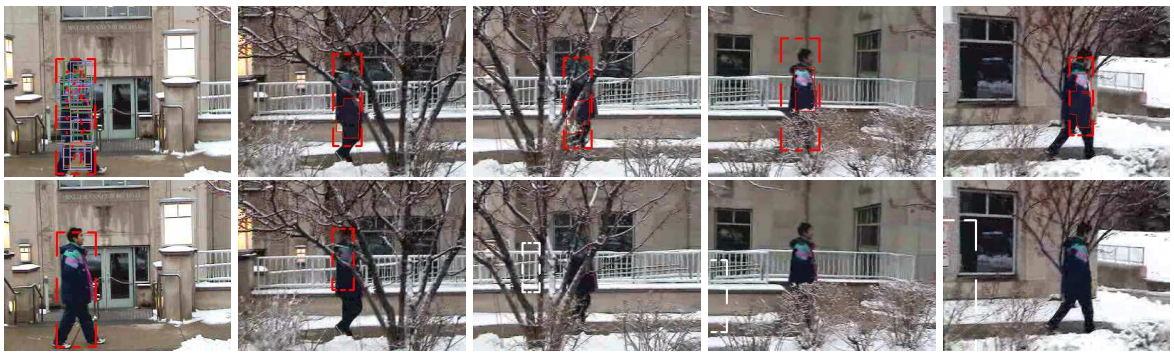


Figure 5.7. Tracking [`Walking`] for frame #1, 130, 164, 254 and 650, (1st row) SS-AVT tracker (N=55), and (2nd row) Mean-shift tracker.

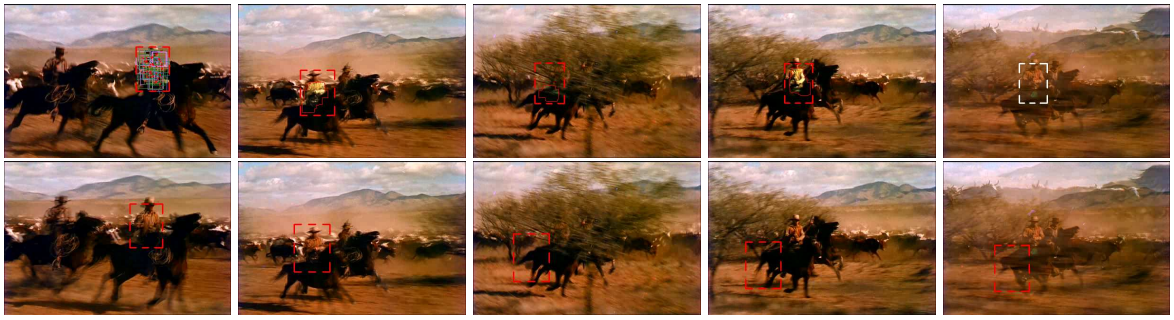


Figure 5.8. Tracking [`Horse Ride`] for frame #1, 40, 54, 58 and 60, (1st row) SS-AVT tracker (N=45), and (2nd row) Mean-shift tracker.

Camouflage environments, *i.e.*, similar or even identical objects around the target, are very challenging for tracking. We demonstrate SS-AVT's advantages by tracking one person in a crowd (Fig. 5.9), and a zebra with similar texture nearby (Fig. 5.10). The scale of Mean-shift tracker becomes unstable when nearby background presents similar color histograms, while SS-AVT is quite robust in camouflage environments due to the selection of D-ARs.

Figure 5.9. Tracking [`Marathon`] for frame #1, 33, 48, 75 and 84, (1st row) SS-AVT tracker (N=40), and (2nd row) Mean-shift tracker.

Figure 5.10. Tracking [`Zebra`] for frame #1, 63, 118, 136 and 160, (1st) SS-AVT tracker (N=57), and (2nd row) Mean-shift tracker.

Tracking objects with complex shapes is difficult in practice. Since it is not reasonable to require initialization to give the accurate boundary of the target, some background image regions will be inevitably included in the target. As illustrated in Fig. 5.11 and Fig. 5.12, the ground and some water are cropped in the targets. The ARs on the background are not correlated to the target's motion, thus they have high cross-entropy and are excluded from the D-AR subset. On the other

hand, Mean-shift tracker tries to match the holistic color histogram which is likely to be distracted by the background regions. More tracking results on a variety of general objects, such as animals, people and vehicles, are shown in Fig. 5.13, Fig. 5.14, and Fig. 5.15.

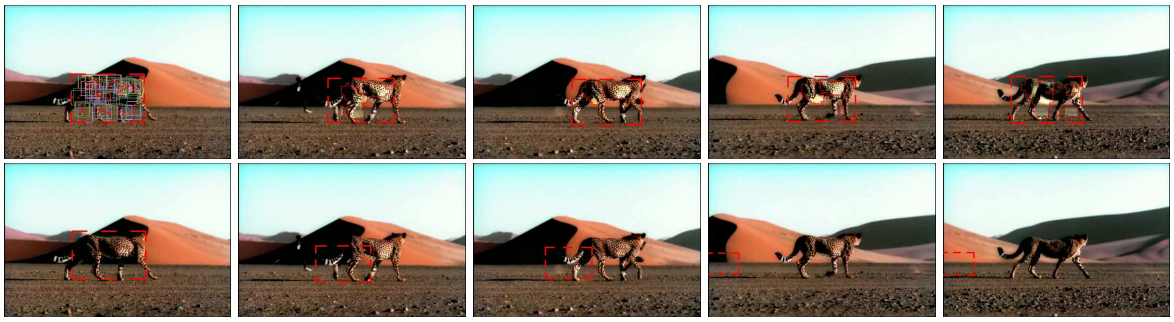


Figure 5.11. Tracking [Cheetah] for frame #1, 50, 80, 130, and 185, (1st) SS-AVT tracker (N=57), and (2nd row) Mean-shift tracker.

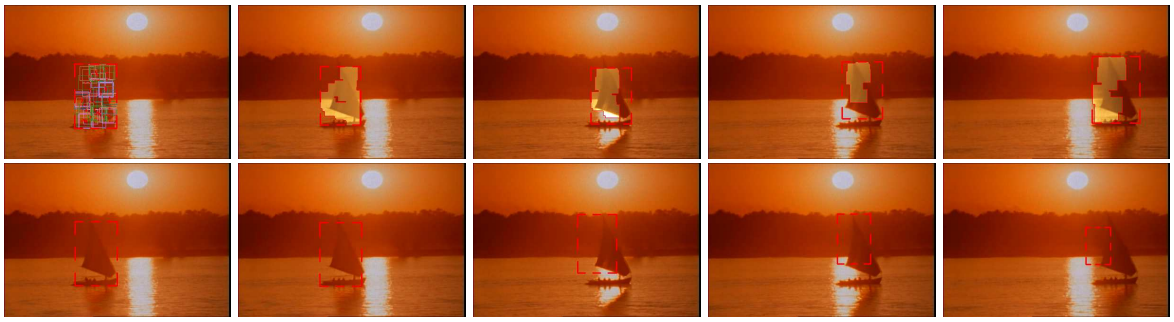
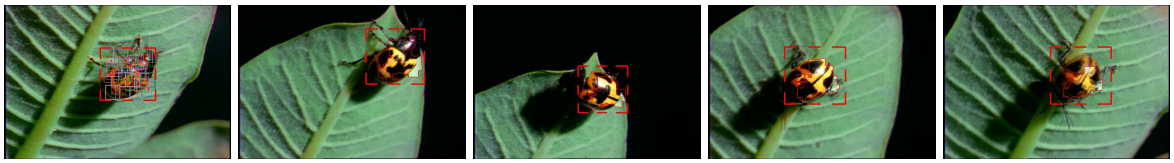
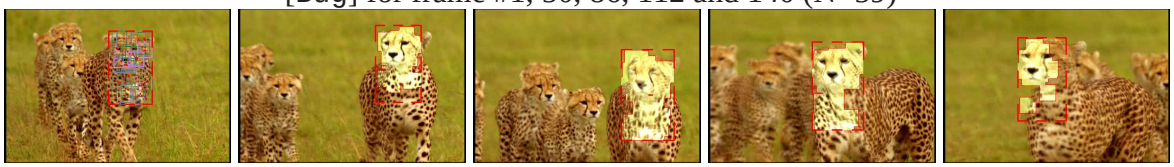


Figure 5.12. Tracking [Boat] for frame #1, 20, 60, 80 and 110 (1st), SS-AVT tracker (N=56), and (2nd row) Mean-shift tracker.

In this section, we have proposed a novel and promising tracking algorithm inspired by findings of human visual perception. It is suitable for tracking general objects without any prior knowledge. The target is represented by an attentional region pool which brings robustness against appearance variations. Dynamically spatial selection of discriminative attentional regions on the fly enables the tracker to handle camouflage environments and objects with complex shapes. In addition, by introducing LSH to on-line tracking, the proposed SS-AVT is computationally feasible. Our future work includes 3 aspects: 1) extending our current SS-AVT tracker to a general region tracking



[Bug] for frame #1, 50, 86, 112 and 140 (N=59)



[Cheetah2] for frame #1, 31, 68, 82 and 102 (N=65)

Figure 5.13. More results of spatially selective attentional visual tracking on animals.

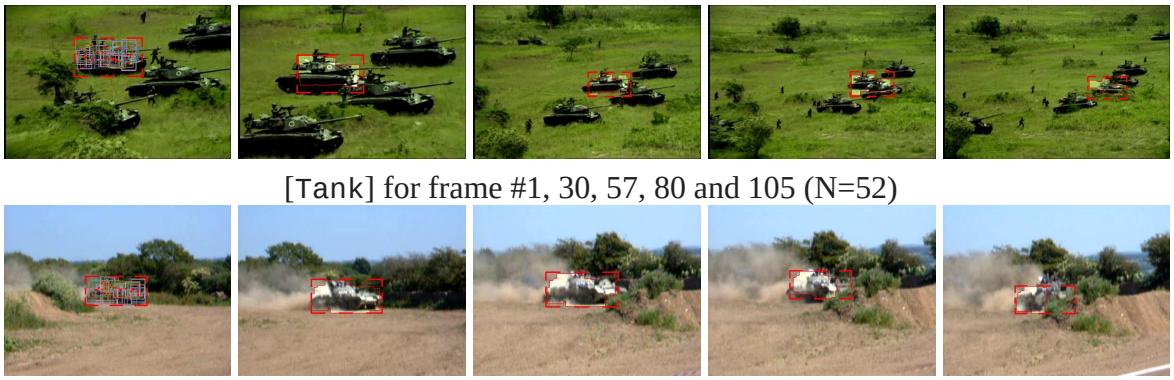
[Marathon2] for frame #1, 64, 90, 121 and 150 (N=21)

[NYC Bicycle] and #1, 57, 118, 146 and 180 (N=22)

[NYC Bicycle2] and #1, 174, 253, 371 and 460 (N=22)

Figure 5.14. More results of spatially selective attentional visual tracking on people.

tool by taking more motion parameters into consideration, 2) instantiating SS-AVT to particular objects by building extensive attentional region pool for different views, and 3) exploring property selection, *e.g.*, color, shape, and size, of attentional regions.



[Tank] for frame #1, 30, 57, 80 and 105 (N=52)

[Tank] for frame #1, 76, 180, 205 and 220 (N=44)

Figure 5.15. More results of spatially selective attentional visual tracking on vehicles.

5.3. Granularity and Elasticity Adaptation for Attentional Visual Tracking

The particular implementation of attentional visual tracking in Sec. 5.1 is generally robust to partial occlusions and camouflaged objects, but can be confronted by targets with large deformation or rotations, since it implicitly enforces strict relation geometrical relations among the attentional regions. Another difficult is how to select the initial scales of the attentional regions. Actually, in most tracking methods, matching is largely simplified and may only focus on certain characteristics of targets, for example, the existences of certain local visual patterns or coherence with certain overall feature statistics in appearance-based tracking. Consequently, successful tracking methods for certain types of targets may not adapt to other targets easily. Therefore, for generally applicable trackers, matching needs to be flexible for distinctive targets and adaptive with respect to target variations. In order to advance towards designing more general trackers, adaptation of more aspects of observation models need to be introduced and incorporated in a unified framework.

Specifically, for appearance-based tracking, there are two key aspects in designing observation models: what is the abstraction level of features, and how to take into account the geometrical structures of targets. For example, in two extreme cases, the template matching method [31] uses

local pixel intensities as features and employs sum of squared differences (SSD) as the matching criterion that enforces rigid geometrical relations among pixels, so it is suitable for small and rigid targets but vulnerable to partial occlusions and deformations. On the other hand, kernel-based tracking algorithms [**15**, **14**, **32**] represent targets by weighted histograms that delineate the overall statistics of targets' appearances and largely ignore their geometrical layouts. Therefore these algorithms can deal with non-rigid targets with sufficient sizes but are insensitive to some motion parameters. In between of these two extreme cases, many other algorithms, such as "super pixels" [**116**, **112**] or "bag-of-patches" approaches [**5**, **1**, **114**, **94**], extract features from some regions of interest on targets and consider their geometrical relations to different degrees.

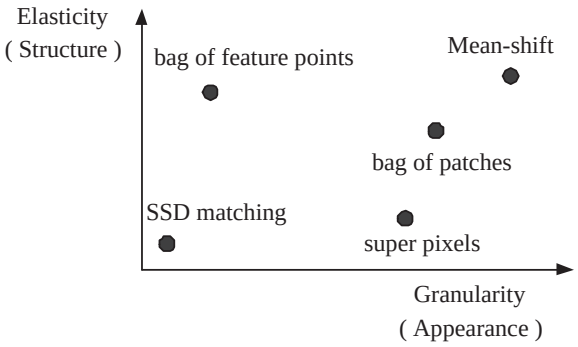


Figure 5.16. Illustration of different tracking approaches in terms of their relative granularity and elasticity.

We refer to the two aforementioned dimensions as the feature *granularity* and model *elasticity*. Granularity is a measure of descriptions of components that make up an object. We use the feature granularity to indicate the abstraction level of features, *e.g.* whether features describe attributes of a pixel, a blob region or a whole object. Elasticity refers to the degree of flexibility. Here we use the model elasticity to indicate the ability that the model tolerates geometrical changes among components, *e.g.* whether a model allows deformations inside targets or not. The feature granularity focuses on the target appearance and the model elasticity puts emphasis on its structure.

Some typical tracking approaches are illustrated qualitatively in Fig. 5.16 in terms of their relative feature granularity and model elasticity.

Humans also perceive different objects at different granularity levels [30]. For objects full of textures but without clear structures, human eyes may focus on their local appearance characteristics. For objects composed of several parts, both the appearances of the parts and their structures may attract attention. In addition, as the scales of objects change or deformation/partial occlusion occurs, the perception of target structure and local appearance may also change.

In this section, we propose another implementation of attentional visual tracking in which targets are represented by Markov random fields (MRF) of a set of attentional regions based on affine invariant features, where the feature granularity and model elasticity can be explicitly adapted with respect to targets' appearances during tracking. The feature vectors that delineate the local appearances of interest regions are extracted in a multi-scale manner. Thus, the scale ratio between the patch sizes that are used to extract feature vectors and the characteristic scales of interest regions specifies the feature granularity. On the other side, the geometrical relations among the interest regions, *i.e.* the structures of targets, are modelled in the pair-site potential functions whose parameters control the elasticity of the model. Thus, by updating the scale ratio and the parameters in the potential functions to maximize the joint likelihood of the MRF, the tracker adaptively balances the requirements of consistency with the local appearances and structures of targets. We refer this algorithm as granularity and elasticity adaptive attentional visual tracking (GE-AVT).

The main differences of GE-AVT from SS-AVT are as follows: 1) the early selection of attentional regions are based on affine invariant feature detection; 2) the attentional regions are organized as a MRF model rather than a set; 3) in late selection stage, certain properties of observation models are adjusted rather than spatial selection of ARs, *i.e.* the granularity of ARs and the elasticity of the MRF model. The target observation models can be viewed as a unification of

many previous tracking algorithms in the sense of how to organize appearance-based features. In addition, the adaptation of feature granularity and model elasticity in this paradigm exhibits a new way to update observation models to handle dynamical targets. The proposed method can estimate multiple motion parameters including translation, rotation and scaling, and handle partial occlusion, deformable targets and camouflaged objects within the unified framework as demonstrated by extensive experiments.

Local invariant features have been used in visual tracking before. The proposed method is different from some recent work [**114**, **94**, **89**] where the target is represented by a constellation of fixed-size (11×11) intensity patches extracted at Harris corners [**114**], or a bag of maximally stable extremal regions (MSER) [**68**], or an attributed relational graph of SIFT features [**89**] where the target model is adapted by eliminating and incorporating SIFT features and matched by graph matching.

5.3.1. Target observation model

In GE-AVT, we employ a unified tracking paradigm where the target is represented by an MRF model of attentional regions, and the feature granularity and the model elasticity can be explicitly modelled in a parametric way. In this section, we first introduce the general tracking paradigm and then describe the specific attentional regions based on affine invariant feature detection and MRF formulation in our implementation.

5.3.1.1. A unified tracking paradigm. Given the target initialization, we construct an MRF based on the attentional regions within the target. The hidden variables $\mathbf{X} = \{\mathbf{x}_i\}$ in the MRF are the parameters of the attentional regions on the target, and the observable variables are the parameters $\mathbf{Z} = \{\mathbf{z}_i\}$ of detected attentional regions based on affine invariant feature detection in

every frame. The adjacent attentional regions are linked in pair-wise cliques that encode their relative geometrical relations, as shown in Fig. 5.17. Then, by matching features extracted from the attentional regions in successive frames, the motion of the targets can be first coarsely estimated based on the motion of each AR. Afterwards, we refine the target's motion parameters by searching for the maximum a posteriori (MAP) estimate $P(\mathbf{X}^*|\mathbf{Z})$. We employ the scale ratio between the sizes of image patches to extract features and the characteristic scales of interest features to model the feature granularity. The elasticity of the model is controlled by the parameters in the potential functions. Assuming the tracking results are true realizations of the MRF, we adapt the granularity and elasticity to maximize the joint probability $P(\mathbf{X}^*)$. The entire paradigm is summarized in Fig. 5.18.

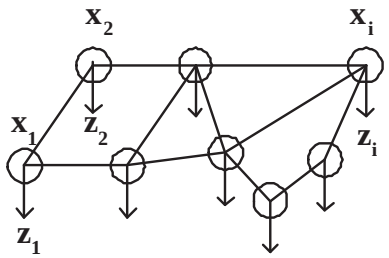


Figure 5.17. Illustration of the MRF model.

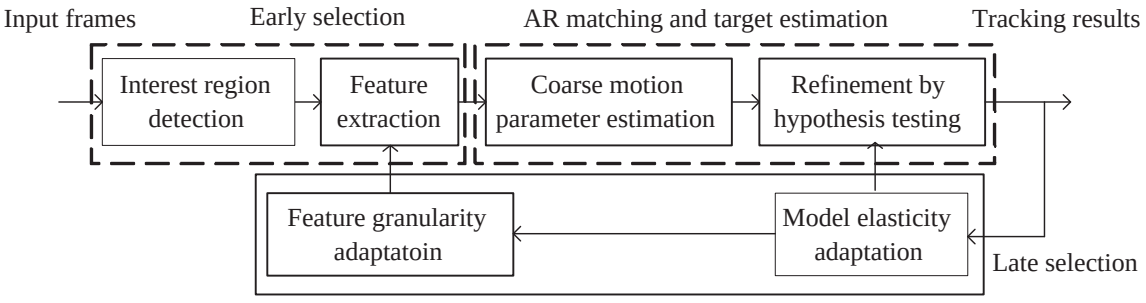


Figure 5.18. The proposed unified attentional tracking paradigm.

With different types of attentional regions and strategies in extracting features, the MRF-based observation model in this tracking paradigm can substantialize to different observation models. For example, if we regard each pixel as an attentional region and enforce strict geometrical relations among the pixels, this model degenerates to template tracking, or if the entire object is an attentional region and features are kernel-weighted histograms, then it turns to kernel-based tracking. Additionally, the paradigm can well explain the "bag-of-patches" method where no geometrical constraints are enforced in the MRF and the motions of targets are estimated from the confidence map or probabilistic occupance map generated from attentional region matching or outputs of classifiers.

5.3.1.2. Attentional region detection. For attentional regions in GE-AVT, salient image patches that are stable in affine transforms are preferable since their motion parameters can be explicitly estimated. There are many successful affine region detection methods [68], and we select Harris-Laplace interest regions mainly due to its computational efficiency and the ability to yield rich candidate regions.

The Harris-Laplace interest feature detector [35, 67] extracts points that are both local maxima of the Harris cornerness measure in spatial domain and maxima of the normalized Laplacian in scale space. The cornerness is measured based on the second moment matrix μ of the image gradient distribution in a neighborhood of a pixel $\{u, v\}$, as

$$\mu(\{u,v\}, s_I, s_D) = s_D^2 g(s_I) \otimes \begin{pmatrix} L_u^2(\{u,v\}, s_D) & L_u L_v(\{u,v\}, s_D) \\ L_u L_v(\{u,v\}, s_D) & L_v^2(\{u,v\}, s_D) \end{pmatrix} \tag{5.7}$$

where $L_u(\{u,v\}, s_D)$ and $L_v(\{u,v\}, s_D)$ are image gradients after smoothed by a Gaussian kernel with variance s_D, *a.k.a.* the derivation scale [67], and $g(s_I)$ indicates the Gaussian kernel to integrate the gradients whose variance s_I is referred as the integration scale or the characteristic

scale [67] of this point. The two eigenvalues $\lambda_1 \geq \lambda_2$ of μ characterize the pixel intensity distributions in the neighborhood. Two large eigenvalues imply the motion of the image patch surrounding this pixel may be phenomenal in all directions [84], thus it is a stable corner. Each Harris corner can be delineated by an ellipse region R centered at $\{u, v\}$ with the characteristic scale s_I and a shape matrix $\bar{\mu}$ that are normalized by the larger eigenvalue λ_1.

After extracting the ellipses $R = \{u, v, s_I, \bar{\mu}\}$ whose centers are Harris corners, we calculate the normalized Laplacian for those nested ellipses, that is, $R^{'}$ and R are nested if $R^{'} \subset R$. Note, the centers are not necessarily the same for the nested ellipses. The regions that are local maxima of the normalized Laplacian $s_D^2|L_{uu}(\{u, v\}, s_D) + L_{vv}(\{u, v\}, s_D)|$ are selected as the detected interest regions $\{R_1^t, \cdots, R_{M^t}^t\}$ where M^t denotes the number of regions detected at frame t.

Please refer to [67] for details about Harris-Laplace interest point detector. In [67], the location and shape of an interest region are iteratively refined in order to reflect the gradient distributions more accurately. As there is no guarantee of the convergence and the computation load is not affordable for tracking, we do not refine the interest regions.

5.3.1.3. MRF model formulation. Given the detected attentional regions $\{R_1^0, \cdots, R_{M^0}^0\}$ within the initial target at frame $t = 0$, we build the MRF including the hidden sites $\mathbf{x}_i = \{u_i, v_i, s_{Ii}, \bar{\mu}_i\}$ that correspond to R_i^0 and incorporate the target's motion parameters in the pair-wise potential functions.

The initial attentional regions $\{R_1^0, \cdots, R_{M^0}^0\}$ are regarded as a true realization of the MRF and denoted as $\{\mathbf{x}_1^0, \cdots, \mathbf{x}_{M^0}^0\}$. Then, the joint probability $P(\mathbf{X}) = P(\mathbf{x}_1, \cdots \mathbf{x}_{M_0})$ is expressed by the Gibbs energy defined over pair-wise clique set C, as

$$P(\mathbf{X}) = \frac{1}{Z} \exp\left(- \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} V(\mathbf{x}_i, \mathbf{x}_j) \right), \tag{5.8}$$

where Z_p is the partition function and V is the pair-wise potential function. $(\mathbf{x}_i, \mathbf{x}_j)$ is a pair-wise clique if the corresponding attentional regions overlap. The higher order cliques and the dependencies among cliques with common attentional regions are ignored to enable the problem tractable.

It is open and flexible to define the potential function V to model the relative geometrical relation between two attentional regions. To allow rotation and scaling of targets, in V we only involve the difference of the angle θ_{ij}^t between $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ at frame t against the reference angle θ_{ij}^0 between $\mathbf{x}_i^0$ and $\mathbf{x}_j^0$, and the target's current rotation angle $\Delta\theta^t$, as

$$V(\mathbf{x}_i^t, \mathbf{x}_j^t) = \frac{(\theta_{ij}^t - \theta_{ij}^0 - \Delta\theta^t)^2}{2\sigma^2}, \tag{5.9}$$

where σ is the assumptive variance of angle differences $\Delta\theta_{ij}^t = \theta_{ij}^t - \theta_{ij}^0$, which can control the elasticity of the MRF, *i.e.* how rigid the relative geometrical relations among attentional regions are enforced. The angle θ_{ij}^t between two adjacent attentional regions is calculated with the link connecting their centers, *i.e.* $\theta_{ij}^t = \arctan(\frac{v_i^t - v_j^t}{u_i^t - u_j^t})$. With these definitions, the partition function Z can be explicitly expressed as $Z_p = (\sqrt{2\pi}\sigma)^{|C|}$ where $|C|$ is the number of pair-wise cliques. An example of MRF model is illustrated in Fig. 5.19 where the attentional regions are drawn as yellow ellipses and the centers of those that are neighbors are linked with red lines.

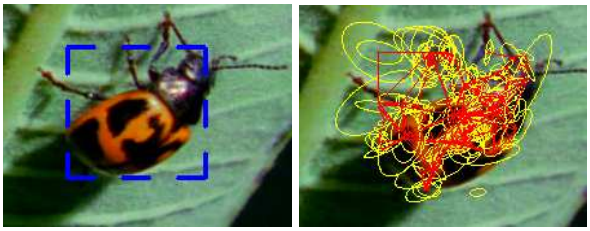


Figure 5.19. An example of the MRF model initialization.

Since histograms are generic and rotation invariant, for each attentional region, we extract a histogram of certain cues to describe its appearance. For a Harris-Laplace interest point, although the characteristic scale s_I is available, how large area around the point should be used to extract the features to insure good matching can not be determined before tracking. Thus, we utilize a scalar r to specify the scale ratio between the size of image patch used to extract the histogram and the characteristic scale s_I. For each $\mathbf{x}_i$, $H(r\mathbf{x}_i)$ represents the histogram extracted from the ellipse with the length of the major axis equal to rs_I. Therefore, the ratio r controls the feature granularity.

For an observation $\mathbf{z}_i^t$ of $\mathbf{x}_i$, we define the likelihood of individual attentional region based on the Bahattachaya coefficient ρ between the corresponding histograms, as

$$P(\mathbf{z}_i^t|\mathbf{x}_i) = \exp(1 - \rho(H(r\mathbf{x}_i^0), H(r\mathbf{z}_i^t))). \tag{5.10}$$

Fixed r may not be appropriate for all tracking scenarios, so r need to be adjusted during tracking.

5.3.2. Motion estimation

We estimate the motion parameters of the target with two steps. First, the attentional regions detected in current frame are matched with the initial regions in the MRF so as to coarsely estimate target's motion parameters, *i.e.* translation, scale and rotation angle, which mainly relies on the resemblance of appearance. Then, a few more motion parameters are sampled guided by the coarse estimates. The hypothesis that yields the highest joint posterior probability of the MRF is regarded as the tracking result, which takes both appearance and structures into consideration.

5.3.2.1. Coarse motion estimation. For every incoming frame, we perform Harris-Laplace interest points detector to locate the attentional regions $\{R_1^t, \cdots, R_{M^t}^t\}$ at current frame in an enlarged

region surrounding the previous tracking result. If one attentional region is matched to an initial attentional region $\mathbf{x}_i^0$, we regard it as an observation of the hidden site $\mathbf{x}_i$ and denote it by $\mathbf{z}_i^t$. The matching can be achieved by a classifier [5, 28], instead, we directly threshold the Bhattacharya coefficient ρ with the scale ratio r by a threshold T, as

$$\rho(H(r\mathbf{x}_i^0), H(r\mathbf{z}_i^t)) > T. \tag{5.11}$$

This matching is not necessarily a one-to-one mapping.

Incremental estimation of the motion parameters of targets, especially for the rotation angle, is not reliable since the estimation error could be accumulated. Thus, we estimate the target motion $\Delta u^t, \Delta v^t, \Delta s^t, \Delta\theta^t$ with respect to the target initialization. These motion parameters are first coarsely estimated by $\Delta u_i^t, \Delta v_i^t, \Delta s_{ij}^t, \Delta\theta_{ij}^t$ of individual observations $\mathbf{y}_i^t$ and each pair of $\mathbf{z}_i^t$ and $\mathbf{z}_j^t$ within a clique.

The translations $\Delta u_i^t = (u_i^t - u_i^0)$ and $\Delta v_i^t = (v_i^t - v_i^0)$ are cast in a 2D histogram. The scale factor and the rotation angle are estimated through 1D histogram of those of the detected pair-wise cliques $(\mathbf{y}_i^t, \mathbf{y}_j^t)$,

$$\Delta s_{ij}^t = \frac{|\{u_i^t, v_i^t\} - \{u_j^t, v_j^t\}|}{|\{u_i^0, v_i^0\} - \{u_j^0, v_j^0\}|}, \tag{5.12}$$

$$\Delta\theta_{ij}^t = \theta_{ij}^t - \theta_{ij}^0. \tag{5.13}$$

The modes of the distributions of these motion parameters present coarse motion estimation for the target, *i.e.* $\bar{\Delta u^t}, \bar{\Delta v^t}, \bar{\Delta s^t}, \bar{\Delta\theta^t}$. The histograms of attentional regions' motion parameters are similar to the confidence map or occupance map used in "bag-of-patches" approaches and the geometrical relations among the attentional regions have not been taken into account. Then, we employ these rough estimates to guide fine sampling of target motions and evaluate the posteriors to refine the motion estimation.

5.3.2.2. Motion parameter refinement. As the interest region detection and matching may contain errors, we further refine the coarse motion estimates $\bar{\Delta u}^t, \bar{\Delta v}^t, \bar{\Delta s}^t, \bar{\Delta\theta}^t$ by sampling a few more motion parameters around them and evaluating these hypotheses.

Given the observations $\mathbf{Z} = \{\mathbf{z_1}, \cdots, \mathbf{z_M}\}$ within a hypothesis target region, the MAP estimation $\mathbf{X}^* = \operatorname{argmax} P(\mathbf{X}|\mathbf{Z})$ presents the upper bound of the posterior of these observations $\mathbf{Z}$. With the Markovian properties and the field model structure $P(\mathbf{z}_i|\mathbf{X}) = P(\mathbf{z}_i|\mathbf{x}_i)$, the joint posterior can be expressed as

$$\begin{aligned} P(\mathbf{X}|\mathbf{Z}) &\propto P(\mathbf{Z}|\mathbf{X})P(\mathbf{X}) \\ &= P(\mathbf{X})\prod_i P(\mathbf{z}_i|\mathbf{x}_i). \end{aligned} \tag{5.14}$$

The joint probability $P(\mathbf{X})$ is calculated with Eq. 5.8 and Eq. 5.9 that utilize the hypothesis $\bar{\Delta\theta}^t$ as a parameter. The likelihood of individual interest region $P(\mathbf{z}_i|\mathbf{x}_i)$ is defined in Eq. 5.10. Then, the hypothesis whose optimal labelling $\mathbf{X}^*$ yields the highest posterior $P(\mathbf{X}^*|\mathbf{Z})$ is regarded as the tracking result.

5.3.3. Granularity and elasticity adaptation

In calculating $P(\mathbf{z}_i|\mathbf{x}_i)$ with Eq. 5.10 and $P(\mathbf{X})$ with Eq. 5.8 and 5.9, the scale ratio r to control the feature granularity and σ to control the elasticity of the MRF play important role in attentional region matching and MAP estimation. Pre-defined fixed r and σ are not likely to assure good matching for different targets and challenging situations such as partial occlusions and camouflage objects nearby. Thus, we adapt them in every frame to maximize the posteriors of tracking results. The updated parameters r^t and σ^t at frame t are used in motion estimation at next frame $t+1$.

5.3.3.1. Feature granularity adaptation. We update the scale ratio by searching $r^t + \Delta r$ until local maximum of $P(\mathbf{Z^t}|\mathbf{X^{t^*}}) = \prod_i P(\mathbf{z}_i^t|\mathbf{x}^{t^*}_i)$. Note, here locations and shapes of $\mathbf{z}_i^t$ and $\mathbf{x}^{t^*}_i$ are given, only r^t affects $P(\mathbf{Z^t}|\mathbf{X^{t^*}})$. This is equivalent to maximize the sum of the Bhattacharya coefficients of all observed attentional regions $\mathbf{z}_i^t$ in the tracked target, as

$$r^{t^*} = \text{argmax} \sum_i \rho(H(r^t \mathbf{x}_i^0), H(r^t \mathbf{z}_i^t))). \tag{5.15}$$

The histograms $H(r^t \mathbf{x}_i^0)$ are pre-calculated and stored at tracking initialization. To reduce the computation overhead of adaptation, we perform local gradient search around $r^t \pm \Delta r$ with $r^0 = 2$ and $\Delta r = 0.1$ in our experiments. Thus, the feature granularity is updated according to the appearance changes. If the target is rigid and stable, good matching can be obtained with large ratio r. If partial occlusion or deformation happen, small r may be appropriate.

5.3.3.2. Model elasticity adaptation. The parameter σ in the pair-site potential functions controls the elasticity of the MRF. To enable σ match the degree of deformation of the target, we solve it by maximize the likelihood of the current tracking result, as

$$\frac{\partial \ln P(\mathbf{X}^{t^*}|\sigma)}{\partial \sigma} = 0. \tag{5.16}$$

Plug in the partition function Z_p and the potential energy in Eq. 5.9 to $P(\mathbf{X}_t^*|\sigma)$, we have

$$\begin{aligned} P(\mathbf{X}_t^*|\sigma) &= \frac{1}{(\sqrt{2\pi}\sigma)^{|C|}} \exp\left(- \sum_{(\mathbf{x}_i^{t*}, \mathbf{x}_j^{t*}) \in C} V(\mathbf{x}_i^{t*}, \mathbf{x}_j^{t*}) \right) \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^{|C|}} \exp\left(- \sum_{(\mathbf{x}_i^{t*}, \mathbf{x}_j^{t*}) \in C} \frac{(\Delta\theta_{ij}^{t*} - \Delta\theta^{t^*})^2}{2\sigma^2} \right). \end{aligned}$$

Solving Eq. 5.16, we obtain the assumptive variance σ^t of angle differences given the current tracking result,

$$\sigma^t = \frac{1}{|C|} \sum_{(\mathbf{x}^{t*}_i, \mathbf{x}^{t*}_j) \in C} (\Delta\theta^{t*}_{ij} - \Delta\theta^{t*})^2. \tag{5.17}$$

The optimal σ^t is the variance of the observed angle differences. So, if the relative geometrical relations of the detected interest regions are stable, σ^t is small, on the other hand, σ^t increases when deformations occur.

5.4. Experiments of Granularity and Elasticity Adaptation

We evaluate the proposed GE-AVT for a variety of real-world sequences that present deformations, partial occlusions, and camouflage objects. In the Harris-Laplace interest point detector, up to 12 different integration scales are tested depending on the size of the target. The features used to match the attentional regions are 2D histograms in normalized-RG space with 24×24 bins and the corresponding matching threshold for the Bhattacharya coefficient is set to $T = 0.75$. The proposed tracker is implemented with C++ which runs at 2-10 frame per second on a Pentium-IV 3GHz desktop. The computation load is jointly determined by the number of scales in the interest feature detector and the number of attentional regions detected.

To exhibit the generality of the proposed method, for different sequences, we compare the performance with three trackers: a Mean-shift tracker that also employs 2D histograms in normalized-RG space with 24×24 bins, a template tracker where the image regions are normalized to grey-level patches and compared with SSD, and a "bag-of-patches" tracker using the same set of interest regions but ignoring their geometrical relations. Although these 3 trackers can deal with different kinds of tracking scenarios, we demonstrate the proposed method can overcome some difficulties to them within the unified tracking paradigm.

5.4.1. Illustration of tracking results

The tracking results are displayed in three rows in Fig. 5.20. At the first row, the initialization of the MRF model is shown in the first image where the cliques are drawn with red lines, and followed by the interest region detection results where the matched regions are drawn as yellow ellipses while the non-matched ones are light blue ellipses. Note the length of the major axis in drawing is the product of the scale ratio r^t and the interest region's characteristic scale s_I. Our tracking results are illustrated at the second row where the target is indicated by a blue dash bounding box and the pixels covered by matched attentional regions are highlighted with red boundaries. The comparison tracking results are shown at the third row.

In sequence [`Sidewalk`], the size of target is small which is suitable for the template tracker. However, when a bicycler is passing by the pedestrian from frame 140, the template tracker is easily distracted, shown in the third row of Fig. 5.20. In our tracker, as the attentional regions on the upper body of the pedestrian remain stable, the tracker can get along with the distractions.

Figure 5.20. Tracking [`Sidewalk`] for frame #1, 140, 145, 152 and 163, (1st row) initialization and interest region detection, (2nd row) the proposed tracker (3nd) the template tracker.

5.4.2. Partial occlusions

The sequence [Face] first used in [1] presents different degrees of partial occlusions. Large scale ratio r may jeopardize the interest region matching when partial occlusions occur. From Fig. 5.21 we can observe that in our method the scale ratio r^t is adapted to follow the changing of degree of occlusion. r^t decreases to about 1.2 at frame 285 and increases to 3 when the book moves away. For the mean-shift tracker, when partial occlusion happens, the scale estimation is no longer reliable and can hardly recover. Some representative frames are shown in Fig. 5.22.

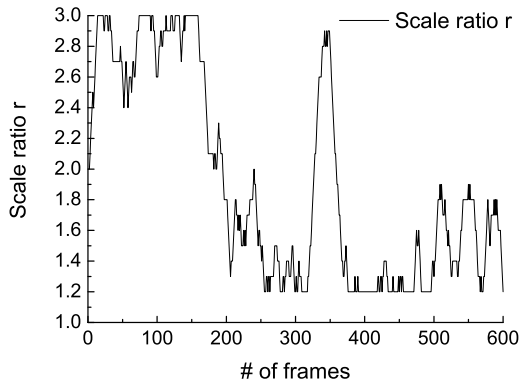


Figure 5.21. Scale ratio r^t for sequence [Face].

5.4.3. Deformable objects

For sequence [Cock fight], when the target cock experiences large deformation around frame 240, σ^t in the potential functions increases considerably, as shown in Fig. 5.23. This means the structure or the relative geometrical relations among the interest regions are largely ignored. Thus, the target is located mainly by matching its appearance. When the cock pauses fighting at frame 250, its structure helps the proposed tracker to locate the target and estimate the scale more accurately than the Mean-shift tracker.

Figure 5.22. Tracking [Face] for frame #1, 285, 345, 585 and 599, (1st row) initialization and interest region detection, (2nd row) the proposed tracker (3nd) the Mean-shift tracker.

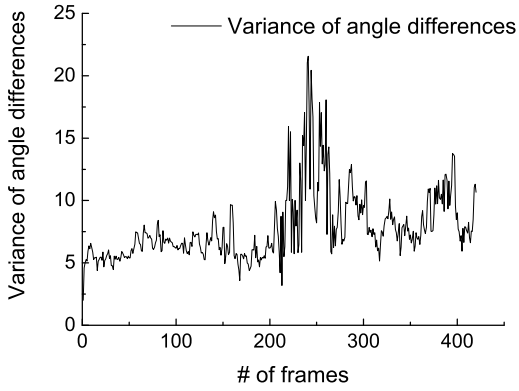


Figure 5.23. σ^t for sequence [Cock fight].

5.4.4. Camouflaged objects

If the appearance of the target is distinctive in the scene, “bag-of-patches” approaches may work well, however, they are usually vulnerable when camouflage, *i.e.* similar or even identical objects, presents close to the target. As shown in Fig. 5.25, when the camouflage package moves close

Figure 5.24. Tracking [Cock fight] for frame #1, 229, 241, 250 and 410, (1st row) initialization and interest region detection, (2nd row) the proposed tracker (3nd) the Mean-shift tracker.

to the target from frame 640, the scale estimation in the pure "bag-of-patches" tracker becomes unstable and it gradually drifts to the wrong target. In our approach, though interest regions detected on the camouflage package have similar appearances, they are excluded since their relative positions are not consistent with the MRF model.

5.4.5. More tracking results

More test results on sequences with in-plane rotation and scale changes are shown in Fig. 5.26 and Fig. 5.27.

In this section, we have introduced a new perspective on adapting target observation models in terms of the feature granularity and model elasticity in a unified tracking paradigm, where targets are represented by MRFs of attentional regions. By employing a multi-scale scheme to extract features from attentional regions and adjusting the parameters that regulate the target geometrical layout, the proposed method automatically tunes the observation model's focus on a target's

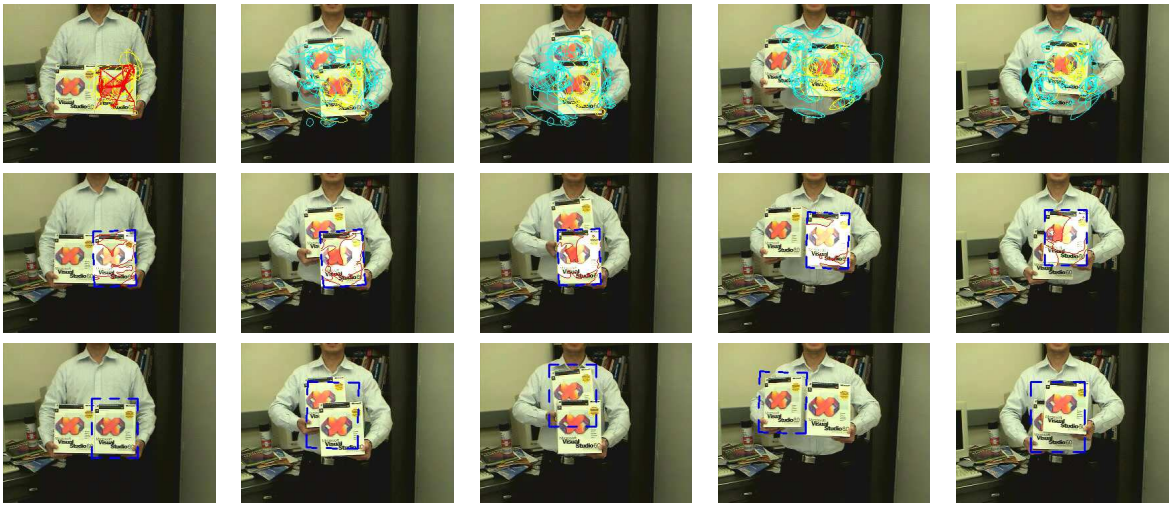


Figure 5.25. Tracking [Package] for frame #1, 640, 702, 740 and 792, (1st row) initialization and interest region detection, (2nd row) the proposed tracker (3nd) the "bag-of-patches" tracker.

Figure 5.26. Tracking [Box] for frame #1, 215, 405, 510 and 598, (1st row) initialization and interest region detection, (2nd row) the proposed tracker.

appearance and structure. Future work will include investigation about how to adapt the feature granularity of individual attentional regions and the potential functions for each clique.

5.5. Discussion on Attentional Tracking

The core strategy of attentional tracking is to construct a highly redundant target representation at initialization, thus during tracking the tracker can adjust its attention to more discriminative parts or properties of the target. This is fundamentally different from directly updating the observation

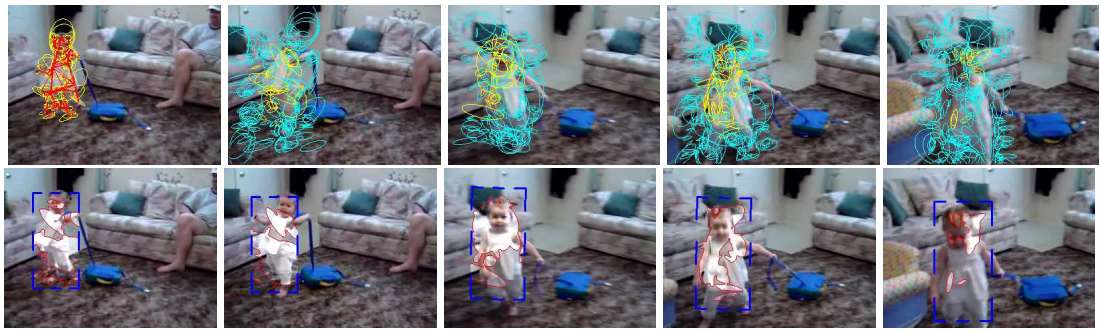


Figure 5.27. Tracking [Kid] for frame #1, 10, 40, 45 and 60, (1st row) initialization and interest region detection, (2nd row) the proposed tracker.

model with the tracking results or on-line learning since no new features are introduced to the observation models after initialization but more agile matching methods are employed. There are several issues open to different implementations: how to select attention regions at early selection to represent the target, how to organize the attentional regions, and how to adapt the focus of matching criteria of the trackers at late selection, as well as the computation costs.

In our implementations, SS-AVT extracts a pool of attentional regions whose motions can be estimated reliably to represent the targets. Since all attentional regions have the same size (30×30 at initialization), we can utilize LSH technique to accelerate the matching. The relative geometrical relations between attentional regions are implicitly assumed to be stationary. The pros are that the target representation is quite robust to partial occlusions or camouflage objects and efficient, but the cons are that the sizes of the attentional regions are not selected in a principled way so it is hard to handle large scale changes especially when the targets zoom out and the sizes get smaller, and the strict requirement of the geometrical structure makes it hard to deal with objects with deformation or rotations.

On the other hand, GE-AVT extracts attentional regions using a multiple scale scheme based on affine invariant feature point detection, thus the characteristic scales and the rotation angles of the attentional regions can be estimated, which enables the tracker to infer more motion parameters.

In addition, the attentional regions are organized in a MRF model so the matching scheme can be more flexible. But the main drawback is that the affine invariant point detection is sensitive to reflections, illumination conditions and view changes, therefore a large portion of attentional regions may not yield good matching. More robust attentional region extraction methods and more flexible matching scheme deserve further investigation to push the attentional tracking algorithm to be more robust and practical.

CHAPTER 6

Game-Theoretic Multiple Target Tracking

Multiple target tracking (MTT) is a challenging task when similar targets are present in close vicinity. The challenge is rooted in the difficulty of estimating the motions of multiple targets cannot be treated independently if they are present in close vicinity. Especially, if their visual observations (or visual evidence) are mixed, it is generally very difficult to figure out the right associations of these observations to the individual targets (that implies a general segmentation problem). To handle this difficulty, the motions of multiple targets have to be jointly estimated from the mixed visual observations.

This joint estimation problem can be performed in a centralized fashion by formulating a joint observation model, as treated in many existing methods [**79**, **64**, **78**, **47**, **41**, **69**, **91**, **117**, **54**]. Because the joint observation model evaluates hypotheses of joint motion states, these methods lead to complicated centralized MTT trackers that generally need to search a rather high dimensional solution space.

This chapter brings a new view to MTT from a game-theoretic perspective, bridging the joint motion estimation and the Nash Equilibrium of a *game*. Instead of designing a centralized tracker, MTT is decentralized and a set of individual trackers is used, each of which tries to maximize its visual evidence for explaining its motion as well as generating interference to others. Modelling this competitive behavior, a special *game* is designed so that the difficult joint motion estimation is achieved at the Nash Equilibrium of this game where no individual tracker has incentives to change its motion estimate. We substantialize this novel idea in a solid case study where individual

trackers are kernel-based trackers. An efficient best response updating procedure is designed to find the Nash Equilibrium. The power of this game-theoretic MTT is shown by promising results on difficult real videos.

6.1. Interference Model for Kernel-based Trackers

In this section, we introduce a new analytical interference model for kernel-based trackers, which is a key component in formulating the game-theoretic MTT. This interference model takes both target appearances and spatial relations into consideration.

6.1.1. Joint likelihood maximization

Denote the motion parameters for the ith target by θ_i. Its corresponding support is denoted by Ω_i, *i.e.* the set of pixels $\{\mathbf{x}_n\}$ within the region of target i. Thus, the motions of a number of N targets can be estimated by maximizing the joint likelihood,

$$\Theta^* = \underset{\{\theta_1,\cdots,\theta_N\}}{\operatorname{argmax}} P(\bigcup_{i=1}^{N} \Omega_i|\theta_1,\cdots,\theta_N). \tag{6.1}$$

If no occlusion is present, *i.e.* $\Omega_i \cap \Omega_j = \emptyset, \forall i,j \leq N$. This joint optimization can be done independently:

$$\theta_i^* = \underset{\theta_i}{\operatorname{argmax}} P(\Omega_i|\theta_i), \quad \forall i \leq N. \tag{6.2}$$

If occlusion is present, *i.e.* $\Omega_i \cap \Omega_j \neq \emptyset, \exists i,j \leq N$, we can assign the pixels in the overlapped regions to different targets probabilistically, thus

$$\Theta^* = \underset{\{\theta_1,\cdots,\theta_N\}}{\operatorname{argmax}} \prod_{i=1}^{N} P(\hat{\Omega}_i|\theta_1,\cdots,\theta_N), \tag{6.3}$$

where $\hat{\Omega}_i$ is the probabilistic support of target i. This is equivalent to an energy minimization problem:

$$\Theta^* = \underset{\{\theta_1,\cdots,\theta_N\}}{\operatorname{argmin}} -\sum_{i=1}^{N} \ln P(\hat{\Omega}_i|\theta_1,\cdots,\theta_N). \tag{6.4}$$

6.1.2. Kernel-based likelihood

Specifically, for a kernel-based tracker, a target is represented by a kernel weighted feature histogram [15]. The motion parameters are denoted by $\theta \triangleq \{\mathbf{y}, h\}$, where $\mathbf{y}$ is the location of the kernel center and h is its scale. Denote by $\mathbf{x}_n$ the 2D pixel location and $z_n \triangleq ||\frac{\mathbf{x}_n - \mathbf{y}}{h}||$. The kernel function $k(z_n^2)$ used is the Epanechnikov kernel:

$$k(z_n^2) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-z_n^2), & z_n^2 < 1 \\ 0, & \text{otherwise} \end{cases}, \tag{6.5}$$

where $d = 2$ and c_d is the area of the unit circle. The negative derivative of the kernel is denoted by $g(z_n^2) \triangleq -k'(z_n^2)$.

Following the notations in [15], for a single tracker without interference, the model of target i is described by an M-bin histogram $\mathbf{q}_i = \{q_{im}\}_{m=1,\cdots,M}$, and the target hypothesis by $\mathbf{p}_i(\mathbf{y}_i) = \{p_{im}(\mathbf{y}_i)\}_{m=1,\cdots,M}$,

$$p_{im}(\mathbf{y}_i) = \sum_{\mathbf{x}_n \in \Omega_i} k(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)\delta[b(\mathbf{x}_n) - m], \tag{6.6}$$

where $\delta[\cdot]$ is the Kronecker delta function and the function $b(\cdot)$ maps the pixel location $\mathbf{x}_n$ to a bin index m. The Bhattacharyya coefficient $\rho(\mathbf{y}_i)$ is employed to measure the similarity between a target hypothesis and the model

$$\rho(\mathbf{y}_i) = \sum_{m=1}^{M} \sqrt{p_{im}(\mathbf{y}_i)q_{im}}. \tag{6.7}$$

Since the distance from the hypothesis histogram $\mathbf{p}_i(\mathbf{y}_i)$ to the model histogram $\mathbf{q}_i$ can be defined as $d(\mathbf{y}_i) = \sqrt{1-\rho(\mathbf{y}_i)}$, the likelihood model for tracker i (in Eq. 6.2) without considering interference can be formulated as:

$$P(\Omega_i|\theta_i) \propto e^{1-\rho(\mathbf{y}_i)}. \tag{6.8}$$

6.1.3. Kernel-based interference model

Due to partial occlusion, we need to consider the interference among the N targets, *i.e.* $\Omega_i \cap \Omega_j \neq \emptyset, \exists i,j \leq N$. The observation model for tracker i is no longer solely determined by $\mathbf{y}_i$ but the joint motion configuration of all trackers (which is denoted by $\{\mathbf{y}_i, \mathbf{y}_{-i}\} = \{\mathbf{y}_i, \cdots, \mathbf{y}_N\}$ to highlight other trackers' interference with tracker i). In view of this, we generalize the kernel-based histogram model by,

$$\hat{p}_{im}(\mathbf{y}_i, \mathbf{y}_{-i}) = \frac{1}{C_i} \sum_{\mathbf{x}_n \in \Omega_i} \left\{ k(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2) \delta[b(\mathbf{x}_n) - m] \cdot \frac{q_{im}(\mathbf{x}_n) k(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)}{\sum_{j=1}^{N} q_{jm}(\mathbf{x}_n) k(||\frac{\mathbf{x}_n - \mathbf{y}_j}{h_j}||^2)} \right\}, \tag{6.9}$$

where $C_i \leq 1$ is a normalization term. The probability that the pixel $\mathbf{x}_n$ is within Ω_i is approximated by

$$\frac{q_{im}(\mathbf{x}_n) k(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)}{\sum_{j=1}^{N} q_{jm}(\mathbf{x}_n) k(||\frac{\mathbf{x}_n - \mathbf{y}_j}{h_j}||^2)}, \tag{6.10}$$

where $q_{im}(\mathbf{x}_n) = \sum_{m=1}^{M} q_{im}[\delta(b(\mathbf{x}_n) - m)]$ is the histogram bin value for pixel $\mathbf{x}_n$ in the target model $\mathbf{q}_i$. Please note when using Epanechnikov kernel with a finite support, if one tracker has no overlap with others, Eq. 6.9 degenerates to Eq. 6.6. To avoid numerical problems, we set $q_{im} = \epsilon > 0, \forall m < M$, where ϵ is a very small value, to guarantee non-zero bins $q_{im}(\mathbf{x}_n)$ and $q_{jm}(\mathbf{x}_n)$.

The *generalized Bhattacharyya coefficient* is defined as $\hat{\rho}(\mathbf{y}_i, \mathbf{y}_{-i}) = \sum_{m=1}^{M} \sqrt{\hat{p}_{im}(\mathbf{y}_i, \mathbf{y}_{-i}) q_{im}}$. Then, the likelihood model for target i with interference is formulated as:

$$P(\hat{\Omega}_i | \theta_1, \cdots, \theta_N) \propto e^{1-\hat{\rho}(\mathbf{y}_i, \mathbf{y}_{-i})}. \tag{6.11}$$

This interference model takes both the appearance similarity and spatial relations into account. For examples, as shown in case A in Fig. 6.1, if the bin values of the pixels in the overlap region are larger in a target model j than in the other, then those pixels have higher weights in Eq. 6.10. On the other hand, if the pixels in the overlap region are equally likely for both target models as in the case B, then the pixels close to the center of one target shall have higher probability to be counted in its model. Furthermore, since Eq. 6.10 is less than 1, this interference model down-weights those pixels that are in the overlapped regions of different trackers and have ambiguous identities.

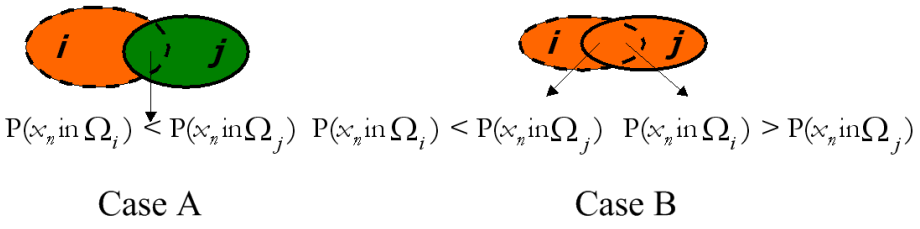


Figure 6.1. Illustration of two cases for the interference model.

6.2. Game-theoretic Multiple Target Tracking

Based on the interference model, we can formulate the joint motion estimation (Sec. 6.2.1) and construct a game (Sec. 6.2.2) whose N.E. corresponds to a local optimum of the joint motion estimation and can be efficiently solved (Sec. 6.2.3). The algorithm is summarized in Sec. 6.2.4.

6.2.1. Joint motion estimation

Assuming that the scales remain constant when multiple targets approach to each other, based on the interference likelihood model (Eq. 6.11), the minimization of the joint energy (in Eq. 6.4) is equivalent to:

$$\max_{\{\mathbf{y}_1,\cdots,\mathbf{y}_N\}} J_1(\mathbf{y}_1,\cdots,\mathbf{y}_N) \quad = \quad \sum_{i=1}^{N} \hat{\rho}_i(\mathbf{y}_i,\mathbf{y}_{-i}). \tag{6.12}$$

Maximizing the joint likelihood is equivalent to optimizing the joint kernel locations of all targets that maximize the sum of the generalized Bhattacharyya coefficients.

Denote the initial locations of the trackers by $\{\mathbf{y}_i^0,\mathbf{y}_{-i}^0\}$. Then, performing Taylor expansion *w.r.t.* $\hat{p}_{im}(\mathbf{y}_i^0,\mathbf{y}_{-i}^0)$ and plugging Eq. 6.9 into $\hat{\rho}_i(\mathbf{y}_i,\mathbf{y}_{-i})$, $\hat{\rho}_i(\mathbf{y}_i,\mathbf{y}_{-i})$ can be approximated by

$$\begin{aligned}
&\hat{\rho}_i(\mathbf{y}_i,\mathbf{y}_{-i}) = \sum_{m=1}^{M} \sqrt{\hat{p}_{im}(\mathbf{y}_i,\mathbf{y}_{-i}) q_{im}} \\
\approx\ & \frac{1}{2}\sum_{m=1}^{M}\left(\sqrt{\hat{p}_{im}(\mathbf{y}_i^0,\mathbf{y}_{-i}^0) q_{im}} + \hat{p}_{im}(\mathbf{y}_i,\mathbf{y}_{-i})\sqrt{\frac{q_{im}}{\hat{p}_{im}(\mathbf{y}_i^0,\mathbf{y}_{-i}^0)}}\right) \\
=\ & \frac{1}{2}\sum_{m=1}^{M}\sqrt{\hat{p}_{im}(\mathbf{y}_i^0,\mathbf{y}_{-i}^0) q_{im}} + \frac{1}{2C_i}\sum_{\Omega_i}\omega_i(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_i}{h_i}||^2)\frac{q_{im}(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_i}{h_i}||^2)}{\sum_{j=1}^{N} q_{jm}(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_j}{h_j}||^2)},
\end{aligned} \tag{6.13}$$

where $\omega_i(\mathbf{x}_n)$ is determined by the initial status of tracker i $\hat{p}_{im}(\mathbf{y}_i^0,\mathbf{y}_{-i}^0)$ and the model histogram $\mathbf{q}_i$ of target i,

$$\omega_i(\mathbf{x}_n) = \sum_{m=1}^{M}\delta[b(\mathbf{x}_n)-m]\sqrt{\frac{q_{im}}{\hat{p}_{im}(\mathbf{y}_i^0,\mathbf{y}_{-i}^0)}}. \tag{6.14}$$

Since only the second term in Eq. 6.13 is related to the variable $\{\mathbf{y}_i,\mathbf{y}_{-i}\}$ given the initial locations, we can ignore the terms in J_1 that are not affected by $\{\mathbf{y}_1,\cdots,\mathbf{y}_N\}$. Then we redefine

the objective function and have:

$$\max_{\{\mathbf{y}_1,\cdots,\mathbf{y}_N\}} J_2(\mathbf{y}_1,\cdots,\mathbf{y}_N) \triangleq \sum_{i=1}^{N} r_i(\mathbf{y}_i,\mathbf{y}_{-i}), \tag{6.15}$$

where $r_i(\mathbf{y}_i,\mathbf{y}_{-i})$ corresponds to the individual matching of tracker i (as the second term in Eq. 6.13):

$$r_i(\mathbf{y}_i,\mathbf{y}_{-i}) \triangleq \frac{1}{2C_i}\sum_{\Omega_i} \frac{\omega_i(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_i}{h_i}||^2)}{1+\sum_{j=1,j\neq i}^{N}\frac{q_{jm}(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_j}{h_j}||^2)}{q_{im}(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_i}{h_i}||^2)}}. \tag{6.16}$$

Since ∇J_2 *w.r.t.* to $\{\mathbf{y}_1,\cdots,\mathbf{y}_N\}$ is intractable, we further approximate it with a lower bound $J_3 \leq J_2$:

$$\max_{\{\mathbf{y}_1,\cdots,\mathbf{y}_N\}} J_3(\mathbf{y}_1,\cdots,\mathbf{y}_N) \triangleq \sum_{i=1}^{N} \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i}), \tag{6.17}$$

where

$$\tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i}) \triangleq \frac{1}{2C_i}\sum_{\Omega_i} \frac{\omega(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_i}{h_i}||^2)}{1+\sum_{j=1,j\neq i}^{N}\frac{q_{jm}(\mathbf{x}_n)}{q_{im}(\mathbf{x}_n)}k(||\frac{\mathbf{x}_n-\mathbf{y}_j}{h_j}||^2)}. \tag{6.18}$$

This proximation means that the pixels in the occlusion regions are further down-weighted as

$$1/\left(1+\sum_{j=1,j\neq i}^{N}\frac{q_{jm}(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_j}{h_j}||^2)}{q_{im}(\mathbf{x}_n)k(||\frac{\mathbf{x}_n-\mathbf{y}_i}{h_i}||^2)}\right) \rightarrow 1/\left(1+\sum_{j=1,j\neq i}^{N}\frac{q_{jm}(\mathbf{x}_n)}{q_{im}(\mathbf{x}_n)}k(||\frac{\mathbf{x}_n-\mathbf{y}_j}{h_j}||^2)\right). \tag{6.19}$$

This is reasonable, since we don't explicitly recover the occlusion relations among the targets and a natural choice is to reduce their contributions to the weighted histograms.

6.2.2. Game construction and formulation

Although it is natural to design a game to model the competition among multiple trackers, the construction of the game cannot be arbitrary, *e.g.* based on intuitions or heuristics, because the equilibrium of the game may not necessarily be a solution to MTT. For example, if we formulate

a naive non-cooperative game $[N, \{\mathbb{R}^2\}, \{\hat{\rho}_i(\mathbf{y}_i, \mathbf{y}_{-i})\}]$, where the players correspond to the individual trackers, the strategy for each player is the motion $\mathbf{y}_i \in \mathbb{R}^2$, and its utility $\hat{\rho}_i(\mathbf{y}_i, \mathbf{y}_{-i})$ is the generalized Bhattacharyya coefficient. This naive game is unable to assure a social optimal behavior (that corresponds to a good joint solution to MTT), because each tracker will try to solely increase its own utility. Special care has to be taken in the game construction.

A local optimum $\{\mathbf{y}_1^*, \cdots, \mathbf{y}_N^*\}$ of $J_3(\mathbf{y}_1, \cdots, \mathbf{y}_N) \stackrel{\triangle}{=} r_{tot}(\mathbf{y}_1, \cdots, \mathbf{y}_N)$ is a good solution to MTT. The solution must satisfy the Karush-Kuhn-Tucker (KKT) conditions,

$$\frac{\partial r_{tot}(\mathbf{y}_1, \cdots, \mathbf{y}_N)}{\partial \mathbf{y}_i}|_{\{\mathbf{y}_1^*, \cdots, \mathbf{y}_N^*\}} = 0, \quad \forall i \leq N. \tag{6.20}$$

Thus, the N.E. of the game we construct must also satisfy these conditions. In view of this, we design a game $G = [N, \{\mathbb{R}^2\}, \{r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i})\}]$. At the N.E. $\{\mathbf{y}_1^*, \cdots, \mathbf{y}_N^*\}$ of this game, $\forall$ player i and its optimal strategy $\mathbf{y}_i^*$, we have $r_{tot}(\mathbf{y}_i^*, \mathbf{y}_{-i}^*) \geq r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}^*), \forall \mathbf{y}_i$, by definition of N.E.. Since r_{tot} is continuous, $\nabla_{\mathbf{y}_i} r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}^*)|_{\mathbf{y}_i^*} = 0, \forall i$, is held at N.E.. Consequently, the N.E. also satisfies the KKT conditions of J_3. Therefore, this construction of the game is plausible, and maximizing J_3 is equivalent to finding the N.E.. Fortunately, this can be solved efficiently by a decentralized best response updating, as described below.

6.2.3. Finding a Nash Equilibrium

To find a N.E., we design a decentralized synchronous scheme to update the best response for each tracker. Namely, $\forall i$, assuming all the other trackers' locations $\mathbf{y}_{-i}$ are given, we find the best $\hat{\mathbf{y}}_i$ that maximizes the utility $r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i})$, *i.e.* to solve $\nabla_{\mathbf{y}_i} r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}) = 0$. The justification of this

iterative process can be found in Sec. 6.3. We have, $\forall i$,

$$\nabla_{\mathbf{y}_i} r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}) = \nabla_{\mathbf{y}_i} \tilde{r}_i(\mathbf{y}_i, \mathbf{y}_{-i}) + \sum_{j \neq i}^{N} \nabla_{\mathbf{y}_i} \tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j}) = 0. \tag{6.21}$$

Eq. 6.21 can be solved in a closed-form. To make the derivation clear, we denote

$$\eta_{ii}(\mathbf{x}_n) \triangleq \frac{\omega_i(\mathbf{x}_n)}{1 + \sum_{j=1, j\neq i}^{N} \frac{q_{jm}(\mathbf{x}_n)}{q_{im}(\mathbf{x}_n)} k(||\frac{\mathbf{x}_n - \mathbf{y}_j}{h_j}||^2)}. \tag{6.22}$$

$$\eta_{ji}(\mathbf{x}_n) \triangleq \frac{\omega_j(\mathbf{x}_n) k(||\frac{\mathbf{x}_n - \mathbf{y}_j}{h_j}||^2)}{(1 + \sum_{l=1, l\neq j}^{N} \frac{q_{lm}(\mathbf{x}_n)}{q_{jm}(\mathbf{x}_n)} k(||\frac{\mathbf{x}_n - \mathbf{y}_l}{h_l}||^2))^2}, \tag{6.23}$$

Then, we have

$$\nabla_{\mathbf{y}_i} \tilde{r}_i(\mathbf{y}_i, \mathbf{y}_{-i}) = \frac{1}{C_i h_i^2} \sum_{\Omega_i} \eta_{ii}(\mathbf{x}_n) g(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)(\mathbf{x}_n - \mathbf{y}_i), \tag{6.24}$$

and for $i \neq j$, we have,

$$\nabla_{\mathbf{y}_i} \tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j}) = -\frac{1}{C_j h_i^2} \sum_{\Omega_j \cap \Omega_i} \eta_{ji}(\mathbf{x}_n) g(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)(\mathbf{x}_n - \mathbf{y}_i). \tag{6.25}$$

Please note $\mathbf{y}_i$ merely influences $\tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j})$ through the overlapped region $\{\mathbf{x}_n \in \Omega_j \cap \Omega_i\}$ and $g(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)$ is uniform for Epanechnikov kernel. $\nabla_{\mathbf{y}_i} \tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j})$ acts as a force of the jth tracker that pushes away the ith tracker.

Plugging Eq. 6.24 and Eq. 6.25 to Eq. 6.21, we can solve the best $\hat{\mathbf{y}}_i$ given $\mathbf{y}_{-i}$ in a closed form. To make things clear, we define two more coefficients $w_{ii}(\mathbf{x}_n)$ and $w_{ji}(\mathbf{x}_n)$ for pixel $\mathbf{x}_n \in \Omega_i$,

$$w_{ii}(\mathbf{x}_n) \triangleq \frac{1}{C_i h_i^2} \eta_{ii}(\mathbf{x}_n) g(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2), \forall \mathbf{x}_n \in \Omega_i, \tag{6.26}$$

$$w_{ji}(\mathbf{x}_n) \triangleq \begin{cases} -\frac{1}{C_j h_i^2}\eta_{ji}(\mathbf{x}_n) g(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2) & \mathbf{x}_n \in \Omega_i \cap \Omega_j \\ 0 & \mathbf{x}_n \notin \Omega_i \cap \Omega_j \end{cases}. \tag{6.27}$$

We have,

$$\begin{aligned} \nabla_{\mathbf{y}_i} r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}) &= \sum_{j=1}^{N} \nabla_{\mathbf{y}_i} \tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j}) \\ = \quad \sum_{\Omega_i} \mathbf{x}_n \sum_{j=1}^{N} w_{ji}(\mathbf{x}_n) - \mathbf{y}_i \sum_{\Omega_i} \sum_{j=1}^{N} w_{ji}(\mathbf{x}_n) &= 0. \end{aligned} \tag{6.28}$$

Therefore, considering the interference of the target i to all the others targets and given the locations of other targets, the best $\hat{\mathbf{y}}_i$ that maximizes the utility is

$$\hat{\mathbf{y}}_i = \frac{\sum_{j=1}^{N} \sum_{\Omega_i} \mathbf{x}_n w_{ji}(\mathbf{x}_n)}{\sum_{j=1}^{N} \sum_{\Omega_i} w_{ji}(\mathbf{x}_n)}, \quad \forall i. \tag{6.29}$$

For each frame $I^{(t)}$, when N trackers approach to each other, we can iteratively update $\mathbf{y}_i, i = 1, \cdots, N$ by Eq. 6.29. This iterative process reaches an equilibrium that achieves a local optimum of the joint motion estimation.

A geometrical explanation is the following. We can view $\hat{\mathbf{y}}_i$ as a combination of forces $\hat{\mathbf{y}}_{i \leftarrow j}$ which is the solution to $\nabla_{\mathbf{y}_i} \tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j}) = 0$ as

$$\hat{\mathbf{y}}_{i \leftarrow j} = \frac{\sum_{\Omega_i} \mathbf{x}_n w_{ji}(\mathbf{x}_n)}{\sum_{\Omega_i} w_{ji}(\mathbf{x}_n)}. \tag{6.30}$$

$\hat{\mathbf{y}}_{i \leftarrow j}$ acts as tracker j's counter force to tracker i when considering $\mathbf{y}_i$'s interference in $\tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j})$. This can be visualized in Fig. 6.2.

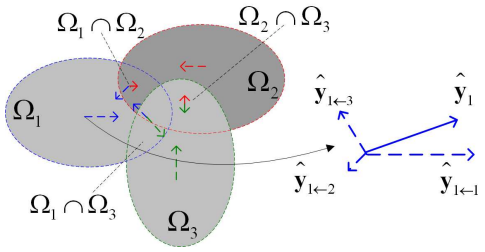


Figure 6.2. Illustration of force combination for $\hat{\mathbf{y}}_i$.

6.2.4. Algorithm summary

We summarize our game-theoretic MTT algorithm. If a subset of targets approach each other, and their hypotheses are overlapped (the distances less than a threshold), we generate a game and use the algorithm in Fig. 6.4 to search for the N.E. If one target is isolated from others we use Mean-shift tracker. The procedure is summarized in Fig. 6.3.

Input : Frame $I^{(t)}$, target models $\{\mathbf{q}_i\}$, and initial states of the set of individual trackers $\theta^{(t-1)} = \{\mathbf{y}_i^{(t-1)}, h_i^{(t-1)}\}$ for $i = 1, \cdots, N'$.
Output: Tracking results $\theta^{(t)} = \{\mathbf{y}_i^{(t)}, h_i^{(t)}\}$ for $i = 1, \cdots, N'$.

(1) Divide trackers into different groups if they are in close vicinity.

(2) For each group of trackers, if it has more than one tracker in the group, generate a game and call the algorithm in Fig. 6.4, otherwise call Mean-shift tracker [15].

(3) For each individual tracker, search $h_i^{(t)}$ with discrete scale factors $\{0.95, 1, 1.05\}$ to maximize its generalized Bhattacharyya coefficient $\hat{\rho}(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_{-i})$.

Figure 6.3. Procedure of game-theoretic MTT.

6.3. Game Theoretic Analysis

In the game G we have constructed, the utility function of each player is the joint matching $r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}) = \sum_i^N \tilde{r}(\mathbf{y}_i, \mathbf{y}_{-i})$, which forces an individual tracker to take other trackers' influences into consideration rather than only focusing on its own interest. $\nabla_{\mathbf{y}_i} \tilde{r}_j(\mathbf{y}_j, \mathbf{y}_{-j})$, *i.e.* the sensitivity

Input : Frame I, target models $\{\mathbf{q}_i\}$, and initial states of the set of individual trackers $\{\mathbf{y}_i^0, h_i\}$ for $i = 1, \cdots, N$.
Output: Target locations $\{\hat{\mathbf{y}}_i, i = 1, \cdots, N\}$ at the equilibrium.

(1) For each tracker i, determine Ω_i and calculate $\hat{\mathbf{p}}_i(\mathbf{y}_i, \mathbf{y}_{-i})$ by Eq. 6.9.

(2) In order to calculate $\nabla_{\mathbf{y}_i} \tilde{r}_i(\mathbf{y}_i, \mathbf{y}_{-i})$ in Eq. 6.24, for each pixel $\mathbf{x}_n \in \Omega_i$, calculate
- $\omega_i(\mathbf{x}_n)$ by Eq. 6.14,
- $\eta_{ii}(\mathbf{x}_n)$ by Eq. 6.22,
- $w_{ii}(\mathbf{x}_n)$ by Eq. 6.26.

(3) In order to calculate $\nabla_{\mathbf{y}_j} \tilde{r}_i(\mathbf{y}_i, \mathbf{y}_{-i})$ in Eq. 6.25 (note switch subscript i and j), for tracker $j \neq i, \Omega_i \cap \Omega_j \neq \emptyset$, for each pixel $\mathbf{x}_n \in \Omega_i \cap \Omega_j$, calculate
- $\eta_{ij}(\mathbf{x}_n)$ according to Eq. 6.23,
- $w_{ij}(\mathbf{x}_n)$ according to Eq. 6.27.

(4) For tracker i, calculate $\hat{\mathbf{y}}_i$ given $\mathbf{y}_{-i}$ by Eq. 6.29.

(5) If all $\{\hat{\mathbf{y}}_i \; \forall i = 1, \cdots, N\}$ are stationary, exit; otherwise go to Step 1.

Figure 6.4. Algorithm for finding N.E. in game-theoretic MTT.

of tracker j's matching w.r.t tracker i's motion $\mathbf{y}_i$, can be regarded as a price tracker j charges tracker i and counter reacts to $\mathbf{y}_i$ through $\hat{\mathbf{y}}_{i \leftarrow j}$.

To analyze whether the Nash Equilibrium can be achieved by the best response updating for game $G = [N, \{\mathbb{R}^2\}, \{r_{tot}(\mathbf{y}_1, \cdots, \mathbf{y}_N)\}]$, we resort to the following definition and theorem in the supermodular game theory [92, 97].

Definition 2. *A game $G = \{N, S, \{f_i\}\}$ is a supermodular (submodular) game if the set S of feasible joint strategies is a sublattice, and each utility function f_i is supermodular (submodular) function on S.*

Theorem 3. *In a supermodular (submodular) game $G = \{N, S, \{f_i\}\}$, (a) there exists at least one Nash Equilibrium; (b) if each player starts from any feasible strategy and uses best response updating, then the joint strategies will eventually converge to a Nash Equilibrium.*

For details about supermodular games, we refer the readers to Chapter 4 in [92] and Chapter 7 in [97].

Based on the supermodular game theory, to show the best response updating can reach a N.E., a sufficient condition includes 1) the solution of Eq. 6.21 is a best response of $\hat{\mathbf{y}}_i$ given fixed $\mathbf{y}_{-i}$, and 2) the game G is a supermodular/submodular game. Condition 1 is satisfied since $r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i})$ is concave on $\mathbf{y}_i$ in that the Epanechnikov kernel function k is non-negative and strictly concave. The details are given in the first part of Appendix C. The condition 2 can be satisfied in certain $\Omega_i, i = 1, \cdots, N$ where each utility function is submodular function, which can be checked as a by-product in the best response updating as given in the second part of Appendix C.

6.4. Experiments and Discussions

We demonstrate the proposed game-theoretic MTT by using both synthesized and real video (downloaded from *Google Video*). The basic individual tracker is a Mean-shift trackers with 32×32 2D histogram in the Hue-Saturation space. To purely evaluate the performance of the proposed method, we do not incorporate motion dynamic prior, object detectors, and background subtraction, although it is easy to incorporate them. The method is implemented in C++ and tested on Pentium IV 3GHz PC. Empirically, the best response updating converges very quickly within 3-10 iterations, so the computations are almost the same as that in multiple independent Mean-shift trackers.

6.4.1. Example of best response updating

First, we show an example of the best response updating for tracking the hands and the face in a sign language video. The first 4 images in Fig. 6.5 show the positions of the hands and the face at the first 3 iterations and at the last iteration during the best response updating. We observe that the

sum of generalized Bhattacharyya coefficients $\sum_{i=1}^{3} \hat{\rho}(\mathbf{y}_i, \mathbf{y}_{-i})$ monotonically increases as shown in the last graph. But the individual $\hat{\rho}(\mathbf{y}_i, \mathbf{y}_{-i})$ may be up and down. This is a rather difficult case because the hands and the face share the same skin tones. In our method, the competition ends up at an equilibrium that gives a good estimation of them.

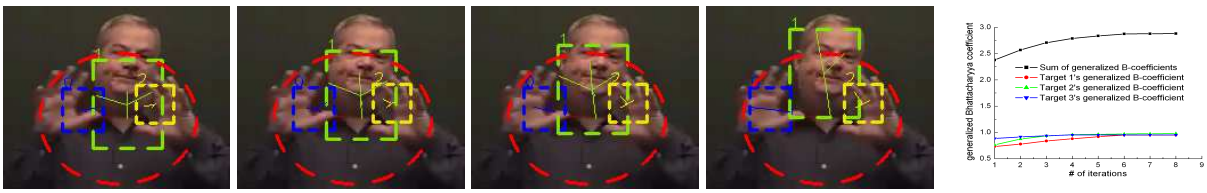


Figure 6.5. Illustration of best response updating procedure: iteration #0, 1, 2, and 8.

6.4.2. Synthesized video

We synthesize two videos in which there are 3 different targets and 5 identical targets, respectively. The backgrounds include random noise and 10-20 small targets that are wandering randomly. Frame samples are shown in Fig. 6.6. The trackers are drawn in different colors and a red dash ellipse indicates the group of trackers that are engaged in the game. The final motion $\hat{\mathbf{y}}_i$ are drawn at the centers of the targets. From the test results, the competition among the targets leads to an equilibrium and largely avoids the coalescence problem.

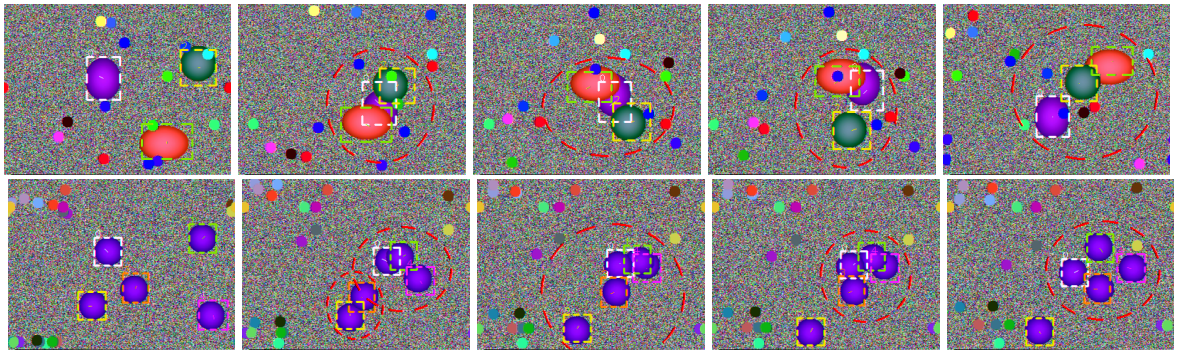


Figure 6.6. Tracking synthesized video: (1st row) 3 different targets for frame #1, 15, 42, 427, and 500; (2nd row) 5 identical targets for frame #1, 13, 19, 20, 25.

6.4.3. Real video

We further test the proposed approach in real sign language and sports videos. These are very challenging tests for MTT. The hand gesturing in sign language video (Fig. 6.7) is fast and the hand shape is deformable. Since the color of the hands and the face are quite similar, when the hands moving in front of the head, it is very likely that independent trackers will fail as shown in the 2nd row of Fig. 6.7. On the contrary, in our method, the interference from the face tracker to the hands tends to push the hands away from the face, which greatly alleviates coalescence phenomenons.

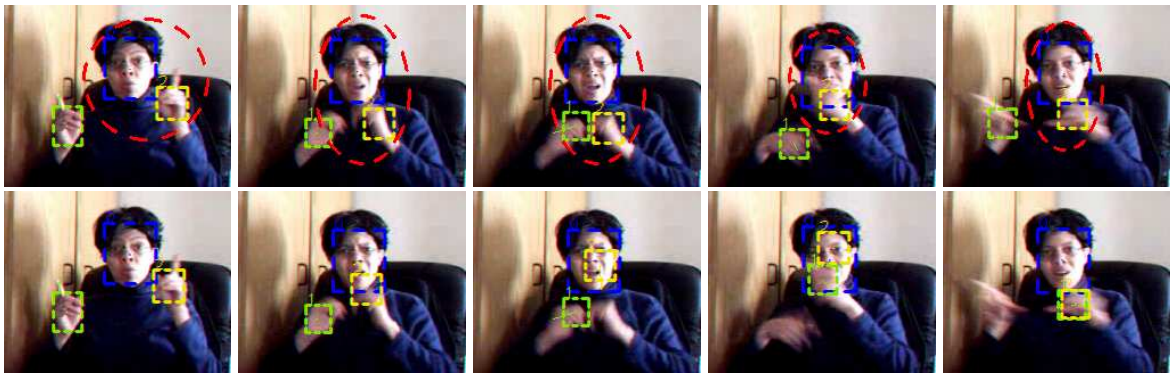


Figure 6.7. Tracking [`sign language`] for frame #1, 171, 172, 305, and 325, (1st row) game-theoretic MTT trackers and (2nd row) multiple independent trackers.

Sports video is another large category where the athletes generally wear similar uniforms and may have very complicated interactions. Therefore tracking people in sports video is a very difficult task. We show the tracking results for `kid soccer`, `free style soccer` and `volleyball` in Fig. 6.8. The proposed method can follow people with complicated occlusions. The comparison to the results of independent trackers are shown below our results, where one single target often traps multiple trackers.

Figure 6.8. (1st row) tracking [`kid soccer`] for frame #40, 64, 79, 101, 109; (3rd row) tracking [`free style soccer`] for frame #1, 100, 250, 280, and 300; (5th row) tracking [`volleyball`] for frame #1, 15, 40, 50, and 120.

6.4.4. Discussions

In this chapter we have introduced a new view of game theory to the study of multiple target tracking. The competition of individual trackers is formulated as a game and we bridge the solution to the joint motion estimation and the Nash Equilibrium of the game. Consequently, the maximization of the joint likelihood can be decentralized. The N.E. of this game can be solved by an efficient iterative procedure in a closed form. The proposed method achieves promising results in tracking quasi-identical targets in both synthesized and real video sequences.

CHAPTER 7

Conclusions

In this thesis, I mainly summarize my work on object-level visual tracking and present the context-aware and attentional visual tracking algorithms for a single target, and a game-theoretic multiple target tracking algorithm. The proposed algorithms mainly focus on how to handle the large variations of targets in real-world video sequences efficiently in order to enhance the generality and reliability of visual tracking algorithms. Since the target variations are unpredictable and tracking algorithms have to deal with them in an unsupervised way, adaptive target observation models with flexible matching criteria are critical to the success of a tracking algorithm.

Using subspace tracking as an example, we reveal that directly updating the observation model with the latest previous tracking results is a chicken-egg problem in nature without any bottom-up constraints. In viewing of this, we propose two novel ideas to enhance and adapt non-stationary observation models: context-aware tracking and attentional tracking. In context-aware tracking, the tracker mines some auxiliary objects automatically as the spatial contexts of the target which have short-term strong motion correlation with the target, these context information can provide additional verification of the tracking results. This is a general method to improve long-term robust tracking, which is effective to deal with the short-term invalidation of the target observation model due to severe occlusion, targets moving out of image boundary, and the distraction of camouflage objects. Further, we present two implementations of attentional visual tracking, where the targets are represented by a rich pool of attentional regions that are stable in motion estimation. In spatially selective attentional tracking, a discriminative subset of attentional regions are dynamically

selected to locate the target, while, in granularity and elasticity adaptive attentional tracking, the scales of the attentional regions and the relative geometrical relations among the attentional regions are tuned to enhance the robustness of the observation model. This rich and redundant target representation is more tolerant to small target variations due to lighting changes and deformation, irregular partial occlusion, and inaccurate target initialization. The spatial selection or granularity and elasticity adaptation do not rely on adjusting the individual attentional region model or inducing new features during the tracking, which largely avoids the chicken-and-egg problem in on-line adaptation. The context-aware and attentional visual tracking algorithms bring novel insights to the visual tracking area and achieve exciting and promising experimental results on real-world video sequences in unconstrained environments.

Multiple target tracking poses additional difficulties in practice and need to be addressed eventually when tracking is applied to real applications. The main challenges are the coalescence problem when targets with similar appearances approaching each other and the high computation complexity due to the joint motion estimation. In the proposed game-theoretic multiple target tracking algorithm, we formulate the problem as a game where individual tracker competes against each other for the visual evidence while also induces interferences to the others. By designing an interference model for kernel-based trackers, the joint motion estimation is solved by seeking the Nash Equilibrium in a particular submodular game using best response updating, which has linear complexity with the number of targets.

Visual object tracking is a fundamental problem in computer vision and deserves more research efforts. For the future research, we will continue pursuing intelligent robust visual tracking algorithms for theoretical study and practical systems. On the theoretical aspect, how to integrate the proposed context-aware tracking and attentional tracking efficiently requires further investigation. The AVT algorithm adjusts the “visual attention” inside the targets to achieve robust matching,

while CAT algorithm resorts to external contexts in the scene, *i.e.* temporally motion correlated regions, to verify the results. How to scalably and automatically fuse these two strategies in a principled way remains open. In terms of practical system design, how to infer more motion parameters, *e.g.* the aspect ratio, how to extract invariant image features with high repeatability as the attentional regions, and how to utilize the prior knowledge about the target and scene, are of great importance to a practical application.

Appendix

Appendix A

Lemma 1. *The solution of the following problem:*

$$\min_{A} \mathtt{tr}(\mathbf{A}^T\mathbf{C}\mathbf{A}), \quad s.t., \quad \mathbf{A}^T\mathbf{A} = I, \tag{7.1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times r}$*, and* $\mathbf{C} = \mathbf{Z}\mathbf{Z}^T \in \mathbb{R}^{m\times m}$*, is given by the eigenvectors that corresponds to the* r *smallest eigenvalues of* $\mathbf{C}$*.*

Proof: It is easy to figure it out. Actually this is the same as the proof for the procedure in PCA. Based on the Lemma, the proof of Theorem 1 is given by the following: Performing SVD on $\mathbf{A}_t$, we have $\mathbf{A}_t = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{V} \in \mathbb{R}^{r\times r}$. It is easy to see: $\mathbf{P}_t = \mathbf{U}\mathbf{U}^T$. Then the optimization problem in Eq. 3.4 is equivalent to:

$$\underset{U}{\operatorname{argmin}}\, J_3(\mathbf{U}) = \underset{U}{\operatorname{argmin}}\{\mathtt{tr}(\mathbf{U}^T\mathbf{C}_t^-\mathbf{U}) - \mathtt{tr}(\mathbf{U}^T\mathbf{C}_t^+\mathbf{U}) + \alpha||\mathbf{U}\mathbf{U}^T - \mathbf{P}_{t-1}||_F^2\}, \quad s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}.$$

The Lagrangian is given by:

$$L(\mathbf{U}) = J_3(\mathbf{U}) + \lambda(\mathbf{U}^T\mathbf{U} - \mathbf{I}).$$

Let $\mathbf{U} = [\mathbf{e}_1, \ldots, \mathbf{e}_r]$, and we have:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{e}} &= 2(\mathbf{C}_t^- - \mathbf{C}_t^+)\mathbf{e} + 2\alpha(\mathbf{e}\mathbf{e}^T - \mathbf{P}_{t-1})\mathbf{e} + 2\lambda\mathbf{e} \\ &= 2(\mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1})\mathbf{e} + 2\lambda\mathbf{e}. \end{aligned}$$

Thus, $\mathbf{e}$ is an eigenvector of $\hat{\mathbf{C}} = \mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1}$. The minimization problem is solved by finding the r eigenvectors that correspond to the r smallest eigenvalues of $\hat{\mathbf{C}}$. **Q.E.D.**

Appendix B

Definition of inconsistency in a two-node Gaussian Markov network

The theorem of inconsistency between two Gaussian sources and the proofs were first proposed by Gang *et. al.* [40]. We consider to define the inconsistency in a two-node Gaussian Markov network, as shown in Fig. 7.1, where the two observation nodes are Gaussian random vectors $\mathbf{z}_1 \sim N(\mu_1, \Sigma_1)$ and $\mathbf{z}_2 \sim N(\mu_2, \Sigma_2)$ with $\mu_1, \mu_2 \in \mathbb{R}^n$. Therefore, the compatible functions between observation nodes and the hidden nodes are Gaussian, *i.e.*,

$$\phi(\mathbf{x}_i, \mathbf{z}_i) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_i|}} \mathbf{e}^{-\frac{1}{2}(\mathbf{z}_i - \mathbf{x}_i)^\top \mathbf{\Sigma}_i^{-1} (\mathbf{z}_i - \mathbf{x}_i)}. \tag{7.2}$$

Assume $\mathbf{x}_1$ can be predicted by a function f of $\mathbf{x}_2$, the compatible or the potential function of $\mathbf{x}_1$ and $\mathbf{x}_2$ can be expressed as a Gaussian

$$\psi(\mathbf{x}_1, \mathbf{x}_2) = \frac{\exp\left\{-\frac{(\mathbf{x}_1 - f(\mathbf{x}_2))^\top (\mathbf{x}_1 - f(\mathbf{x}_2))}{2\sigma_{12}^2}\right\}}{\sqrt{(2\pi)^n}\sigma_{12}^n} \tag{7.3}$$

$$\doteq \frac{\exp\left\{-\frac{(\mathbf{x}_1 - \mathbf{A}_{12}\mathbf{x}_2 - \mu_{12})^\top (\mathbf{x}_1 - \mathbf{A}_{12}\mathbf{x}_2 - \mu_{12})}{2\sigma_{12}^2}\right\}}{\sqrt{(2\pi)^n}\sigma_{12}^n}, \tag{7.4}$$

which indicates if $\mathbf{x}_1$ and $f(\mathbf{x}_2)$ can be regarded as being generated from one common model and σ_{12}^2 is the scalar variance. When f is nonlinear, we linearize it by Taylor expansion, *i.e.*, $\mu_{12} = f(\mathbf{0})$ and $\mathbf{A}_{12} = \frac{\partial f_{12}(\mathbf{x}_2)}{\partial \mathbf{x}_2}|_{\mathbf{x}_2 = \mathbf{0}}$ is the $n \times n$ Jacobian. So we only consider the linearized relation of $\mathbf{x}_1$ and $\mathbf{x}_2$ in Eq. 7.4.

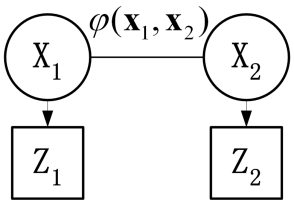


Figure 7.1. Two-node Markov network.

The variance σ_{12}^2 indeed models the uncertainties between the estimate $\mathbf{x}_1$ and the neighborhood estimate $\mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$. Assume $\mathbf{A}_{12}$ and μ_{12} are known, given all the $\{\mathbf{z}_1, \mathbf{z}_2\}$, the estimate of σ_{12}^2 is a natural indicator of whether $\mathbf{x}_1$ and $\mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$ should be consensus, *i.e.*, if σ_{12}^2 is very small, then they should be in consensus since $\psi(\mathbf{x}_1, \mathbf{x}_2)$ is approaching to an impulse delta function, and vice versa.

The Bayesian MAP inference of $\mathbf{x}_1$ and the ML estimate of σ_{12} can be obtained by the following Bayesian EM algorithm [73], *i.e.*,

$$\begin{aligned} \mathbf{x}_1 &= (\boldsymbol{\Sigma}_1^{-1} + \frac{1}{\sigma_{12}^2}\mathbf{I})^{-1} \\ &\times (\boldsymbol{\Sigma}_1^{-1}\mathbf{z}_1 + \frac{1}{\sigma_{12}^2}(\mathbf{A}_{12}\mathbf{x}_2 + \mu_{12})) \end{aligned} \tag{7.5}$$

$$\sigma_{12}^2 = \frac{1}{n}(\mathbf{x}_1 - \mathbf{A}_{12}\mathbf{x}_2 - \mu_{12})^\top(\mathbf{x}_1 - \mathbf{A}_{12}\mathbf{x}_2 - \mu_{12}) \tag{7.6}$$

Fixing σ_{12}, the E-Step in Eq. 7.5 obtains the MAP estimate of $\mathbf{x}_1$ by fixed-point iteration. Fixing $\mathbf{x}_1$ and $\mathbf{x}_2$, the M-Step in Eq. 7.6 maximizes $p(\mathbf{x}_1, \mathbf{x}_2|\sigma_{12}, \mathbf{z}_1, \mathbf{z}_2)$ w.r.t. σ_{12}. Combining the two steps together also constitutes a fixed-point iteration for σ_{12}^2.

We measure the consistency of two observation sources $\mathbf{z}_1$ and $\mathbf{z}_2$ by examining if their estimates $\mathbf{x}_1$ and $\mathbf{x}_2$ are in consensus, *i.e.* if $\mathbf{x}_1$ is predictable from $\mathbf{x}_2$ through a linear relation $\mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$ with small variance σ_{12}^2. Therefore, when $\mathbf{z}_1$ and $\mathbf{z}_2$ are consistent, the estimate of $\mathbf{x}_1$ and $\mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$ will show a consensus, *i.e.*, they will be almost the same. In this case, from

Eq. 7.6, the estimate of σ_{12}^2 will always approach to zero, *i.e.*, zero is the only fixed-point. On the contrary, if they are inconsistent, the estimate of $\mathbf{x}_1$ and $\mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$ may deviate from each other, *i.e.*, the convergent results of σ_{12}^2 may be non-zero. This indicates that there exist non-zero fixed-points for σ_{12}^2. These motivate us to define the inconsistency of two Gaussian sources as follows.

Definition 3. *If zero is the only fixed-point for* σ_{12}^2 *in the Bayesian EM,* i.e. *in Eq. 7.5 and Eq. 7.6,* $\{\mathbf{z}_1, \mathbf{\Sigma}_1\}$ *and* $\{\mathbf{z}_2, \mathbf{\Sigma}_2\}$ *are* consistent*; if there exist non-zero fixed-points for* σ_{12}^2*, they are* inconsistent.

Proof of inconsistency criterion

Given the aforementioned definition of inconsistency for two Gaussian sources in two-node Markov network, we propose a sufficient condition to check the convergent value of σ_{12}^2 as stated in Theorem 2. The basic idea of the proof is to check if Eq. 7.6 has non-zero solutions. With some manipulations we express Eq. 7.6 as a function $F(\sigma_{12}^2)$ in Eq. 7.12. Then, we show if the condition number C_p of $\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2$ satisfies Eq. 4.15 in Theorem 2, there exist two positive numbers $0 < k_2 < k_1$ such that $F(k_1) < 0$ and $F(k_2) > 0$, which indicates there is a non-zero solution. If C_p satisfies Eq. 4.16, $F(\sigma_{12}^2) < 0$ for all $\sigma_{12}^2 > 0$, thus there is no non-zero solution for Eq.7.6.

PROOF. Fixing σ_{12}^2, the fixed-point iteration in Eq. 7.5 is guaranteed to obtain the exact MAP estimate on the joint posterior Gaussian. For simplification of notation, we denote $\hat{\mathbf{x}}_2 = \mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$ and $\hat{\mathbf{z}}_2 = \mathbf{A}_{12}\mathbf{z}_2 + \mu_{12}$. Define $\mathbf{P} = \mathbf{\Sigma}_1 + \mathbf{\Sigma}_2$ and $\mathbf{S} = \mathbf{P} + \sigma_{12}^2\mathbf{I}$. The convergent result in the

E-Step in Eq. 7.5 is the same as,

$$\begin{bmatrix} \mathbf{x}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} (\sigma_{12}^2\mathbf{I} + \hat{\mathbf{\Sigma}}_2)\mathbf{S}^{-1}\mathbf{z}_1 + \mathbf{\Sigma}_1\mathbf{S}^{-1}\hat{\mathbf{z}}_2 \\ \hat{\mathbf{\Sigma}}_2\mathbf{S}^{-1}\mathbf{z}_1 + (\sigma_{12}^2\mathbf{I} + \mathbf{\Sigma}_1)\mathbf{S}^{-1}\hat{\mathbf{z}}_2 \end{bmatrix}. \tag{7.7}$$

Embedding it to the M-Step in Eq. 7.6, we have

$$\sigma_{12}^2 = \frac{1}{n}\sigma_{12}^2\sigma_{12}^2(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^\top\mathbf{S}^{-1}\mathbf{S}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2). \tag{7.8}$$

To prove Theorem 2, since zero is a solution of σ_{12}^2 for Eq. 7.8, we only need to analyze the existence of non-zero solutions of σ_{12}^2 for

$$\frac{1}{n}\sigma_{12}^2(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^\top\mathbf{S}^{-1}\mathbf{S}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) - 1 = 0. \tag{7.9}$$

$\mathbf{P}$ is the sum of two covariance matrices so it is *real positive definite*. Thus there exists an orthonormal matrix $\mathbf{Q}$ such that $\mathbf{P} = \mathbf{Q}\mathbf{D}_p\mathbf{Q}^\top$, where

$$\mathbf{D}_p = diag[\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2]$$

is the eigen-matrix with $\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_n^2 > 0$ and $C_p = \frac{\sigma_1^2}{\sigma_n^2}$. Then we have $\mathbf{S} = \mathbf{Q}\mathbf{D}_s\mathbf{Q}^\top$, where

$$\mathbf{D}_s = diag[\sigma_1^2 + \sigma_{12}^2, \sigma_2^2 + \sigma_{12}^2, \ldots, \sigma_n^2 + \sigma_{12}^2].$$

Furthermore, $\mathbf{S}^{-1} = \mathbf{Q}^\top\mathbf{D}_s^{-1}\mathbf{Q}$ where

$$\mathbf{D}_s^{-1} = diag[\frac{1}{\sigma_1^2 + \sigma_{12}^2}, \frac{1}{\sigma_2^2 + \sigma_{12}^2}, \ldots, \frac{1}{\sigma_n^2 + \sigma_{12}^2}].$$

We also denote $\tilde{\mathbf{z}} = \mathbf{Q}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) = [\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n]^\top$. Then, we can simplify the expressions in Eq. 7.9 and Eq. 4.15 in Theorem 2 (Sec. 4.3) as,

$$\frac{1}{n}\sigma_{12}^2(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^\top \mathbf{S}^{-2}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) = \frac{1}{n}\sum_{i=1}^{n} \frac{\sigma_{12}^2 \tilde{z}_i^2}{(\sigma_i^2 + \sigma_{12}^2)^2}, \tag{7.10}$$

$$\frac{1}{n}(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^\top \mathbf{P}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2}. \tag{7.11}$$

From Eq. 7.10, we express Eq. 7.9 as a function $F(\cdot)$ of σ_{12}^2 and only need to analyze the solution of σ_{12}^2 for

$$F(\sigma_{12}^2) = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} \cdot \frac{1}{2 + \frac{\sigma_i^2}{\sigma_{12}^2} + \frac{\sigma_{12}^2}{\sigma_i^2}} - 1 = 0. \tag{7.12}$$

Now we proceed to prove the conclusions in Theorem 2.

Denote the left-hand side of Eq. 4.15 in Theorem 2 as d and plug Eq. 7.11 in, thus Eq. 4.15 means

$$d = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} > 2 + \sqrt{\frac{\sigma_1^2}{\sigma_n^2}} + \sqrt{\frac{\sigma_n^2}{\sigma_1^2}} \geq 4.$$

When $\sigma_{12}^2 = k_1 = (d-2)\sigma_1^2$, for any i, we have

$$\frac{1}{2 + \frac{\sigma_i^2}{\sigma_{12}^2} + \frac{\sigma_{12}^2}{\sigma_i^2}} < \frac{1}{2 + 0 + d - 2} = \frac{1}{d}.$$

Thus,

$$F(k_1) < \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} \cdot \frac{1}{d} - 1 = 0.$$

When $\sigma_{12}^2 = k_2 = \sqrt{\sigma_1^2 \sigma_n^2}$, for any i,

$$\frac{1}{2 + \frac{\sigma_i^2}{\sigma_{12}^2} + \frac{\sigma_{12}^2}{\sigma_i^2}} \geq \frac{1}{2 + \frac{\sigma_n^2}{k_2} + \frac{k_2}{\sigma_1^2}} = \frac{1}{2 + \sqrt{\frac{\sigma_1^2}{\sigma_n^2}} + \sqrt{\frac{\sigma_n^2}{\sigma_1^2}}} \geq \frac{1}{d},$$

thus

$$F(k_2) \geq \frac{1}{n}\sum_{i=1}^{n}\frac{\tilde{z}_i^2}{\sigma_i^2}\cdot\frac{1}{d} - 1 = 0.$$

Since $0 < k_2 < k_1$ and $F(\cdot)$ is continuous, there must exist a k_3 such that $k_2 \leq k_3 < k_1$ and $F(k_3) = 0$. This proves that the inequality Eq. 4.15 in Theorem 2 holds can indicate a non-zero solution for Eq. 7.9, namely there exists at least one non-zero fixed point for σ_{12}^2 in the Bayesian EM, which means the two Gaussian sources are not in consensus according to our definition of inconsistency. Thus, the first claim in Theorem 2 is proved.

Eq. 4.16 means $d = \frac{1}{n}\sum_{i=1}^{n}\frac{\tilde{z}_i^2}{\sigma_i^2} < 4$, then we have

$$F(\sigma_{12}^2) \leq \frac{1}{n}\sum_{i=1}^{n}\frac{\tilde{z}_i^2}{\sigma_i^2}\cdot\frac{1}{4} - 1 = \frac{d}{4} - 1 < 0$$

for all $\sigma_{12}^2 > 0$. Therefore, there does not exist a non-zero solution for Eq. 7.12. Eq. 4.16 in Theorem 2 is proven.

□

Appendix C

Proof that Eq. 6.29 is a best response

To show Eq. 6.29 is the best response of $\hat{\mathbf{y}}_i$ given fixed $\mathbf{y}_{-i}$, we need to show the solution $\hat{\mathbf{y}}_i$ of Eq. 6.21 is a global optimum of $r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i})$. We prove this by showing $r_{tot}(\mathbf{y}_i, \mathbf{y}_{-i}) = \sum_{j=1}^{N}\tilde{r}_j(\mathbf{y}_1, \cdots, \mathbf{y}_N)$ is concave.

Denote $\mathbf{y}_i = \{u_i, v_i\}$, given $\mathbf{y}_{-i}$ are fixed, $\tilde{r}_i(\mathbf{y}_i) = \tilde{r}_i(u_i, v_i)$ and $\tilde{r}_j(\mathbf{y}_i) = \tilde{r}_j(u_i, v_i)$. Note $g(||\frac{\mathbf{x}_n - \mathbf{y}_i}{h_i}||^2)$ is positive and uniform for Epanechnikov kernel. From Eq. 6.24 and Eq. 6.25, we

have

$$\frac{\partial \tilde{r}_i(u_i,v_i)}{\partial u_i \partial v_i} = 0, \frac{\partial \tilde{r}_i(u_i,v_i)}{\partial u_i \partial u_i} = \frac{\partial \tilde{r}_i(u_i,v_i)}{\partial v_i \partial v_i} = -\sum_{\Omega_i} w_{ii}(\mathbf{x}_n).$$

$$\frac{\partial \tilde{r}_j(u_i,v_i)}{\partial u_i \partial v_i} = 0, \frac{\partial \tilde{r}_j(u_i,v_i)}{\partial u_i \partial u_i} = \frac{\partial \tilde{r}_j(u_i,v_i)}{\partial v_i \partial v_i} = -\sum_{\Omega_i} w_{ji}(\mathbf{x}_n).$$

So in the Hessian matrix of $\sum_{j=1}^{N} \tilde{r}_j(u_i,v_i)$, the elements on the diagonal are $-\sum_{j=1}^{N}\sum_{\Omega_i} w_{ji}(\mathbf{x}_n)$ and 0 for elements off the diagonal, it is negative definite which indicates it is concave over $\mathbf{y}_i = \{u_i, v_i\}$.

Conditions for G being a submodular game

To show a game is supermodular (submodular) game we need to show the joint strategy space is defined on a sublattice and all utility functions are supermodular (submodular) functions on the joint strategy space. Any non-empty compact subset of $\mathbb{R}^n$ is a sublattice of $\mathbb{R}^n$ [97]. So the first requirement is satisfied in our game G. For the second condition, we have this theorem [97]:

Theorem 4. *Let $X \subset \mathbb{R}^n$ and $f : X \to \mathbb{R}$. The function f is supermodular iff it satisfies increasing (decreasing) differences on X. If f is twice differentiable, f is supermodular iff $\frac{\partial^2 f}{\partial x_i \partial x_j} \geq 0$, or submodular iff $\frac{\partial^2 f}{\partial x_i \partial x_j} \leq 0, \forall i, j$.*

Denote $\mathbf{y}_i = \{u_i, v_i\}$. We need to examine $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial u_i \partial u_j}$, $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial v_i \partial v_j}$, $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial u_i \partial v_j}$, and $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial v_i \partial u_j}$ for $i \neq j$. In addition, we need to check $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial u_k \partial u_l}$, $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial v_j \partial v_l}$, $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial u_j \partial v_l}$, and $\frac{\partial \tilde{r}_i(\mathbf{y}_i,\mathbf{y}_{-i})}{\partial v_j \partial u_l}$ for $j, l \neq i$. Whether these conditions hold depends on the $\{\Omega_i, i = 1, \cdots, N\}$ and can be checked analytically. We observe the constructed game G is submodular when the occlusion regions are small and the kernel centers are not occluded. Each term can be derived from Eq. 6.24 and Eq. 6.25, thus these conditions can be checked as a by-product in the best response updating given $\{\Omega_i, i = 1, \cdots, N\}$.

References

[1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 798 – 805, NYC, June 17-22, 2006. 16, 18, 19, 64, 82, 95

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, pages 487 – 499, Santiago, Chile, 12-15, 1994. 38

[3] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for near neighbor probem in high dimensions. In *The ACM Symposium on Foundations of Computer Science (FOCS'06)*, pages 459 – 468, Berkeley, CA, October 22-24, 2006. 17, 64

[4] Shai Avidan. Support vector tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 26:1064 – 1072, aug 2004. 9, 16, 18, 21

[5] Shai Avidan. Ensemble tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 494 – 501, June 20-26, 2005. 9, 16, 18, 19, 64, 82, 90

[6] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press, 1988. 9

[7] Yaakov Bar-Shalom and Edison Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451 – 460, November 1975. 13

[8] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'98)*, pages 232 – 237, Santa Barbara, CA, June 23-25, 1998. 13, 14, 16, 18, 52

[9] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *European Conf. on Computer Vision (ECCV'96)*, pages 329–342, Cambridge, UK, April 1996. 9, 13, 14, 21

[10] Thomas Brox, Andres Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conf. on Computer Vision (ECCV'04)*, volume 4, pages 25 – 36, May 2004. 14

[11] Aeron Buchanan and Andrew Fitzgibbon. Interactive feature tracking using K-D tress and dynamic programming. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 626 – 633, NYC, June 17-22, 2006. 17, 70

[12] Erik B. Budderth, Michael I. Mandel, William T. Freeman, and Alan S.Sillsky. Distributed occlusion reasoning for tracking with nonparametric belief progagation. In *Neural Information processing Systems (NIPS'04)*, Vancouver, Canada, December 13-18, 2004. 13, 14, 17

[13] Robert T. Collins. Mean-shift blob tracking through scale space. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, volume 2, pages 234 – 240, Madison, WI, June 16-22, 2003. 13, 17, 19

[14] Robert T. Collins and Yanxi Liu. On-line selection of discriminative tracking features. In *IEEE Int'l Conf. on Computer Vision (ICCV'03)*, volume 2, pages 346–352, Nice, France, October 13-16, 2003. 22, 23, 29, 64, 82

[15] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, volume 2, pages 142–149, Hilton Head Island, South Carolina, June 13-15, 2000. 9, 13, 14, 17, 18, 28, 75, 82, 103, 111

[16] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5):564 – 577, May 2003. 17, 41, 43

[17] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991. 73

[18] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(2):138–150, February 1996. 9, 14

[19] Mayur Datar, Piotr Indyk, Nicole Immorlica, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the ACM Symposium on Computational Geometry (SoCG'04)*, Brooklyn, NY, June 9-11, 2004. 64, 69, 70, 74

[20] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, volume 2, pages 126–133, Hilton Head Island, SC, June 13- 15, 2000. 9, 14, 17

[21] Maneesh Dewan and Gregory D. Hager. Toward optimal kernel-based tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 618– 625, NYC, June 17-22, 2006. 9, 19

[22] Zhimin Fan, Ying Wu, and Ming Yang. Multiple collaborative kernel tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 502 – 509, San Diego, CA, June 20-26, 2005. 9, 17, 19

[23] Zhimin Fan, Ming Yang, Ying Wu, Gang Hua, and Ting Yu. Efficient optimal kernel placement for reliable visual tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 658 – 665, NYC, June 17-22, 2006. 9, 19, 67, 68

[24] Rob Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, volume 2, pages 264 – 271, June 16-22, address = 2003. 9

[25] Andrew Fitzgibbon and Andrew Zisserman. On affine invariant clustering and automatic cast listing in movies. In *European Conf. on Computer Vision (ECCV'02)*, volume 3, pages 304 – 320, 2002. 41

[26] EC funnded CAVIAR project/IST 2001 37540. http://homepages.inf.ed.ac.uk/rbf/caviar/. 58

[27] Bogdan Georgescu, Ilan Schimshoni, and Peter Meer. Mean shift based clustering in high dimensions: a texture classification example. In *IEEE Int'l Conf. on Computer Vision (ICCV'03)*, volume 1, pages 456–463, Nice, France, October 13-16, 2003. 70

[28] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 260 – 267, NYC, June 17-22, 2006. 9, 16, 19, 64, 90

[29] Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover's distance. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 220 – 227, Washington, DC, June 27 - July 2, 2004. 17, 70

[30] Chengen Guo, Song-Chun Zhu, and Ying Nian Wu. Towards a mathematical theory of primal sketch and sketchability. In *IEEE Int'l Conf. on Computer Vision (ICCV'03)*, volume 2, pages 1228 – 1235, Nice, France, October 13-16, 2003. 83

[31] Greg Hager and Peter Belhumeur. Real-time tracking of image regions with changes in geoemtry and illumination. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'96)*, pages 403–410, San Francisco, CA, June 18-20, 1996. 13, 17, 18, 21, 81

[32] Greg Hager, Maneesh Dewan, and Charles Stewart. Multiple kernel tracking with SSD. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 790–797, Washington, DC, June 27 - July 2, 2004. 9, 19, 69, 82

[33] Arun Hampapur, Lisa Brown, Jonathan Connell, Ahmet Ekin, Norman Haas, Max Lu, Hans Merkl, Sharath Pankanti, Andrew Senior, Chiao-Fe Shu, and Yingli Tian. Smart video surveillance. *IEEE Signal Processing Mag.*, 22(2):38– 51, March 2005. 8, 13

[34] Mei Han, Wei Xu, Hai Tao, and Yihong Gong. An algorithm for multiple object trajectory trackingt. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 864 – 871, Washington, DC, June 27 - July 2, 2004. 14

[35] Chris Harris and Mike Stephens. A combined corner and edge detector. In *ALVEY Vision Conference*, pages 147 – 151, 1998. 86

[36] Jeffrey Ho, Kuang-Chih Lee, Ming-Hsuan Yang, and David J. Kriegman. Visual tracking using learned linear subspace. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 782–789, Washington, DC, June 27 - July 2, 2004. 9, 19, 22, 23, 29

[37] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185 – 203, 1981. 14, 18

[38] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, and Wen-Fong Hu. Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Trans. Intell. Transport. Syst.*, 7(2):175–187, June 2006. 9, 13

[39] Gang Hua and Ying Wu. Sequential mean field variational analysis of structured deformable shapes. *Computer Vision and Image Understanding*, 101(2):87– 99, February 2004. 9, 13, 14, 17

[40] Gang Hua and Ying Wu. Measurement integration under inconsistency for robust tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 650– 657, NYC, June 17-22, 2006. 51, 121

[41] Carine Hue and Jean-Pierre Le Cadre. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. Signal Processing*, 50(2):309 – 325, February 2002. 17, 101

[42] Piotr Indyk. *Nearest neighbors in high-dimensional spaces. Handbook of Discrete and Computational Geometry*, chapter 39. CRC Press, 2nd edition, 2004. 17

[43] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *The 30th Annual ACM Symposium on Theory of Computing (STOC'98)*, pages 604 – 613, Dallas, TX, May 23-26, 1998. 17, 64, 69, 70

[44] Sergey Ioffe and David Forsyth. Human tracking with mixtures of trees. In *IEEE Int'l Conf. on Computer Vision (ICCV'01)*, volume 1, pages 690–695, Vancouver, Canada, July 7-14, 2001. 14

[45] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *European Conf. on Computer Vision (ECCV'96)*, pages 343–356, Cambridge, UK, April 1996. 9, 13, 14, 17, 18

[46] Michael Isard and Andrew Blake. CONDENSATION - conditional density propagation for visual tracking. *Int'l Journal of Computer Vision (IJCV)*, 29(1):5–28, May 1998. 9, 17

[47] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *IEEE Int'l Conf. on Computer Vision (ICCV'01)*, volume 2, pages 34–41, Vancouver, Canada, July 7-14, 2001. 9, 14, 101

[48] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck. *Machine Vision*. McGrawHill, Inc, 1995. 42

[49] Allan D. Jepson, David Fleet, and Thomas El-Maraghi. Robust online appearance models for visual tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, volume 1, pages 415–422, Hawaii, December 8-14, 2001. 17, 18, 22, 23, 64

[50] Simon J.Julier and Jeffrey K.Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference (ACC'97)*, pages 2369 – 2373, Albuquerque, New Mexico, June 4-6, 1997. 50

[51] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME - Journal of Basic Engineering*, 82:35 – 45, 1960. 9, 13

[52] James M. Rehg Takeo Kanade. Model-based tracking of self-occluding articulated objects. In *IEEE Int'l Conf. on Computer Vision (ICCV'95)*, pages 612–617, Cambridge, MA, June 20- 23, 1995. 14

[53] Robert Kaucic, Barney Dalton, and Andrew Blake. Real-time lip tracking for audio-visual speech recognition applications. In *European Conf. on Computer Vision (ECCV'96)*, volume 2, 1996. 8

[54] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(11):1805 – 1819, November 2005. 14, 101

[55] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang, and David J. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, volume 1, pages 313 – 320, Madison, Wisconsin, June 18-20, 2003. 13, 22

[56] Kuang-Chil Lee and David J. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 852 – 859, San Diego, CA, June 20-26, 2005. 13, 19, 22

[57] Marius Leordeanu and Robert Collins. Unsupervised learning of object features from video sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1142 – 1149, San Diego, CA, June 20-25, 2005. 38

[58] Open Source Computer Vision Library. http://www.intel.com/technology/computing/opencv/. 52

[59] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *IEEE Conf. on Image Processing (ICIP'02)*, volume 1, pages 900 – 903, New York, NY, 2002. 52

[60] Hwasup Lim, Vlad I. Morariu, Octavia I. Camps, and Mario Sznaier. Dynamic appearance modeling for human tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 751 – 757, NYC, June 17-22, 2006. 22

[61] Jongwoo Lim, David Ross, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems 17 (NIPS'04)*, pages 801–808, Vancouver, Canada, December 13-18, 2004. 9, 19, 22

[62] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE Int'l Conf. on Computer Vision (ICCV'99)*, volume 2, pages 1150 – 1157, Corfu, Greece, 20-27, 1999. 9, 41

[63] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, pages 121 – 130, April 1981. 14, 18

[64] John MacCormick and Andrew Blake. A probabilistic exclusion principle for tracking multiple objects. In *IEEE Int'l Conf. on Computer Vision (ICCV'99)*, pages 572 – 578, Corfu, Greece, 21-22 1999. 14, 101

[65] Stephen J. McKenna, Yogesh Raja, and Shaogang Gong. Tracking colour objects using adaptive mixture models. *Image and Vision COmputing*, 17:225 – 231, October 1999. 18

[66] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *IEEE Int'l Conf. on Computer Vision (ICCV'01)*, volume 1, pages 525 – 531, Vancouver, Canada, July 7 - 14, 2001. 41

[67] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European Conf. on Computer Vision (ECCV'02)*, pages 128–142, May.27-Jun.2, 2002. 86, 87

[68] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int'l Journal of Computer Vision*, 65(1-2):43 – 72, November 2005. 9, 84, 86

[69] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J. Little, and David G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conf. on Computer Vision (ECCV'04)*, volume 1, pages 28 – 39, Prague, Czech Republic, May 2004. 9, 14, 19, 101

[70] Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. The MIT Press, Cambridge, Massachusetts, 1999. 62, 63

[71] Stavros Paschalakis and Miroslaw Bober. Real-time face detection and tracking for mobile videoconferencing. *Real-Time Imaging*, 10(2):81 – 94, April 2004. 8

[72] Harold E. Pashler. *The Psychology of Attention*. The MIT Press, Cambridge, Massachusetts, 1998. 62, 72

[73] Vladimir I. Pavlovic. *Dynamic Bayesian Networks for Information Fusion with Application to Human-Computer Interfaces*. PhD thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, 1999. 122

[74] Faith Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 829 – 836, San Diego, CA, June 20-25, 2005. 68, 75

[75] Zhen Qian, Dimitris N. Metaxas, and Leon Axel. Boosting and nonparametric based tracking of tagged mri cardiac boundaries. In *Int'l Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, volume 1, pages 636 – 644, Copenhagen, October 1-6, 2006. 8

[76] DARPA Grand Challenge Race. http://www.darpa.mil/grandchallenge. 9

[77] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. Tracking people by learning their appearance. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(1):65 – 81, January 2007. 14

[78] Christopher Rasmussen and Gregory D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(6):560–576, June 2001. 9, 14, 17, 18, 101

[79] Donald B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automat. Contr.*, 24(6):843 – 854, December 1979. 9, 13, 14, 101

[80] David Ross, Jongwoo Lim, and Ming-Hsuan Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proceedings of the Eighth European Conference on Computer Vision (ECCV'04)*, volume 1, pages 215–227, Prague, Czech Republic, May 2004. 19, 22, 23

[81] Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. In *IEEE Int'l Conf. on Computer Vision (ICCV'05)*, volume 1, pages 42 – 49, October 2005. 14

[82] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'97)*, pages 731–737, San Juan, Puerto Rico, June 17-19, 1997. 28

[83] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *IEEE Int'l Conf. on Computer Vision (ICCV'98)*, pages 1154 – 1160, Bombay, India, January 1998. 9

[84] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, Seattle, WA, June 21 - 23, 1994. 9, 14, 41, 87

[85] Josef Sivic and Andrew Zisserman. Video google: a text retrieval approach to object matching in videos. In *IEEE Int'l Conf. on Computer Vision (ICCV'03)*, volume 2, pages 1470 – 1477, Nice, France, October 13-16, 2003. 38

[86] Josef Sivic and Andrew Zisserman. Video data mining using configurations of viewpoint invariant regions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 488 – 495, Washington, DC, June 27 - July 2, 2004. 38

[87] Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, volume 2, pages 246 – 252, Fort Collins, CO, June23-25, 1999. 9

[88] Zehang Sun, George Bebis, and Ronald Miller. On-road vehicle detection: a review. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(5):694 – 711, May 2006. 9

[89] Feng Tang and Hai Tao. Object tracking with dynamic feature graph. In *IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS'05)*, pages 25 – 32, Beijing, China, October 15-16, 2005. 84

[90] Hai Tao, , Harpreet S. Sawhney, and Rakesh Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(1):75 – 89, 2002. 17

[91] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar. A sampling algorithm for detecting and tracking multiple targets. In *Proceedings of ICCV'99 Workshop on Vision Algorithms: Theory and Practice*, pages 53 – 58, Corfu, Greece, 21-22 1999. 101

[92] Donald M. Topkis. *Supermodularity and Complementarity*. Priceton University Press, 1999. 112, 113

[93] Kentaro Toyama and Andrew Blake. Probabilistic tracking in a metric space. In *IEEE Int'l Conf. on Computer Vision (ICCV'01)*, volume 2, pages 50–57, Vancouver, Canada, July 7-14, 2001. 9, 18, 21

[94] Son Tran and Larry Davis. Robust object tracking with regional affine invaraiant features. In *IEEE Int'l Conf. on Computer Vision (ICCV'07)*, Rio de Janeiro, Brazil, October 14-20, 2007. 16, 18, 82, 84

[95] Jaco Vermaak, Patrick Perez, Michel Gangnet, and Andrew Blake. Towards improved observation models for visual tracking: Selective adaptation. In *European Conf. on Computer Vision (ECCV'02)*, volume 1, pages 645–660, Copenhagen, Denmark, May 2002. 22

[96] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, volume I, pages 511 – 518, Hawaii, December 8-16 2001. 9

[97] Rakesh V. Vohra. *Advanced Mathematical Economics*. Routledge, 2005. 112, 113, 127

[98] Jianyu Wang, Xilin Chen, and Wen Gao. Online selecting discriminative tracking features using particle filter. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1037 – 1042, San Diego, CA, June 20-26, 2005. 19, 22

[99] Shuo Wang, Xiaocao Xiong, Yan Xu, Chao Wang, Weiwei Zhang, Xiaofeng Dai, and Dongmei Zhang. Face-tracking as an augmented input in video games: enhancing presence, role-playing and control. In *ACM Special Interest Group on Computer-Human Interaction (SIGCHI'06)*, pages 1097 – 1106, Montreal, Canadian, April 22 - 27, 2006. 8

[100] Oliver Williams, Andrew Blake, and Roberto Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *IEEE Int'l Conf. on Computer Vision (ICCV'03)*, volume 2, pages 353– 360, Nice, France, October 13-16, 2003. 18

[101] Oliver Williams, Andrew Blake, and Roberto Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(8):1292 – 1304, August 2005. 9

[102] Ying Wu, Gang Hua, and Ting Yu. Tracking articulated body by dynamic markov network. In *IEEE Int'l Conf. on Computer Vision (ICCV'03)*, volume 2, pages 1094–1101, Nice, France, October 13-16, 2003. 14

[103] Ying Wu and Thomas S. Huang. Color tracking by transductive learning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, volume 1, pages 133–238, Hilton Head Island, SC, June 13- 15, 2000. 17, 18, 19

[104] Ying Wu and Thomas S. Huang. A co-inference approach to robust visual tracking. In *IEEE Int'l Conf. on Computer Vision (ICCV'01)*, volume 2, pages 26–33, Vancouver, Canada, July 7-14, 2001. 18, 19, 22

[105] Ying Wu and Thomas S. Huang. Human hand modeling, analysis and animation in the context of human computer interaction. *IEEE Signal Processing Mag.*, 18(3):51 – 60, May 2001. 8

[106] Ying Wu and Thomas S. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *Int'l Journal of Computer Vision (IJCV)*, 58(1):55–71, June 2004. 9, 18, 19

[107] Ying Wu, John Lin, and Thomas S. Huang. Capturing natural hand articulation. In *IEEE Int'l Conf. on Computer Vision (ICCV'01)*, volume 2, pages 426–432, Vancouver, Canada, July 7-14, 2001. 14

[108] Ying Wu, John Lin, and Thomas S. Huang. Analyzing and capturing articulated hand motion in image sequences. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(12):1910–1922, December 2005. 8, 14

[109] Bin Yang. Projection approximation subspace tracking. *IEEE Trans. Signal Processing*, 43(1):95–107, January 1995. 27

[110] Ming Yang and Ying Wu. Tracking non-stationary appearances and dynamic feature selection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1059 – 1066, San Diego, CA, June 20-25, 2005. 9

[111] Ming Yang, Ying Wu, and Shihong Lao. Intelligent collaborative tracking by mining auxiliary objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 697 – 704, NYC, June 17-22, 2006. 39, 45

[112] Ming Yang, Junsong Yuan, and Ying Wu. Spatial selection for attentional visual tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'07)*, Minneapolis, MN, June 18-23, 2007. 17, 18, 82

[113] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, December 2006. 13

[114] Zhaozheng Yin and Robert Collins. On-the-fly object modeling while tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'07)*, Minneapolis, MN, June 17-22 2007. 16, 18, 82, 84

[115] Ting Yu and Ying Wu. Collaborative tracking of multiple targets. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 1, pages 834 – 841, Washington, DC, June 27 - July 2, 2004. 14

[116] Ting Yu and Ying Wu. Differential tracking based on spatial-appearance model (SAM). In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 720 – 727, NYC, June 17-22, 2006. 18, 19, 82

[117] Tao Zhao and Ram Nevatia. Tracking multiple humans in crowded environment. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, volume 2, pages 1063 – 1069, Washington, DC, June 27 - July 2, 2004. 14, 101

[118] Shaohua Kevin Zhou, Rama Chellappa, and Baback Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Processing*, 13(11):1491 – 1506, November 2004. 17, 18

[119] Xiang Sean Zhou, Dorin Comaniciu, and Alok Gupta. An information fusion framework for robust shape tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(1):115–129, January 2005. 8, 14

[120] Zhiwei Zhu and Qiang Ji. Eye gaze tracking under natural head movements. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 918 – 923, San Diego, CA, June 20-26, 2005. 8

Vita

Education

- Ph.D, Aug. 2004 – Jun. 2008

  Department of Electrical Engineering and Computer Science

  Northwestern University, Evanston, IL, USA

  Thesis: Context-aware and attentional visual object tracking

  Advisor: Prof. Ying Wu
- M.E., Department of Electronic Engineering, Sept. 2001 – Jul. 2004

  Tsinghua University, Beijing, P.R.C

  Thesis: Research on fast algorithms in H.264 video coding.
- B.E., Department of Electronic Engineering, Aug. 1997 – Sept. 2001

  Tsinghua University, Beijing, P.R.C

  Thesis: PC-based MPEG-II video decoder in DVB-S system.

Selected Publications:

- **Ming Yang**, Ying Wu. Granularity and elasticity adaptation in visual tracking. *IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, June 24-26, 2008, (CVPR'2008).
- **Ming Yang**, Qiong Liu, Thea Turner, Ying Wu. Vital sign estimation from passive thermal video. *IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, June 24-26, 2008, (CVPR'2008).

- Jialue Fan, **Ming Yang**, Ying Wu. A bi-subspace model for robust visual tracking. *IEEE Int'l Conf. on Image Processing*, San Diego, CA, October 12-15, 2008 (ICIP'2008).
- Yan Gao, **Ming Yang**, Xiaonan Zhao, Bryan Pardo, Ying Wu, Thrasyvoulos N. Pappas, Alok Choudhary. Image spam hunter. *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, March 30 - April 4, 2008 (ICASSP'2008).
- **Ming Yang**, Ting Yu, Ying Wu. Game-theoretic multiple target tracking. *IEEE Int'l Conf. on Computer Vision*, Rio de Janeiro, Brazil, October 14-20, 2007 (ICCV'2007).
- **Ming Yang**, Junsong Yuan, Ying Wu. Spatial selection for attentional visual tracking. *IEEE Conf. on Computer Vision and Pattern Recognition*, Minneapolis, MN, June 18-23, 2007 (CVPR'2007).
- Shengyang Dai, **Ming Yang**, Ying Wu, Aggelos K. Katsaggelos. Detector Ensemble. *IEEE Conf. on Computer Vision and Pattern Recognition*, Minneapolis, MN, June 18-23, 2007 (CVPR'2007).
- Junsong Yuan, Ying Wu, **Ming Yang**. Discovery of collocation patterns: from visual words to visual phrases. *IEEE Conf. on Computer Vision and Pattern Recognition*, Minneapolis, MN, June 18-23, 2007 (CVPR'2007).
- Zhimin Fan, **Ming Yang**, Ying Wu. Multiple collaborative kernel tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.29, no.7, pp.1268-1273, July 2007.
- Junsong Yuan, Ying Wu, **Ming Yang**. Frequent itemsets to semantically meaningful visual patterns. *ACM Int'l Conf. on Knowledge Discovery and Data Mining*, San Jose, CA, August 13-15, 2007 (ACM SIGKDD'2007).
- **Ming Yang**, Senthil Periaswamy, Ying Wu. False positive reduction in lung GGO nodule detection with 3D volume shape descriptor. *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, HI, vol.1 pp.437-440, April 15-17, 2007 (ICASSP'2007).
- **Ming Yang**, Ying Wu, Shihong Lao. Mining auxiliary objects for tracking by multibody grouping. *IEEE Int'l Conf. on Image Processing*, San Antonio, TX, September 16-19, 2007 (ICIP'2007).

- **Ming Yang**, Ying Wu, Shihong Lao. Intelligent collaborative tracking by mining auxiliary objects. *IEEE Conf. on Computer Vision and Pattern Recognition*, New York City, NY, vol.1, pp.697-704, June 17-22, 2006 (CVPR'2006).
- Zhimin Fan, **Ming Yang**, Ying Wu, Gang Hua, Ting Yu. Efficient optimal kernel placement for reliable visual tracking. *IEEE Conf. on Computer Vision and Pattern Recognition*, New York City, NY, vol.1, pp.658-665, June 17-22, 2006 (CVPR'2006).
- Shengyang Dai, **Ming Yang**, Ying Wu, Aggelos K. Katsaggelos. Tracking motion-blurred targets in video. *IEEE Int'l Conf. on Image Processing*, Atlanta, GA, pp.2389-2392, October. 8-11, 2006 (ICIP'2006).
- **Ming Yang**, James Crenshaw, Bruce Augustine, Russell Mareachen, Ying Wu. Face detection for automatic exposure control in handheld camera. *IEEE Int'l Conf. on Computer Vision Systems*, New York City, NY, pp.17, January 5-7, 2006 (ICVS'2006).
- **Ming Yang**, Ying Wu. Tracking non-stationary appearances and dynamic feature selection. *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, vol.2, pp.1059-1066, June 20-26, 2005 (CVPR'2005).
- Zhimin Fan, Ying Wu, **Ming Yang**. Multiple collaborative kernel tracking. *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, vol.2, pp.502-509, June 20-26, 2005 (CVPR'2005).
- **Ming Yang**, Huijuan Cui, Kun Tang. Efficient tree structured motion estimation using successive elimination. *IEE Proceedings - Vision, Image and Signal Processing*, vol.151, no.5, pp.369-377, October 2004 (IEE-VIS).
- **Ming Yang**, Wensheng Wang. Fast macroblock mode selection based on motion content classification in H.264/AVC. *IEEE Int'l Conf. on Image Processing*, Singapore, pp.741-744, October 24-27, 2004 (ICIP'2004).