Visual Non-rigid Object Tracking

in Dynamic Environments

by

Hadi Firouzi

B.Sc., Azad University of Tehran, 2005
M.Sc., University of Tehran, 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE COLLEGE OF GRADUATE STUDIES

(Electrical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

June 2013

Abstract

This research presents machine vision techniques to track an object of interest visually in an image sequence in which the target appearance, scale, orientation, shape, and position may significantly change over time. The images are captured using a non-stationary camera in a dynamic environment in a gray-scale format, and the initial location of the target is given. The contributions of this thesis include the introduction of two robust object tracking techniques and an adaptive similarity measure which can significantly improve the performance of visual tracking.

In the first technique, the target is initially partitioned into several sub-regions, and subsequently each sub-region is represented by two distinct adaptive templates namely immediate and delayed templates. At every tracking step, the translational transformation of each sub-region is preliminarily estimated using the immediate template by a multi-start gradient-based search, and then the delayed template is employed to correct the estimation. After this two-step optimization, the target is tracked by robust fusion of the new sub-region locations. From the experiments, the proposed tracker is more robust against appearance variance and occlusion in comparison with the traditional trackers.

Similarly, in the second technique the target is represented by two heterogeneous Gaussian-based templates which models both short- and long-term changes in the target appearance. The target localization of the latter technique features an interactive multi-start optimization that takes into account generic transformations using a combination of sampling- and gradient-based algorithms in a probabilistic framework. Unlike the two-step optimization of the first method, the templates are used to find the best location of the target, simultaneously. This approach further increases both the efficiency and accuracy of the proposed tracker.

Lastly, an adaptive metric to estimate the similarity between the target model and new images is proposed. In this work, a weighted L2-norm is used to calculate the target similarity measure. A histogram-based classifier is learned on-line to categorize the L2-norm error into three classes which subsequently specify a weight to each L2-norm error. The inclusion of the

proposed similarity measure can remarkably improve the robustness of visual tracking against severe and long-term occlusion.

Preface

This dissertation is an original work by the author, Hadi Firouzi. I was responsible for all major area of research including problem definition, data collection, concept formulation, algorithm development, computer programming, implementation, experimental analysis, and manuscript composition. Homayoun Najjaran was the supervisory author on this work and was involved throughout the project in problem definition and manuscript composition.

Chapter 3 has been published in the Pattern Recognition journal [H. Firouzi and H. Najjaran, Robust decentralized multi-model adaptive template tracking, Pattern Recognition, vol. 45, no. 12, pp. 4494-4509, Dec. 2012.] with some modifications.

A version of Chapter 4 has been submitted after revision to Computer Vision and Image Understanding journal. Also, the material presented in Chapter 5 has been compiled in a journal paper format and been submitted to the Image and Vision Computing journal.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to acknowledge the help and support of the wonderful people around me without whom this thesis would have not been possible.

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Homayoun Najjaran for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance not only helped me through my research, but also was a tremendous insight into my life.

Besides, I would like to thank the rest of my thesis committee: Dr. Rudolf Seethaler, Dr. Thomas Johnson, and Prof. Herbert Yang, and the anonymous reviewers for their insightful comments, which have helped me to greatly improve this thesis.

I also thank my wife, Dr. Golriz Rezaei, and other fellows in the Advanced Control and Intelligent Systems laboratory at the University of British Columbia for their help and support during my Ph.D. journey.

Last but not the least, my sincerest thanks goes to my parents Zahra Alishahi and Rahim Firouzi, for giving birth to me at the first place and supporting me spiritually throughout my life, and my brother and sisters for their invaluable supports all in my life.

Dedication

This thesis is dedicated to my beloved wife
who has unconditionally and patiently supported me all the time,
and to my loving parents without whom it would not have been possible.

Chapter 1

Introduction

1.1 Motivations

"A picture is worth a thousand words". Evidences found from first nations show the existence of graphical figures and images in the life of mankind since early stages. Yet in modern life, visual information forms the main part of our perception from the world around us. Thanks to advances in electronics and computer hardware systems, inexpensive and powerful digital cameras are now common in our daily life as well as a wide range of industries from high-tech fields such as robotics to conventional domains such as agriculture. In the digital world, an image is viewed as a two dimensional matrix in which each cell represents the smallest part of an image known as a *pixel* and videos are, in general, a sequence of images captured from a camera. Computer vision is a field – categorized under the field of computer science, robotics, and artificial intelligence (AI) – that consists of capturing, processing, and understanding images or videos. Similar to our visual perception, using computer vision we are able to retrieve large amounts of information which can then be used for different purposes.

Among the tasks in the field of computer vision and robotics, motion analysis and specifically *visual tracking* has long been considered a challenging and important topic. In its simplest form, visual tracking is defined as the problem of locating three-dimensional (3D) target objects (such as a human or a car) in a two-dimensional (2D) image plane as they move around a scene [99]. The main reason for an extensive attention on visual tracking from researchers is its fundamental and essential role in many real-world applications including Automatic visual surveillance which is a system to detect, track, and understand activity of different targets such as humans in dynamic scenes (e.g., airport or mall) for the purpose of safety and/or security [18, 36, 41, 48, 53], Behavior analysis which recognize and learn a pattern of activity by tracking objects in a video [24, 89, 98], Motion capture and animation [2, 28, 79], Video games (e.g., EyeToy [75]), Vehicle navigation and tracking [6, 40], Traffic monitoring [17], Intelligent preventive safety systems [39, 52, 74], Human computer interaction [13], Industrial robotics

[71], and Medical diagnosis [3, 45, 93].

Estimating the trajectory of objects in the image plane has a long tradition in robotic and computer vision research [37, 56, 57, 80]. A large number of methods have been introduced in the literature which can only track the target with some limitations and under a controlled situation. Nevertheless, visual tracking is still considered as an unsolved problem under general conditions. These challenges are mainly due to the inevitable object appearance variations, scale changes, occlusion, illumination changes, image noise, unpredictable and complex motion, and cluttered and dynamic background. For instance, a target moving far from the camera can be occluded partially or fully for a short-term by some closer objects. The location and shape of targets may significantly change during the tracking task. Specifically, the main difficulty in tracking non-rigid objects – which has been emphasized in this work – is related to the high dimensional complexity and uncertainty in the real applications [91]. As a result, developing an efficient and robust non-rigid object tracking capable of attacking the mentioned problems is necessary to fulfill the demands for the current and future real-world machine vision applications.

1.2 Research Objective

The ultimate aim of this research is to develop a machine vision framework capable of tracking non-rigid objects (with variable appearance, shape, and scale) using sequential images in dynamic environments where other stationary or moving objects may coexist. It is assumed that the target object has been detected manually or automatically (by any existing object detection method) at the earliest frame and the goal is to adaptively track the object without any prior knowledge whereas the object appearance, shape, and scale are changing over time. It is noted that these changes are not drastic between two consecutive frames and some overlaps can be found due to the real-time image acquisition. Also the target object is a real-world object such as human, face, or car and cannot have a chaotic or huge movement between two consecutive images. However, both the camera and the target object can move freely in any direction.

1.3 Methodology

In general there are two wide categories in visual tracking namely region-based and feature-based methods. In the latter, the target object is tracked

based on several features which are extracted at every step while the former does not require preprocessing step to track the target and is able to find the next location of the target using the previous object images. In this research, I employ the region-based approach to develop my tracker because of several reasons. Firstly, extracted features do not cover all the visual and spatial information which can be obtained from the object image. Moreover, the accuracy and robustness of a feature-based method highly depends on the specific features set used. Therefore, these methods cannot obtain a satisfactory result under general conditions. Last but not least, feature extraction are computationally expensive and not suitable for real-time applications. This argument has been observed in the early stages of my research where I developed a feature-based non-rigid object tracker [30]. Moreover, the main focus of this research is on single target tracking based on target representation and localization. Multi-target tracking is usually considered as the problem of data association task which has its own roots in control theory [21].

1.4 Research contributions and publications

The proposed machine vision technique in Chapter 3 has been published in *Patter Recognition* journal [31]. A version of Chapter 4 and Chapter 5 has been submitted to *Computer Vision and Image Understanding* and *Image and Vision Computing* respectively.

Before starting my PhD program, I studied several preliminary works which were main motivations to choose *Visual Object Tracking* as my PhD research topic. In the first preliminary work, I focused on a real-time face tracking method using an eye-in-hand visual servoing system [30]. This work has been implemented on a motorized camera for the task of face detection and tracking. In another preliminary work, I proposed a new feature-based object detection and tracking technique [29]. This technique fuses visual and motional features to track real-world objects more effectively than the traditional techniques. Aligned with the PhD research, I proposed a scale adaptive non-rigid object tracking method which is robust to the object appearance and shape variations [32]. A modified and improved version of this work is presented in Chapter 4.

Most of these methods have been implemented in Matlab and a few in C++, visit `http://acis.ok.ubc.ca/~hfirouzi/` for several demo movies.

1.5 Research Scope and Structure

In the next chapter, the visual tracking problem is defined and described from different point of views. Also, fundamental tracking components including representation model and target localization algorithms are explained in detail.

After describing a typical visual tracker, in Chapter 3 my first tracking method is presented which was published in Pattern Recognition journal [31]. The target representation model used in this method consists of multiple decentralized and heterogeneous templates which are adaptively updated over time. Since it is assumed that the target is a non-rigid object, each part of the target is represented by two different templates namely immediate and delayed templates for modeling the short-term and long-term appearance variations. At every tracking step, first each template is tracked using a gradient-based optimization algorithm, and then the new location of the target is robustly estimated by fusion of that of the templates. The provided comparison results of this method with several state-of-the-art trackers show its accuracy and robustness against appearance and pose changes as well as illumination variation and occlusions.

Following my first visual tracker, in Chapter 4 a multi-template tracking method based on Gaussian functions is described. This tracker inputs several starting points to an interactive and parallel gradient-based search for finding the best location of the target at every image frame. From the experimental results provided in this chapter, the proposed tracker outperforms other state-of-the-art methods using several challenging image sequences.

In Chapter 5, an adaptive similarity measure for matching the target model and the received image is proposed. In this method, a histogram-based classifier is learned on-line to robustly classify the matching errors into three categories namely i) small appearance variations, ii) significant changes in the target appearance, and iii) large errors due to the outliers or occlusions. According to the error type, a different weight is assigned to each matching error. The accuracy and robustness of the proposed similarity measure has been compared with several robust regression methods and the result shows the superiority of my method against sever outliers and long-term occlusion.

At the end, in Chapter 6 several conclusions and potential future works are discussed.

Chapter 2

Background

Visual object tracking is a computer vision technique that detects, locates, and corresponds one or more image regions related to a unique target object in a streaming video or sequential 2-D[1] images. In spite of similarities in definition, detection and tracking algorithms do not solve the same problem. The main difference between these two is that the former is an off-line process which learns the model of the target object from the training data and the goal is to find and locate the best match(es) to the model in new images. For example, Viola and Jones [94] used a training set of positive and negative sample faces to train a boosting classifier which is then employed to detect faces in new images as an object model. Contrary, a tracker not only trains an on-line model of the target for the task of detection, but it also matches the objects between sequential image frames.

Based on the above definition, the first difficulty in a tracking algorithm is to learn an adaptive and robust object model as new information (i.e., new image frames) are received. Thus, the only information previously provided for a tracking method is the 2-D projection of the 3-D target object at the first image frame. In most cases, this 2-D projection cannot completely and accurately represent the object in entire image sequence due to the changes in appearance, pose, shape, and scale of the target object. Therefore, a representation model needs to be generated as the object is tracked.

Given the representation model, localization and corresponding a unique target object between consecutive images are other significant challenges associated to the visual tracking task. This is even more challenging as it has to be executed in real-time. From the localization point-of-view, object tracking can be partly similar to the image registration problem [69, 81, 90] as both methods optimize a likelihood type function. However, in tracking as opposed to registration the object appearance and location may slightly change between two consecutive images. Fortunately, in a visual tracking scenario, images are captured with proportionally high frame rate so that there are overlaps between consecutive images which imply that the object

[1] Two Dimensional

cannot move and change largely between sequential frames.

In the following sections, visual tracking is formally defined from different perspectives and its important components are explained in detail.

2.1 Tracking Components

Although a visual tracking method can be composed of several parts, it usually consists of four important components which are (1) object representation or appearance model, (2) object motion or dynamic model, (3) observation likelihood or similarity measure, and (4) search method or object localization algorithm. Thus. the accuracy, efficiency, and robustness of a method highly depend on all of these components. For instance, the target object can be accurately modeled and efficiently located by a method but tracking will fail if the similarity of the model and observation (i.e., images) is not measured properly. In the following subsections, these components are explained in detail.

2.1.1 Object Representation Model

The target object of a tracker can be anything that is of interest for further analysis. Depending on the application, objects vary; examples are a person walking on a sidewalk, a car on a highway, a moving face in an office like environment, a boat on a river, an animal in a movie, or several particles in water. In general, the object is represented by a shape model or a joint shape-appearance model. Shapes basically locate the object region in the image, whereas, in the latter, the object is represented by both shape and appearance. In this section, different appearance models followed by common shape representations are presented.

1. **Shape Representation**

   Various geometrical models from a single point to complex shape have been used to specify the object region in the image. In the following subsections, several frequently used models are presented.

   (a) **Points**

   One centroid point or a set of points can be used to represent the target object in an image (Figure 2.1). These point are usually found by a feature extraction algorithm such as SIFT [58]. Single points are usually used to locate small objects. For instance, Figure 2.1(a) shows a number of birds which are flying together.

Since the target objects (i.e., birds) have a high contrast with the background (i.e., sky) gray-scale values, each bird can be presented by a point. In this sub-figure, the white lines are the trajectory of each target.

(a) Single points representing small objects (birds) in a simple background [85], white lines show the birds' trajectories

(b) Multiple points representing two complex objects (i.e., a car and a human) in a cluttered environment [60]

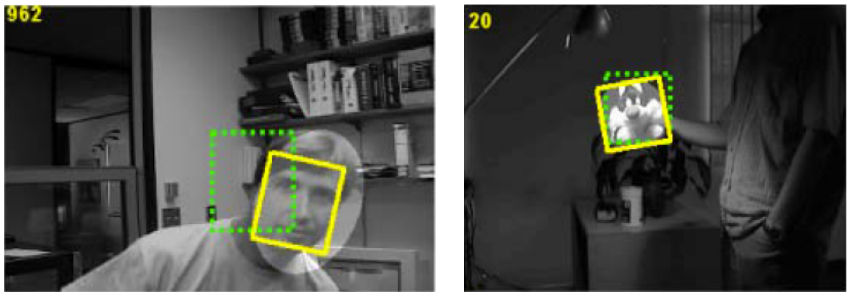
Figure 2.1: Different point representation models

On the other hand, multiple points can represent a complex object in a cluttered environment. Illustrated in Figure 2.1(b), two objects, which are car and human, have been tracked based on a multiple point representation model. In this sub-figure, the green rectangle shows the target object, the red and the white small-boxes are the points which are tracked and missed respectively, also the yellow small-boxes specify the object trajectory.

(b) **Rectangular or elliptical patches**

A deformable rectangle or ellipse can be used to represent the shape of the target object. Figure 2.2 shows different rectangular and elliptical representation of a target in two publicly available image sequences which are dudek[2] and sylv[3] sequences. In this figure, the yellow rectangle is obtained by robust incremental tracker [82], the highlighted ellipse is the result of WSL tracker [43], and the green dashed rectangle is the object representation by Mean Shift tracker[21]. Although deformable primitive shapes

[2] `http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html`
[3] `http://www.cs.toronto.edu/~dross/ivt/`



(a) Dudek image sequence (b) Sylv image sequence

Figure 2.2: Rectangular (yellow and green dotted rectangles) and elliptical (highlighted ellipse) shape representations of a face and a dog doll [82]

may contain irrelevant information such as background pixels, they are commonly used for representing both simple and non-rigid objects in different environments.

(c) **Contour and silhouette**

(a) Four colorful silhouettes representating different persons [70]

(b) A contour representating a car [15]

Figure 2.3: Silhouette and contour representations

The shape of the target object can be defined by either a contour or silhouette. Illustrated in Figure 2.3 the boundary of the target object is defined as its contour, and the silhouette is the region inside the contour. Figure 2.3(b) shows the contour of a car,

and in Figure 2.3(a) the silhouettes of four people are specified. In general, contour and silhouette representations are suitable where the object and background has significant color or gray-scale contrast.

(d) **Skeleton**

Object shape can be represented by its skeleton. Figure 2.4 shows the skeleton representation of a human toy. This model is usually used for representing articulated objects which have a changeable shape and structure.

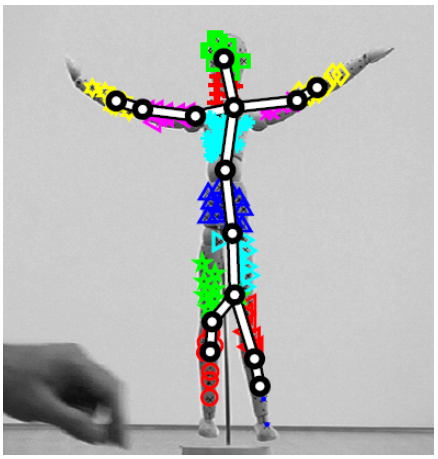


Figure 2.4: Skeleton shape representation [83], body parts are illustrated by several primitive shapes in different colors

(e) **Mixed**

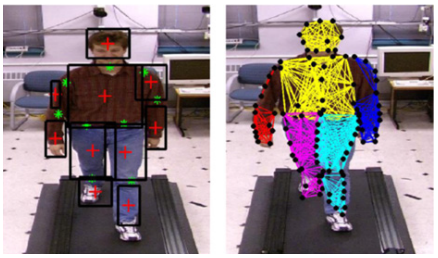


Figure 2.5: Mixed shape representation [4], the shape of the object (person) is represented by different rectangles (left) and connected points (right)

Shown in Figure 2.5, different shapes can be combined to rep-

resent an object. In this figure, each human body part (e.g., head, arm, and leg) is defined by a rectangle, and these boxes are accordingly connected to represent the full body. A mixed representation model is usually composed of several primitive shapes which are linked with each others to form a more complex shape and structure. These models can be used for non-rigid and articulated objects.

2. **Appearance Representation**

   Generally speaking, object appearance is represented either by direct (image-based) or indirect (feature-based) models. In latter approach, different feature descriptors such as texture [96], local invariant features [58], Haar-like features [7], and histograms [21, 49] are used to model the object appearance. Feature descriptors such as SIFT [58] can – to some degree – handle illumination and scale changes. However, their suitability and robustness may significantly change from one application to another depending on the appropriateness of the feature descriptors used.

   On the other hand, direct models use the object image usually without any preprocessing to represent the appearance. Templates and subspace representations are common direct appearance models which have been widely used for the task of visual tracking. In the following subsections, different direct (region-based) and indirect (feature-based) appearance models are presented.

   (a) **Region-based**

   In this subcategory, the target object is represented using the pixel values usually without any preprocessing.

   i. **Template**
   Object appearance can be presented by a fix template which is indeed the object image at the first frame. Since a fix template can only represent the object from a limited point of view, different adaptive templates have been introduced. A template can be adapted to represent rigid as well as non-rigid objects. In Figure 2.6(a), a template (the small picture on top left) has been used to model the appearance a moving car.

   ii. **Sub-spaces representation**

(a) A template representing a car [62]

(b) A PCA-based subspace representing a face [82]

Figure 2.6: Direct appearance representation models

Subspace representation is another way of direct modeling of the object appearance. Subspace representation usually transforms a high dimensional observation space into a lower dimension subspace by modifying the original variables which may be correlated into a smaller set of possibly uncorrelated variables. These models are –to some degree– robust to illumination changes and outliers. Figure 2.6(b) illustrates a PCA[4]-based subspace representation. In this figure several eigenbasis, shown in the bottom of the figure, have been learned on-line to represent the target object.

(b) **Feature-based**

These models represent the target appearance based on several features extracted from the object image. Feature-based methods are also categorized into part-wise and target-wise approaches.

Part-wise methods extract different features such as corner, edge, texture, and interest points from a small patch or even a pixel inside the object region.

i. **Texture**
Different texture analysis such as LBP[5][100] shown in Figure 2.7 have been introduced to represent the target appearance. Generally targets with high contrast to the background or fairly well textures can be suitably mod-

[4]Principal Component Analysis
[5]Local Binary Pattern

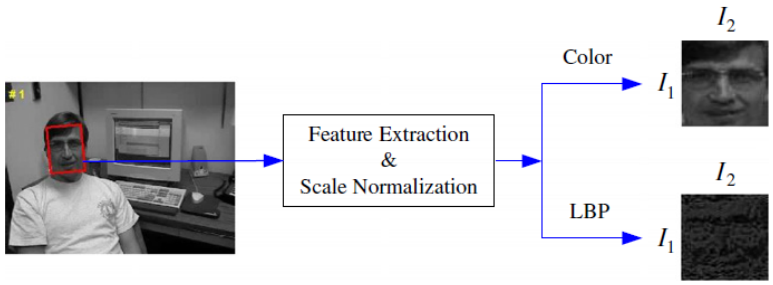


Figure 2.7: Target representation by color and LBP as a texture [96]

eled and tracked based on its color or gray-scale texture.

ii. **Interest points and local invariant features**

Figure 2.8: SIFT, a local invariant feature extraction method used to represent objects [58]

Objects specially with variable shape and structure can be represented by different interest points such as corners or high curvature points. Also, local invariant features such as SIFT[58], SURF[10], and Haar-like features[94] have been widely used to extract invariant and significant features from an image.

Target-wise representation techniques model the target based on global features such as histogram or density distribution.

i. **Histogram**

Figure 2.9: Color-histogram representation (bottom row) for two different object boundaries (top row) [27]

The object appearance can be represented using different histograms including gray-scale, color, or gradient histograms. In general, histograms only model the visual features but not the spatial features. This model can work properly when there is a high color or gray-scale contrast between the object and the background. In Figure 2.9, the color-histogram of a target object is illustrated. Based on this figure, the color-histogram values can be related to the object boundary in the image.

ii. **Feature Set**

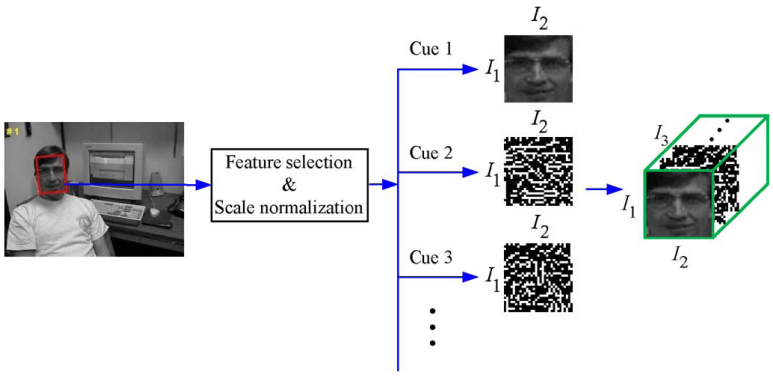


Figure 2.10: Object representation by a set of different features [97]

A combination of different features can be used to present an object. Multi-cue representation will usually increase the modeling robustness against illumination variation, noise, and outliers. In Figure 2.10, different methods such as texture analysis or local feature extractors used to model the object.

2.1.2 Motion model

Dynamic or motion model relates the location of the target object over sequential images. Thus, the next location of the target object can be predicted based on the current location and the motion model.

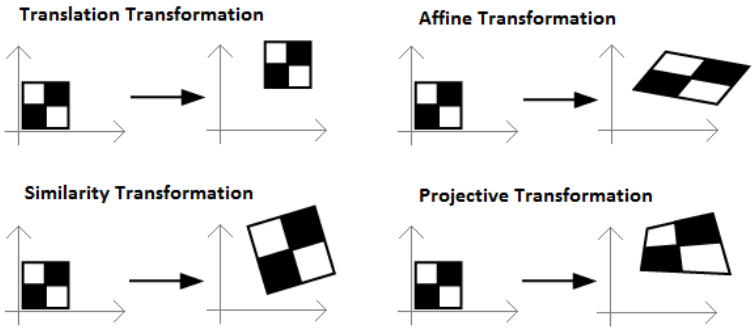


Figure 2.11: Different transformations used to model object motion: translation (top left), similarity (bottom left), affine (top right), projective (bottom right)

Different transformation functions shown in Figure 2.11 have been used to model the object motion between sequential image frames. While translation or similarity transformations can be suitable to estimate the motion of usually rigid and simple objects with less parameters, the more generic models such as projective or affine transformations are employed to model the dynamic of non-rigid and complex targets with the cost of higher processing and convergence time.

2.1.3 Similarity Measure

An important factor to localize the target object in new images is how to measure the similarity of the representation model in comparison with candidate sub-images. In general, both accuracy and robustness of a localization

method –and accordingly, a tracking method– is considerably affected by the similarity measure.

In the literature, different probabilistic and heuristic methods have been introduced to estimate the observation likelihood or similarity to the object model. The sum of squared differences (SSD) has been widely used in early image-based trackers [59] to estimate the similarity between the reference and the candidate images. As SSD is not robust against illumination changes, noise, and specifically outliers, m-estimators [12] have been introduced for robust visual tracking. However, the performance of m-estimators is noticeably sensitive to the algorithm parameters, large illumination changes, and the number of estimating parameters. Recently other metrics such as cross cumulative residual entropy (CCRE) [95], mutual information (MI) [23], and sum of conditional variance (SCV) [78] have been proposed to measure the similarity level between the object model and the candidate sub-images.

2.1.4 Localization and Tracking Method

Object localization method yields the most probable location of the target object at every image frame. In general, the target location is estimated by either a *search strategy* or *filtering algorithm*. The main difference between the two is that in the former, the object is located by searching for an image region similar to the representation model within a close neighborhood around the previous location. However, a typical filtering algorithm evaluates different hypotheses based on the target dynamics to estimate the most probable object's state (e.g., location) in the current image frame. Depending on the application, these two approaches can be combined with different priorities. For example, for tracking [24] faces in a crowded environment, a search method is more suitable than a filtering algorithm due to the fact that the object representation model is more reliable than predicting the target dynamics. On the other hand, for the applications where the target motion is reliable and predictable such as aerial video surveillance [53], a filtering algorithm is mostly used for tracking.

Search Methods

Object localization task can be viewed as the problem of searching the current image frame I_k for the best match of the target state x_k (e.g., location). Assume A_{k-1} is the object representation model learned from images up to time step $k-1$, the state x_k at time step k is calculated by minimizing

a cost function Q.

$$x_k = \arg\min_{x^i} Q(A_{k-1}, I_k(x^i)) \tag{2.1}$$

where $I_k(x^i)$ is a candidate sub-image at location x^i.

A gradient-based optimization such as Gradient decent can be used to recursively find x_k in Eq. 2.1. Let J and H be the first (Gradient) and second (Hessian) derivative of the smooth function Q respectively, and x^* is a minimum of Q (i.e., $J(x^*) = 0$).

$$J(x) = \frac{\partial Q}{\partial x}, H(x) = \frac{\partial^2 Q}{\partial x^2} \tag{2.2}$$

From Taylor expansion, the quadratic approximation of Q can be obtained as:

$$Q(x^i) = Q(x^*) + \frac{1}{2}(x^i - x^*)^T H(x^*)(x^i - x^*) \tag{2.3}$$

One solution for Eq. 2.1 is to iteratively update x^i in negative gradient direction to reach to the optimum point (i.e., x^*).

$$x^{i+1} = x^i - \beta J(x^i) \tag{2.4}$$

where β is a small number to ensure that the algorithm converges to a local minimum.

Newton's algorithm can be also used for faster convergence.

$$x^{i+1} = x^i - H^{-1}(x^i)J(x^i) \tag{2.5}$$

However, calculating the inverse Hessian H^{-1} can be computationally expensive and not-practical in many cases. Moreover, for non-convex cost functions, the gradient-based search may not reach the global minimum point (i.e., the optimal solution), also, the result is considerably sensitive to the initial condition of the optimization process (i.e., $x^{'} = x^0$).

Another solution for Eq. 2.1 is to use an exhaustive search, evolutionary methods such as Genetic Algorithm, or heuristic algorithms.

Filtering Algorithms

Target localization and tracking can be viewed as a process to solve the state-space problem in discrete-time dynamic systems using noisy measurements [9]. Shown in Figure 2.12, a first-order Markov chain can be used for the target state.

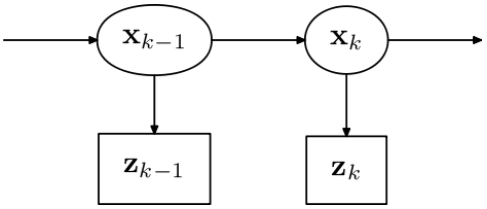


Figure 2.12: Visual Tracking can be viewed as a first-order Markov chain of latent variables $\{x_i\}_{i=1,..,k}$ with corresponding noisy observations $\{z_i\}_{i=1,..,k}$

Under Markovian assumption, the state at time step k (i.e., x_k) only depend on the previous state (i.e., x_{k-1}), also, the observation at time step k (i.e., z_k) depend only on the state at time step k (i.e., x_k). Therefore, we have:

$$p(x_k|x_0,..,x_{k-1}) = p(x_k|x_{k-1}) \tag{2.6}$$
$$p(z_k|x_0,..,x_k,z_1,..,z_{k-1}) = p(z_k|x_k) \tag{2.7}$$

In this context, the state-space dynamic equation can be defined by a non-linear, time-dependent, vector-valued function $f_k : \Re^{n_x} \times \Re^{n_v} \rightarrow \Re^{n_x}$ such that:

$$x_k = f_k(x_{k-1}, v_k) \tag{2.8}$$

where x_k, v_k are the state and process noise vectors at time step $k \in \mathbf{N}$[6], and n_x, n_v are the dimension of the state space and process noise vectors respectively.

The measurement equation is also defined by function $h_k : \Re^{n_x} \times \Re^{n_n} \rightarrow \Re^{n_z}$ with the same properties as that of f_k as follow:

$$z_k = h_k(x_k, n_k) \tag{2.9}$$

where z_k, n_k are the observation and measurement noise vector at time step $k \in \mathbf{N}$, and n_z, n_n are the dimension of the observation and measurement noise vectors at time step k. It is noted that n_k and v_k are independent and identically distributed (i.i.d.).

Given all noisy observations $Z_{1:k} = \{z_i, i = 1,..,k\}$ up to time step k, the main goal is to estimate the state vector x_k. In theory the optimal

[6] $\mathbf{N}$ is the set of natural numbers

solution can be found by the Bayesian Filtering (BF). BF is a recursive two-step (i.e., prediction and update steps) process which approximates the probability density function (pdf) $p(x_k|Z_{1:k})$.

In the prediction or estimation step, the prior pdf $p(x_k|Z_{1:k-1})$ at time step k is calculated using the Chapman-Kolmogorov equation and based on the observation data up to time step $k-1$, i.e., $Z_{1:k-1}$.

$$p(x_k|Z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Z_{1:k-1})dx_{k-1} \tag{2.10}$$

where $p(x_k|x_{k-1})$ is the state transition pdf and $p(x_{k-1}|Z_{1:k-1})$ is the previous conditional state pdf which is known from time step $k-1$. Note that the initial pdf, known as a prior, $p(x_0|z_0) \equiv p(x_0)$ is available to us, and also the transition probability can be obtained based on the process model 2.8.

Accordingly in the second step, the update or correction step, the observation likelihood function $p(z_k|x_k)$ is used to update the posterior pdf $p(x_k|Z_{1:k})$ at time step k via the Bayes rule.

$$p(x_k|Z_{1:k}) = \frac{p(z_k|x_k)p(x_k|Z_{1:k-1})}{p(z_k|Z_{1:k-1})} \tag{2.11}$$

where $p(z_k|Z_{1:k-1})$ is the normalization factor which can be calculated as:

$$p(z_k|Z_{1:k-1}) = \int p(z_k|x_k)p(x_k|Z_{1:k-1})dx_k \tag{2.12}$$

Based on the functions f_k,h_k and noise models, different filtering methods have been proposed. In its simplest form, when the dynamic and measurement equations are linear and the disturbance is white noise (i.e. zero mean Gaussian function) the optimal solution can be found by the Kalman Filter (KF) [9]. In KF, all density functions are Gaussian distribution. If the functions f_k,h_k are nonlinear but the noise is Gaussian, the Extended Kalman Filter (EKM) [9] and Unscented Kalman Filter (UKF) [46] can be used to estimate the posterior pdf which is still modeled as Gaussian. Unlike EKM, UKF uses a parametric model to estimate the mean and covariance of the posterior pdf using a set of discretely sampled points. In the case that the state space is composed of a discrete and finite set of states, the tracking problem can be solved by the Hidden Markov Model (HMM) [73]. In the most general case, i.e. the functions f_k,h_k can be nonlinear and there is no assumption on prior and posterior distribution functions, the problem can be solved based on a sequential Monte Carlo method such as the Particle Filter (PF) [50] (also called Bootstrap Filter [35]). In PF, the prior pdf is modeled by a set of random samples with different importance weights and

the posterior pdf is approximated based on these samples and associated weights (see [5, 25] for reviews).

Table 2.1: Summary of different filtering methods

Method	**State Space**	**Equations**	**Noise Model**	**Posterior pdf**
KF	Continuous	Linear	Gaussian	Gaussian
EKF	Continuous	Nonlinear	Gaussian	Gaussian
UKF	Continuous	Nonlinear	Gaussian	Parametrized Gaussian
PF	Continuous	Nonlinear	Non-Gaussian	Non-Gaussian

Chapter 3

Robust Decentralized Multi-Model Adaptive Template Tracking

Visual tracking is defined as the problem of localizing and corresponding a target of interest in an image sequence. This is an open topic in the field of computer vision and robotics mainly because of complex and unpredicted changes in target appearance and nature of noisy images. As I discussed in Chapter 2, a typical visual tracker has four important components i.e., 1) target representation, 2) motion model, 3) similarity measure, and 4) localization and tracking method. Among these components, target representation plays a crucial role in forming a tracking algorithm. There are two different approaches for representing the target i.e., region-based and feature-based. Feature-based models can be robust to appearance, shape, and scale variations, however, they are required to extract and match specific features between consecutive images. First of all, selecting a discriminative feature set can be an important task which in most cases has to be done by the designer depending on different applications. In addition, popular feature extraction algorithms are usually computationally expensive which practically limits their usage in many real-world applications such as vehicle navigation and tracking. Moreover, feature-based methods rise another challenging topic i.e., Data Association[7] to this problem which makes the tracking algorithm even more complicated. On the other hand, region-based methods employ all information obtained from the target image region to generate a representation model. Conventional region-based trackers such as Lucas-Kanade optical flow algorithm [59] are fairly simple and fast, however, they are not considered as robust methods mainly due to the fact that they cannot properly model image noise, occlusion, and non-rigidity and variations in the target appearance. Region-based trackers usually drift from the target over time either because of updating the representation model

[7]Refer to [9] for a review of different Data Association algorithms

from wrong information e.g., background pixel values or by not updating the target model.

In this chapter, I propose a robust region-based tracking method based on a decentralized target representation model and a parallel search for localizing and tracking the target within sequential images. The proposed tracker consists of a multi-initializing points EM[8]-like optimization algorithm and multiple heterogeneous adaptive templates. This method is capable of tracking non-rigid objects with variable appearance, shape, scale, and unpredicted motion in cluttered environments. It is assumed that the target object has been located manually or automatically (by any existing object detection method[9]) at the first frame and the goal is to adaptively track the target object without any prior knowledge about the representation model or the object motion pattern. Note that the target of interest can generally be the whole or part(s) of a real-world object (e.g., a human, face, or car) and cannot have a chaotic or huge movement between two consecutive images, whereas, the camera may move independently in any direction.

Considering two major object localization and tracking approaches (i.e., search methods and filtering algorithms) mentioned in Section 2.1.4, the focus of the method proposed in this chapter is more on a gradient-based optimization method using multiple adaptive templates rather than a sampling algorithm which estimates the posterior probability distribution of the object location based on the observation likelihood function. In fact, for tracking of real-world and specially non-rigid objects, target localization based on shape and appearance adaptation can be more reliable and informative than the target dynamic modeling and motion estimation. In the case of tracking with no information about the motion dynamics (although dynamic model of the target can be estimated over time, it is not always reliable due to the unpredicted and complex target and camera motion), localization and tracking based on the appearance and shape changes play a crucial role in developing a robust method.

In the following sections, first in Section 3.1 some relevant visual trackers are reviewed, and then, in Section 3.2 the template matching problem is defined in detail and a formal definition of template-based visual tracking is presented. The details of the proposed robust decentralized template-based tracking method are explained in Section 3.3 where the object representation

[8]Expectation Maximization, see [65] for more information

[9]The main role of the detector is to locate the target in the first image frame, however, in this research it is assumed that it is not feasible to train a detector in advance because of not having the training data, therefore, the target is usually located by human interaction at the beginning of tracking

model and decentralized localization are proposed. Next, in Section 3.4 the proposed tracker has been applied on several challenging videos and the results have been compared with four state-of-the-art methods as well as manually labeled ground truth data. Some conclusions and potential extensions for future work are provided in Section 3.5.

3.1 Related Work

As an intuitive method, silhouette-based tracking can be used to track isolated and non-rigid objects such as humans. In a simple environment with a fixed camera, silhouettes and counters are robust to non-rigidity, appearance changes, and shape variations. However, they only model the boundary of the objects and do not represent the whole object region. These methods are very unstable to occlusion and scale variations especially in cluttered and dynamic environments. Adaptive background subtraction and modeling [64, 88] are the common methods for silhouette-based tracking where the camera is fixed. Also several shadow removal techniques (e.g., [54]) have been proposed to improve the accuracy of silhouette tracking against illumination changes. However, most of them are task-dependent and cannot be used in a general case. Chen et al. [16] proposed a contour-based object tracking method based on Hidden Markov Model (HMM) framework where the transitions probabilities are estimated by the Joint Probability Data Association Filter (JPDAF). Although this tracker can use multiple cues and handle the target appearance changes, it is unstable to the unpredicted and complex target motion specially in a cluttered environment. Ponsa and López [72] proposed a particle-based contour tracking method. They used Particle Filtering algorithm to model and track contours. Although it is robust to shape variations and noise, the proposed method cannot track objects with complex appearance in cluttered and crowded environments.

In contrast to silhouette-based trackers, histogram-based trackers are robust to the appearance non-rigidity, outliers, partial occlusion, and to some degree scale and shape deformations. But histograms only encode the content information insider the target object and they do not consider the target spatial properties such as shape and structure. Also histogram adaptation usually suffers from the "drift" problem. Comaniciu et al. [20], [21] proposed a mean-shift based non-rigid object tracking method. They used the Bhattacharyya coefficient to measure the similarity of the target model (i.e., color distribution) and the possible location of the target in the current image. The object location is estimated by maximizing the Bhat-

tacharyya coefficient which is closely related to the Bayes error between the target model and candidate density distributions. This method, to some degree, can handle small appearance changes and partial occlusion. However, the proposed method cannot handle illumination changes and unpredicted object motion. It is also unstable in cluttered scenes and dynamic environments. Kr et al. [51] extended the mean-shift procedure to find the position of a local mode of a density function as well as the covariance matrix for estimating the local mode shape. The covariance matrix is used for scale and shape adaptation. However, the algorithm is not robust to multiple targets and cluttered environment. It also cannot handle rapid motion and appearance changes. Zhou et al. [101] optimized the performance of the method proposed by Kr et al. [51] in complex scenes by optimally adapting an ellipse outlining the target object. They optimized a new cost function to track non-rigid objects. The new cost function is composed of two terms; (1) the standard least squared error, (2) a regulator that dynamically changes the error between the estimated pdf and the expected one. Although the proposed algorithm can adapt a better ellipse to scale and shape changes, it has more estimation parameters to be defined a priori and also, it is more time consuming than the previous work by [51]. As another shortcoming, the proposed method is not stable to partial occlusion. Adam et al. [1] used multiple image patches for visual object tracking. These patches are selected arbitrary and the next object location and scale are obtained by statistically combining the patches' votes. They used the integral histogram data structure for efficiently computing the histogram of multiple patches. They claimed that using multiple patches improves the robustness of the algorithm against partial occlusion and pose changes. Besides the color histogram, Ning et al. [67] introduced the Mean-Shift algorithm using the joint color-texture features. Their method outperformed the standard Mean-Shift method especially in complex environments. They also extended the Mean-Shift algorithm in [68] by estimating the scale and orientation of the target object. Shan et al. [86] used Mean-Shift algorithm to improve the sampling efficiency of Particle Filtering. The proposed tracking method, called "Mean-Shift embedded Particle Filter", can handle rapid motions using fewer particles than conventional Particle Filtering methods. However, the number of particles is still large for tracking a nonrigid object with complex appearance and shape. Also, this method is not robust to occlusion and cluttered background. Khan et al. [49] integrated the Particle Filter and Mean-Shift to track objects in complex scenes. The tracking problem is solved using some independent trackers. Despite the previous Mean-Shift methods, they used a multi-mode anisotropic Mean-Shift via partitioning

the object bounding box. The tracking task is performed by estimating the target shape using the Particle Filter algorithm and managing the appearance dynamics by the embedded Mean-Shift. However, the proposed method needs several parameters to be adjusted carefully, it is sensitive to large image changes, and it is computationally expensive specially for objects with large size.

In addition to the above methods, subspace representation models are successful in handling the small appearance variations and illumination changes. They are also stable to partial occlusion and outliers. However, they usually fail in handling rapid appearance, shape, and scale changes and they are also unstable to appearance non-rigidity and long-term occlusion. Ross et al. [82] developed an object tracking method capable of incremental learning and online updating a low-dimensional PCA subspace representation of the target object. They used a Particle Filter algorithm to approximate the target location in sequential images. In this work, the visual tracking task is viewed as a sequential inference task in a Markov model with hidden state variables describing the target motion parameters at a specific time instant. The proposed method can learn and update the target appearance over time. However, similar to the other methods based on Particle Filter; a sufficient number of particles is needed to appropriately approximate the posterior distribution (in fact a large number of particles can dramatically increase the cost of computations). The authors mentioned that the proposed tracker occasionally drifts from the target object. Also the presented dynamical model which is based on the Gaussian distribution is not a valid model for all applications. Gai and Stevenson [33] used the Student t-distribution based PCA within a Dynamical System (DS) framework to improve the tracking robustness against outliers. They claimed that the original Probabilistic PCA (i.e., Gaussian-based subspace representation) is not fully compatible with object appearance modeling. However, the robustness of the proposed observation distribution modeling to outliers is sensitive to the different probability density of the auxiliary variables which are required to be chosen beforehand by the designer. Wang et al. [96] extracted different cues for on-line learning of multiple appearance models to represent the target. These models are then fused and used as the observation model within a Bayesian inference framework where a Particle Filter method is used to estimate the target state. The proposed tracker can handle target appearance changes and is robust to illumination changes due to the use of color and LBP cues. It is also stable in the presence of short-term partial occlusion and very small targets. However, this method cannot track targets moving in unpredicted patterns (i.e., a common problem for

Particle Filter based methods).

Similar to the subspace representation models, deformable templates can be used to handle small appearance, shape, and scale changes. However, they are not generally robust to appearance non-rigidity, occlusion, and outliers. In the following sections, the proposed robust template based method followed by a formal definition of the template matching problem is presented.

3.2 Template Matching

Template matching is a well-studied computer vision problem which was first introduced by Lucas and Kanade [59] optical flow algorithm for the task of visual tracking. In the optical flow or image alignment algorithms, an image patch is specified in the first frame as a template and then the task is to find the best match to the template in the following image frames. In [59], the target image region is considered as the template $T(X)$ where $X = (x, y)$ is the pixel coordinates, and the goal is to find the best corresponding match in the next image I_n based on the target dynamical model $W(X; P)$ where $P = \{p_1, .., p_k\}$ are the template transformation parameters. The sum of squared difference (SSD) between the template and the next image can be the similarity measure to find the best match.

$$P_n = \arg\min_P \sum_{X \in T} [I_n(W(X; P)) - T(X)]^2 \tag{3.1}$$

A nonlinear optimization algorithm to solve Eq. 3.1 was introduced in [59]. Since the first template tracking method, the accuracy and efficiency of the template tracking has been improved in different ways using a more general template transformation [11], linear appearance variation [12, 38], real-time implementation [38, 61], Active Appearance Models (AAMs) to model non-rigid appearance [22, 76]. Vasconcelos and Tavares [92] used AAM to segment objects based on their models. They proposed a method to automatically extract significant object feature points and build point distribution models. Georgescu et al. [34] proposed a method that models the target appearance variations by maintaining several templates during the time. The target appearance is divided into several regions named components which are tracked separately. The proposed method is suitable for shape tracking due to multiple component models and can handle partial occlusion and appearance changes. However, it is computationally expensive because of having multiple components and many optimization steps.

Based on the experimental results, this method is mostly suitable for rigid object tracking and also the authors mentioned that global motion estimation is required as well. Matthews et al. [62] proposed a visual tracking method based on an adaptive appearance template which does not suffer from the "drift" problem. They introduced a template update strategy with drift correction. To some degree, the tracking method is stable to the local minimum by reinitializing the gradient search based on the naive template update strategy. However, the proposed method cannot handle occlusion and it fails when it tracks non-rigid objects specially when the object shape is changing over time. Silveira and Malis [87] used several image transformation models for optimizing a template based tracking method. They proposed a new illumination model which can be used to track a deformable target with illumination change. In this method, a general image formation model which covers both geometric and photometric deformations is defined to track a rigid or deformable object. Also, a nonlinear gradient-based optimization is used to estimate the geometric and photometric transformation parameters. Although the proposed method is robust to illumination change and general object deformation, it cannot handle large and unpredicted pose and appearance changes due to the gradient-based optimization and large number of parameters being estimated. Also the tracking method is unable to track non-rigid objects with variant shape and structure. Another problem of this method is related to the target image content; this method is not stable when the target image is not sufficiently textured.

3.3 Decentralized Template Tracking

Given the object bounding box $B_o = \{left_o, top_o, right_o, bottom_o\}$ at the first image frame I^1, the visual tracking can be defined as the problem of finding the object bounding box $\{B_o^t\}_{t=2:t_c}$ in a set of sequential images $\{I^t\}_{t=2:t_c}$ up to the current time instant t_c. In this chapter, a novel decentralized multi-templates tracking method inspired by the works in [34, 62] is presented to solve the above problem. In the proposed method, the object image region is partitioned into several non-overlapping subregions. Partitioning the object region into small subregions has several advantages. First, it can model real-world objects with different shapes and formations by relating each subregion to a specific object part. Moreover, it provides a suitable framework for handling the shape and formation variations by considering subregions both individually and as a group of relatives. In addition, multiple partitions can improve the accuracy and robustness of the

tracking. Since each subregion is tracked independently, the fusion of individual trackers can decrease the uncertainty of the object localization and therefore, increase the accuracy of the whole object tracker. Besides managing non-rigidity and shape variations, multiple subregions can be used for scale adaptation by changing the formation and distribution of the subregions to handle different object scales. For example, the target bounding box becomes smaller (i.e., down scaling) if some of the non-overlapped subregions become overlapping in the next image frame. In the proposed method, each subregion is modeled by two adaptive templates called immediate and delayed templates to tackle the different time-varying appearance changes. The former handles short-term appearance changes, and the latter models the long-term variations. The combination of short-term and long-term representation modeling can improve the accuracy and robustness of tracking especially against rapid appearance and shape changes, noise, outliers, and occlusion. A gradient-based search with multiple initial points is also used as the localization method to handle unpredicted and complex subregion motions. More precisely, the localization method is an EM-like algorithm capable of minimizing a mixture of Gaussian error functions between the template and the candidate sub-image. In this algorithm, each Gaussian error function represents a possible solution for the localization problem. Thus, the best location is found by considering all possible locations in a single optimization process. As another advantage, the use of multiple initializing points decreases the probability of being trapped in local minima which is a common problem of gradient-based search methods. In addition to the multiple initializing points, a two-step template-matching optimization is used to improve the localization accuracy and robustness. First the EM-like optimization algorithm is performed using the immediate template to handle rapid appearance and shape changes. Then the best location found is employed to initiate the second optimization process using the delayed template. In fact, the second optimization process can be viewed as a fine tuning step which re-localizes the subregion more precisely and it also solves the "drift" problem (a common problem of template-based tracking methods [62]). A summary of the proposed tracking algorithm is shown in Algorithm 2. In the following subsections, different parts of this algorithm are explained in detail.

3.3.1 Object Representation Model

The proposed object representation model consists of several adaptive heterogeneous templates which are distributed within the object region. At

the first image frame I^1, the target object region is partitioned into the grid cells named subregions $S = \{s_i\}_{i=1:N_S}$. Each subregion s_i is defined by a bounding box $B_i = \{top_i, left_i, right_i, bottom_i\}$ and two different adaptive templates i.e., immediate T_{i_M} and delayed T_{i_E} templates. The first template models the short-term appearance variations whereas the second one encodes the long-term appearance changes which is in fact necessary to solve the "drift" problem. Each template T_i is defined by a mean matrix $\mu_i = [\mu_i(x,y)]_{x,y \in B_i}$ and variance matrix $\sigma_i = [\sigma_i(x,y)]_{x,y \in B_i}$ consisting of the mean and the variance values of each point inside the subregion's bounding box, respectively. These templates are initialized and updated according to the following subsections.

Template Initialization

Subregion templates (i.e., T_{i_M} and T_{i_E}) are initialized from the first image I_1 specified by the subregion bounding box B_i.

$$\mu_{i_M} = \mu_{i_E} = I^1\{left_i : right_i, top_i : bottom_i\} \tag{3.2}$$

$$\sigma_{i_M} = \sigma_{i_E} = \{1\}_{width_i \times height_i} \tag{3.3}$$

where $I^1\{left_i : right_i, top_i : bottom_i\}$ is a sub-image specified by the subregion bounding box B_i, $width_i = right_i - left_i + 1$, $height_i = bottom_i - top_i + 1$, and $\{1\}$ is a matrix of "1"s.

Template Updating

Given the subregion bounding box B_i^t in the current image frame I^t, the templates are updated based on the following equations.

$$\begin{aligned} \hat{e_i} &= I^t\{left_i^t : right_i^t, top_i^t : bottom_i^t\} - \mu_i^{t-1} \\ \mu_i^t(x,y) &= (1-\alpha_\mu)\mu_i^{t-1}(x,y) + \alpha_\mu \hat{e_i}(x,y) \ ; \ (x,y) \in B_i^t \\ \sigma_i^t(x,y) &= \frac{1}{t}\left[(t-1)\sigma_i^{t-1}(x,y) + \hat{e_i}(x,y)^2\right] \ ; \ (x,y) \in B_i^t \end{aligned} \tag{3.4}$$

where $\hat{e_i}$ is the updating residual error matrix and $\alpha_\mu \in [0,1]$ is the template updating rate which is predefined based on the template type. For the immediate template, the parameter α_μ is set to a high value (e.g., 0.8), whereas this value is small for the delayed template such as 0.02. These templates represent different time-varying appearance variations i.e., short-term and long-term changes respectively. This property of the proposed object representation model improves the tracking robustness to the both fast and slow appearance changes as well as the "drift" problem.

3.3.2 Subregion Localization

As mentioned in Section 2.1.4, the localization and tracking task is categorized into two different approaches: (1) search or optimization methods and (2) filtering algorithms. While the latter focuses more on estimating the target motion dynamics, the former locates the target based on the object appearance and shape. In this section, an EM-like gradient-based optimization algorithm with multiple initializing points is proposed to locate the target object based on the appearance and shape. In the following subsections, for the sake of clearance the time instant is not specified within the equations.

Localization Problem

Given the template $T = \{\mu, \sigma\}_{w \times h}$ where w, h are the width and height of the corresponding bounding box B, and the candidate sub-image $F = I\{W(B)\}$ where W is the transformation function, the ultimate goal is to find the best values of W in a way that the candidate sub-image is the best match for the template. This problem can be solved by optimizing the sum of Gaussian errors between the template and the candidate sub-image. In the general case, the matrix W can be defined by an affine transformation map similar to the one used in [55]. However, the best suitable transformation matrix should be defined based on the object representation model. Although a generic transformation matrix with more parameters (e.g., affine transformation) seems to be more reliable for finding the new object location, a large number of unknown parameters may increase the uncertainty and the search space dramatically (indeed estimating a large number of unknown parameters is impractical for many real-time applications). Considering the spatial distribution of subregions defined in the proposed object representation model (see Section 3.3.1), one can assume that each subregion is related to a small part of the object and also, between two consecutive low-interval image frames, this small part can be viewed as a rigid object which has been moved to a new location close to the previous one. In this work, it is assumed that the subregion movement is only made of translation transformation. In other words, $W(X) = X + \delta X$. Although this small movement may consist of other transformations such as rotation, scale, and perspective, these small transformations can be implicitly modeled by the multiple adaptive heterogeneous templates. This assumption has been also verified from the several experiments discussed in Section 4.4.

Considering the subregion movement as a translation, I define the sum

of Gaussian errors (SGE) based on the following equation.

$$SGE = \sum_{(x,y)\in B} \exp\left(-\frac{1}{2}\left(\frac{\mu(x,y) - I(x+\delta x, y+\delta y)}{\sigma(x,y)}\right)^2\right) \tag{3.5}$$

The best values of δx and δy can be found by optimizing Eq. 3.5.

$$(\delta x^*, \delta y^*) = \arg\max_{(\delta x, \delta y)} SGE \tag{3.6}$$

To solve the optimization problem defined in Eq. 3.6, in this work, a multi-initializing points EM-like algorithm is proposed. In addition to the fact that EM is a powerful optimization algorithm which can manage incomplete data in a Bayesian framework, different starting points increases the robustness of the optimization algorithm against local optimum and outliers. In the following subsections, the proposed optimization algorithm is explained in detail.

Formulation

If for each point, I define: $X = \{x, y\} \in [1..w, 1..h]$, $\delta X = \{\delta x, \delta y\}$, $\mu_X = \mu(x, y)$, $\sigma_X = \sigma(x, y)$, $F_X = F(X)$, and $F_{\delta X} = F(X + \delta X)$, the conditional error density function of each point can be modeled by a mixture of Gaussian error functions based on Eq. 3.7.

$$P(X|\Theta) = \sum_{l=1}^{L} \alpha_l P_l(X|\theta_l) \ , \ \sum_{l=1}^{L} \alpha_l = 1 \tag{3.7}$$

where $\{P_l(X|\theta_l)\}_{l=1..L}$ and $\{\alpha_l\}_{l=1..L}$ are the Gaussian error functions and the contributing weights respectively for L Gaussians. Also $\Theta = \{\alpha_l, \theta_l\}_{l=1:L}$ and $\theta_l = \delta X_l = \{\delta x_l, \delta y_l\}$. Based on the above notations, the Gaussian error function is defined as:

$$P_l(X|\theta_l) = \exp\left(-\frac{1}{2}\left(\frac{\mu_X - F_{\delta X_l}}{\sigma_X}\right)^2\right) \tag{3.8}$$

Also based on the Bayes's rule we obtain:

$$P(l|X, \Theta) = \frac{\alpha_l P_l(X|\theta_l)}{\sum_{k=1}^{L} \alpha_k P_k(X|\theta_k)} \tag{3.9}$$

Similar to the EM algorithm used in [77] for estimating a mixture density model, the EM problem can be viewed as maximizing the following function.

$$Q(\Theta, \Theta^g) = \sum_{l=1}^{L} \sum_{X \in B} \left(\log(\alpha_l) + \log\left(P_l(X|\theta_l)\right)\right) P(l|X, \Theta^g) \tag{3.10}$$

where Θ^g is the value of the EM parameters obtained from the previous optimization iteration.

Taking the derivative of Eq. 3.10 with respect to α_l and considering the constraint $\sum_{l=1}^{L} \alpha_l = 1$, we obtain:

$$\frac{\partial Q(\Theta, \Theta^g)}{\partial \alpha_l} = 0 \rightarrow \alpha_l = \frac{1}{N_B} \sum_{X \in B} P(l|X, \Theta^g) \tag{3.11}$$

where N_B is the number of points inside the subregion bounding box B.

Substituting the $F_{\delta X_l}$ with its linear approximation around the previous parameters' values δX_l^g in Eq.3.8 and taking the derivative of Eq. 3.10 with respect to δX_l, we obtain:

$$\frac{\partial}{\partial \delta X_l} \left\{ \sum_{X \in B} \left(-\frac{\left(\mu_X - F_{\delta X_l^g} - (\delta X_l - \delta X_l^g) F'_{\delta X_l^g}\right)^2}{2\sigma_X^2} \right) P(l|X, \Theta^g) \right\} = 0 \tag{3.12}$$

Therefore the δX_l is computed as the following equation:

$$\delta X_l = \frac{\sum_{X \in B} \sigma_X^{-2} \left[\mu_X - F_{\delta X_l^g} + \delta X_l^g F'_{\delta X_l^g}\right] F'_{\delta X_l^g} P(l|X, \Theta^g)}{\sum_{X \in B} \sigma_X^{-2} \left(F'_{\delta X_l^g}\right)^2 P(l|X, \Theta^g)} \tag{3.13}$$

where $F'_{\delta X_l^g}$ is the first derivative of the $F_{\delta X_l^g}$.

Algorithm

Based on the formulation provided in the previous section, the best location of the subregion in the current image frame is found by initializing the $\{\delta X_l\}_{l=1..L}$ with several possible subregion movements (e.g., small movement in every 90 degrees) and then the EM parameters α_l and δX_l are iteratively estimated based on Eq. 3.11 and Eq. 3.13 to reach the best solution. The optimization procedure is discontinued either if it has been performed for a specific number of iterations or the difference between two consecutive values of parameter δX_l is less than a small value. After finding the best value

of each δX_l, the one with the highest corresponding weight α_l is selected as the best overall solution which is in fact the subregion movement between two sequential image frames. A simplified version of the EM optimization algorithm is presented in Algorithm 1.

Shown in Algorithm 1, simultaneous optimization of the subregion localization problem using possible solutions (i.e., the subregion possible movements) can be different from sequentially performing the optimization algorithm with a different initializing point. Indeed parallel optimization can improve the accuracy and convergence speed of the overall optimization process. At each optimization iteration, the result of each possible solution is compared with the others and only those whose results are better will stay in the optimization process. In other words, a wrong solution will have a proportionally low contributing weight (α_l) and it will be removed from the optimization process before other solutions; therefore, the convergence speed will be increased by having less solutions being optimized. Although a comparatively better solution will be removed from the optimization process before the others as well, still it can modify the contributing weights of the other solutions and reduce the number of iterations. Also in the proposed parallel optimization algorithm, the best solution is found by comparing the potential solutions with each others in a unified probabilistic framework.

Algorithm 1: Subregion localization

1. **Initializing the parameters:**
   (a) $i = 0$. This parameter indicates the number of iterations.
   (b) $valid_l = true$ for $l = 1..L$. This parameter indicates the validity of the l^{th} parameter in the optimization process.
   (c) $\alpha_l = \frac{1}{L}$ for $l = 1..L$
   (d) $\{\theta_l\}_{l=1..L} \equiv \{\delta X_l\}_{l=1..L}$ are initialized by all possible subregion movements. For example they can be the movements in any possible directions.
2. **Computing the distributions**
   (a) Computing the distribution $P_l(X|\theta_l)$ based on Eq. 3.8 for $l = 1..L$ where $valid_l = true$.
   (b) Computing the distribution $P(l|X, \Theta)$ based on Eq. 3.9 for $l = 1..L$. Note that for non-valid parameters, the latest valid estimation of the distribution $P_l(X|\theta_l)$ is used.
3. **Estimating the parameters**
   (a) Estimating α_l based on Eq. 3.11 for $l = 1..L$.
   (b) Estimating δX_l based on Eq. 3.13 for $l = 1..L$ where $valid_l = true$.
4. **Checking the parameters validity**
   (a) $valid_l = false$ if ($\|\delta X_l - \delta X_l^g\| < valid_{thr}$ OR $\alpha_l \cong 0$) for $l = 1..L$. The threshold $valid_{thr}$ is set to a small value (e.g., 0.1) to terminate the optimization of the l^{th} solution when it reaches its extreme value.
   (b) $i = i + 1$, go to 5 if $i > i_{max}$. The parameter i_{max} limits the maximum number of optimization iterations.
   (c) Go to 2 if $\forall l \in [1..L], \quad \exists\, valid_l = true$.
5. **Finding the best solution**
   (a) $\delta X_i^* \equiv \delta X_{l^*}$ is the best estimation of the i^{th} subregion movement and $\alpha_i^* \equiv \alpha_{l^*}$ is the corresponding weight where $\alpha_{l^*} = \max\{\alpha_l\}_{l=1..L}$

3.3.3 Decentralized Object Motion Estimation

After localizing all subregions in the current image frame based on Algorithm 1, subregion movements are used to estimate the location of the object and its new bounding box. For object movement estimation, first a rough estimation of the object movement is estimated, and then all outliers (i.e., invalid subregions) are rejected. The final object location is estimated by aggregating the valid subregion movements and previous object motion. Also the new object bounding box is found only based on the valid locations of the subregions.

Outlier Rejection

In this section a preliminary estimation of the object movement $\delta\hat{X}^t$ at time instant t is used to filter invalid subregion movements. This rough object movement is estimated by weighted averaging of the movements obtained from optimizing all subregion templates (including both T_{i_M} and T_{i_E}), see Eq. 3.14.

$$\delta\hat{X}^t = \frac{1}{2N_S}\sum_{i=1}^{N_S}\left(\alpha^*_{i_M}\delta X^*_{i_M} + \alpha^*_{i_E}\delta X^*_{i_E}\right) \tag{3.14}$$

Based on Eq. 3.15, the subregion movement is considered as an outlier if the Euclidean distance of both the immediate and delayed template movements and the preliminary object movement is greater than a predefined threshold (thr_{Ol}). In fact, the template movement is considered as an outlier when its difference with the weighted average of all movements (i.e., the object preliminary movement) is proportionally significant.

$$if\ \min\left(\|\delta\hat{X}^t - \delta X^*_{i_M}\|, \|\delta\hat{X}^t - \delta X^*_{i_E}\|\right) > thr_{Ol},\ then\ i \in S_{Ol} \tag{3.15}$$

where $i \in [1..N_S]$ and S_{Ol} is the set of outliers.

Object Motion and Bounding Box Estimation

After eliminating the invalid subregions from the object localization step, only the best subregion template movements are used for estimating the final object movement. The best subregion template is the one whose movement has the least difference with the preliminary object movement. Shown in Eq. 3.16, the final object movement is obtained by weighted averaging only

the valid subregion movements which is the best subregion template movements.

$$\delta X^t = \frac{1}{2N_{S'}} \sum_{i \in S'} \left(\alpha^*_{i_B} \delta X^*_{i_B} \right) \; ; \; S' = S - S_{Ol} \tag{3.16}$$

where $\delta X^*_{i_B}$ and $\alpha^*_{i_B}$ are the best template movement and the best template contributing weight, respectively. Also S' is the set of valid subregions and $N_{S'}$ is the number of valid subregions. The new object bounding box B^t_o is also computed by finding the bounding box which contains all valid subregions.

$$\begin{aligned} Left^t_o &= \min \left\{ left^{t-1}_i + \delta x^*_{i_B} \right\}_{i \in S'} \\ right^t_o &= \max \left\{ right^{t-1}_i + \delta x^*_{i_B} \right\}_{i \in S'} \\ top^t_o &= \min \left\{ top^{t-1}_i + \delta y^*_{i_B} \right\}_{i \in S'} \\ bottom^t_o &= \max \left\{ bottom^{t-1}_i + \delta y^*_{i_B} \right\}_{i \in S'} \end{aligned} \tag{3.17}$$

where $B^t_o = \{left^t_o, right^t_o, top^t_o, bottom^t_o\}$ is the new object bounding box.

Subregion Motion Constrain

To improve the robustness of the tracking method against the object appearance, shape, and scale changes as well as noise, all subregions are reconfigured based on the new object bounding box. Indeed, each subregion has a relative location inside the object bounding box ($\hat{B}^1_i$) called "base-location". This relative location is defined at the first image frame, and at every time instant t. The new subregion location is estimated by considering the difference between the original base-location and the new base-location ($\hat{B}^t_i$) as well as the template movement. The subregion new location is obtained based on Eq. 3.18

$$B^t_i = B^{t-1}_i + \left((1-\beta) \delta X^*_{i_B} + \beta (\hat{B}^1_i - \hat{B}^t_i) \right) \tag{3.18}$$

where $\beta \in [0, 1]$ is the coefficient that controls the subregion free movements. If β is close to "1", the subregion new location is restricted by the object bounding box; however, this parameter should be small (e.g., 0.1) for tracking objects with variable appearance, shape, and scale. Based on Eq. 3.4, the subregion templates (i.e., T_{i_M} and T_{i_E}) are then updated from

the sub-image specified by the subregion bounding box B_i^t.

Algorithm 2: Summary of the proposed tracking algorithm

1. Locate the target object bounding box either manually or by an object detection method.
2. Partition the object bounding box into several subregions with a specific size.
3. Base on Eq. 3.2, initialize two templates (immediate and delayed) and assign them to each subregion.
4. For each subregion, use Algorithm 1 to localize the immediate template and then localize the delayed template.
5. Use Equations 3.16, 3.17, and 3.18 to estimate the object motion and accordingly locate the object bounding box and the subregions bounding boxes.
6. Update the subregion templates based on Eq. 3.4.
7. Go to step 4 until the end of image sequences.

3.4 Experimental Results

In this section, three indoor and one outdoor videos have been used to evaluate the empirical performance and robustness of the proposed tracking method against object pose, appearance, and scale changes as well as occlusion and noise. In addition, the target object and camera have large and complex movement. Note that all videos consist of gray-scale images which are scaled to $[0, 30]$ and the algorithm parameters are the same for all of the experiments. It is assumed that the target object bounding box is specified manually (or by an object detector) beforehand. The object is then partitioned into non-overlapping subregions with the size of 22×22 pixels by the proposed tracking method at the first frame. In the following experiments, depending on the target object size, different number of subregions are generated for the proposed decentralized object tracking method. Also the following conditions have been applied in all experiments.

– The updating rate α_μ is set to 0.8 and 0.02 for the immediate and delayed templates respectively, see Eq. 3.4.

- In Algorithm 1,
  - for localizing a subregion by its immediate template, the δX_l values are initialized with 5 possible relative movements which are {(0,0), (5,0), (0,5), (-5,0), (0,-5)}.
  - for localizing a subregion by its delayed template, the δX_l values are initialized with 13 possible relative movements which are {(0,0), (5,0), (0,5), (-5,0), (0,-5), (10,0), (7,7), (0,10), (-7,7), (-10,0), (-7,-7), (0,-10), (7,-7)}. Note that the subregion movement obtained from the immediate template is also added to the suggested possible movements for initializing the delayed template localization.
  - the validity threshold of the localization solution $valid_{thr}$ is set to 0.1.
  - the maximum optimization iterations i_{max} are set to 30 and 10 for the immediate and delayed templates respectively.
- In Eq. 3.15, the outlier threshold thr_{Ol} is set to 3.
- The parameter β in Eq. 3.18 is set to 0.1.

Experiment 1: In the first experiment, a challenging movie studied in [43] has been used. This image sequence is composed of 1145 gray-scale images which are recorded at 30Hz with the size of 360×240. Fig. 3.1 shows the tracking result using the proposed method. The dashed (red) box indicates the object bounding box and the solid (blue) small boxes are the object subregions. Based on Fig. 3.1, the proposed method is robust to track the target object in several challenging situations including: different poses (e.g., 80, 170, 354, 688, 750), scaling (e.g., 4, 296, 688, 795, 1002), illumination changes (e.g., 688, 880), shape deformation (e.g., 363, 367, 458, 795), and temporary occlusion (e.g., 208, 367). In contrast to the typical template tracking methods, the proposed method is robust to the drift problem and can track the target even if the object appearance is changed significantly (e.g., 208).

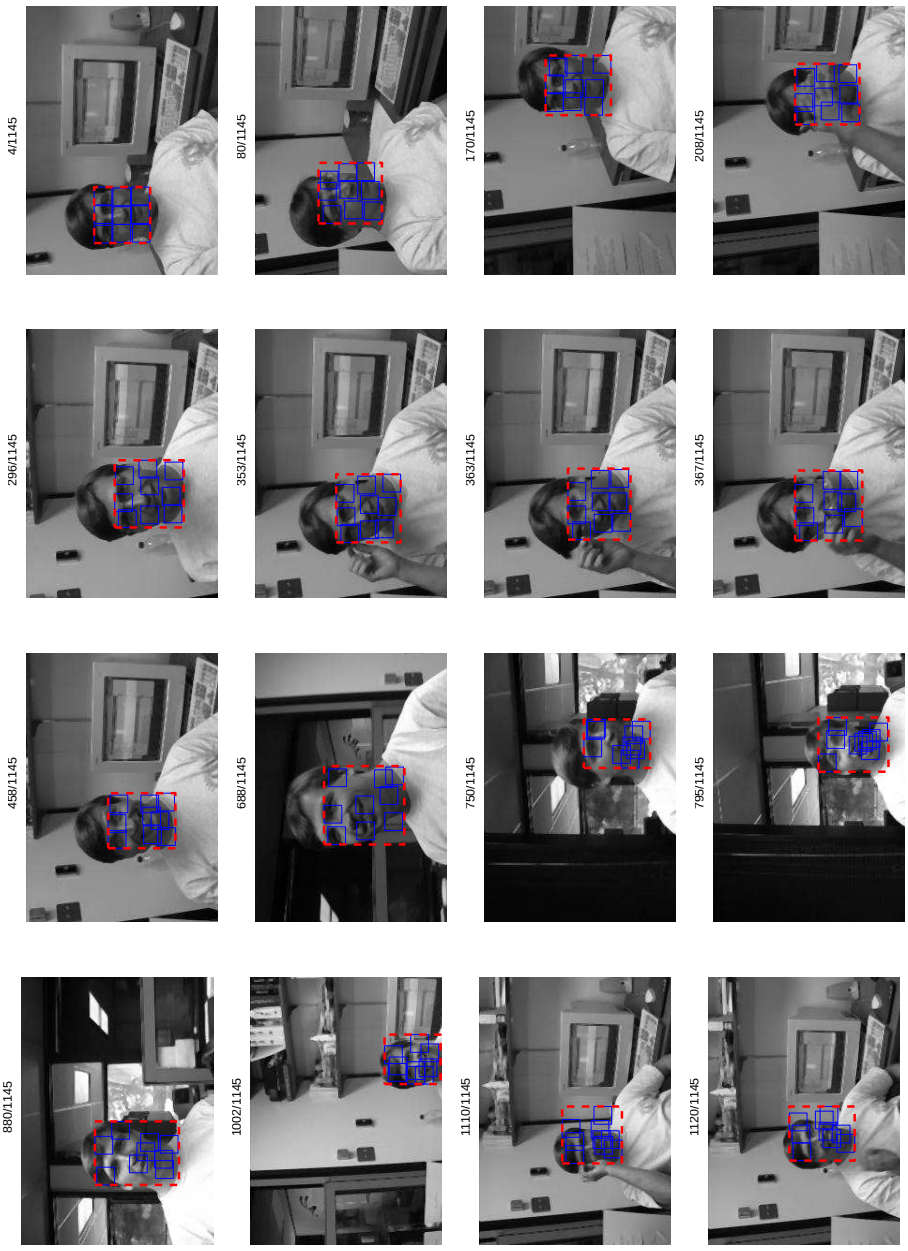


Figure 3.1: A video containing pose, appearance, shape, scale, and illumination changes as well as large motion and occlusions. The dashed (red) box shows the object bounding box and the solid (blue) small boxes are the object subregions.

Experiment 2: In this experiment the target object is a cube which is moved by a person's hands. Shown in Fig. 3.2, the second video is composed of 300 gray-scale images which are recorded at 7Hz with a size of 300×400. This video is challenging due to the similar gray-scale values of the cube and the person's hands, proportionally untextured object surface, and unpredicted object motion. Although at some frames the cube is passing from the right hand to the left hand and vice versa (e.g., 80, 169), the proposed method can track the object in the low-contrast scene where the object scale is also changing over time (e.g., 48, 98, 237, 284).

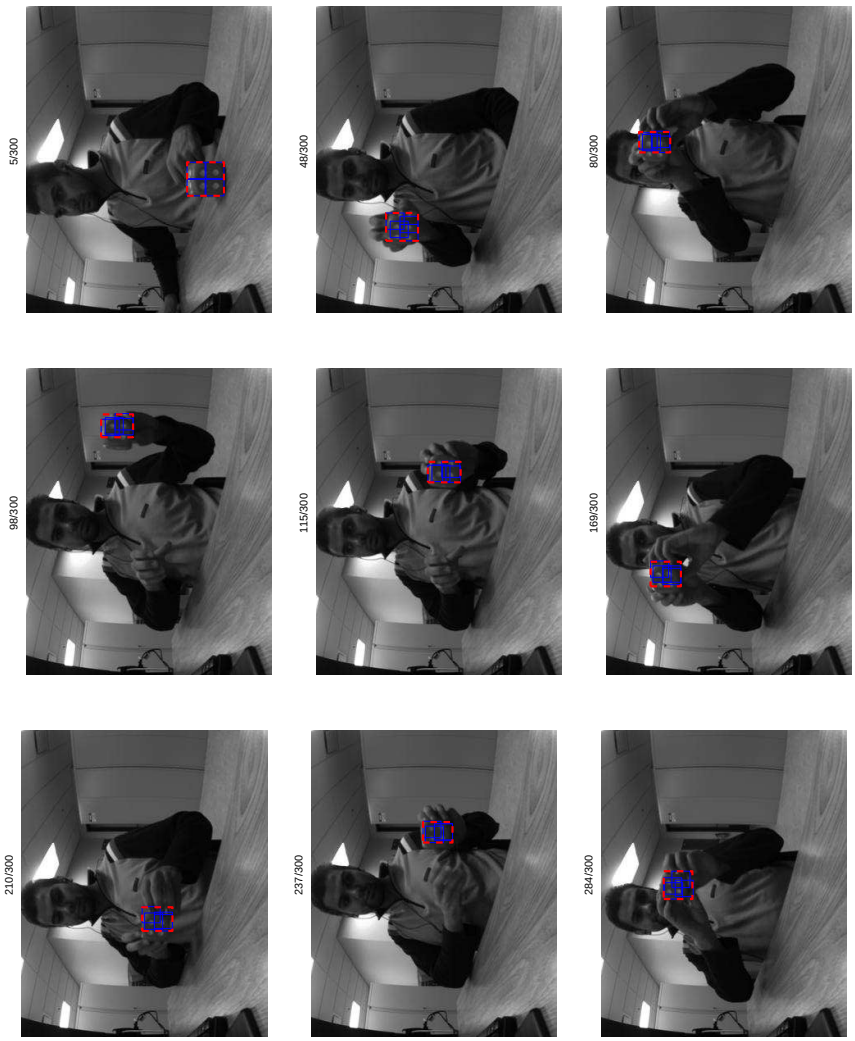


Figure 3.2: A cube moving by a person's hands, dashed (red) box shows the object bounding box and solid (blue) small boxes are the subregions

Experiment 3: The third video, shown in Fig. 3.3, consists of 1270 gray-scale images recorded at 30Hz with the size of 240×320. This video shows a moving dog doll under different situations such as pose, scale, and lighting changes (e.g., 55, 101, 171, 277, 612, 795, 934, 1197).

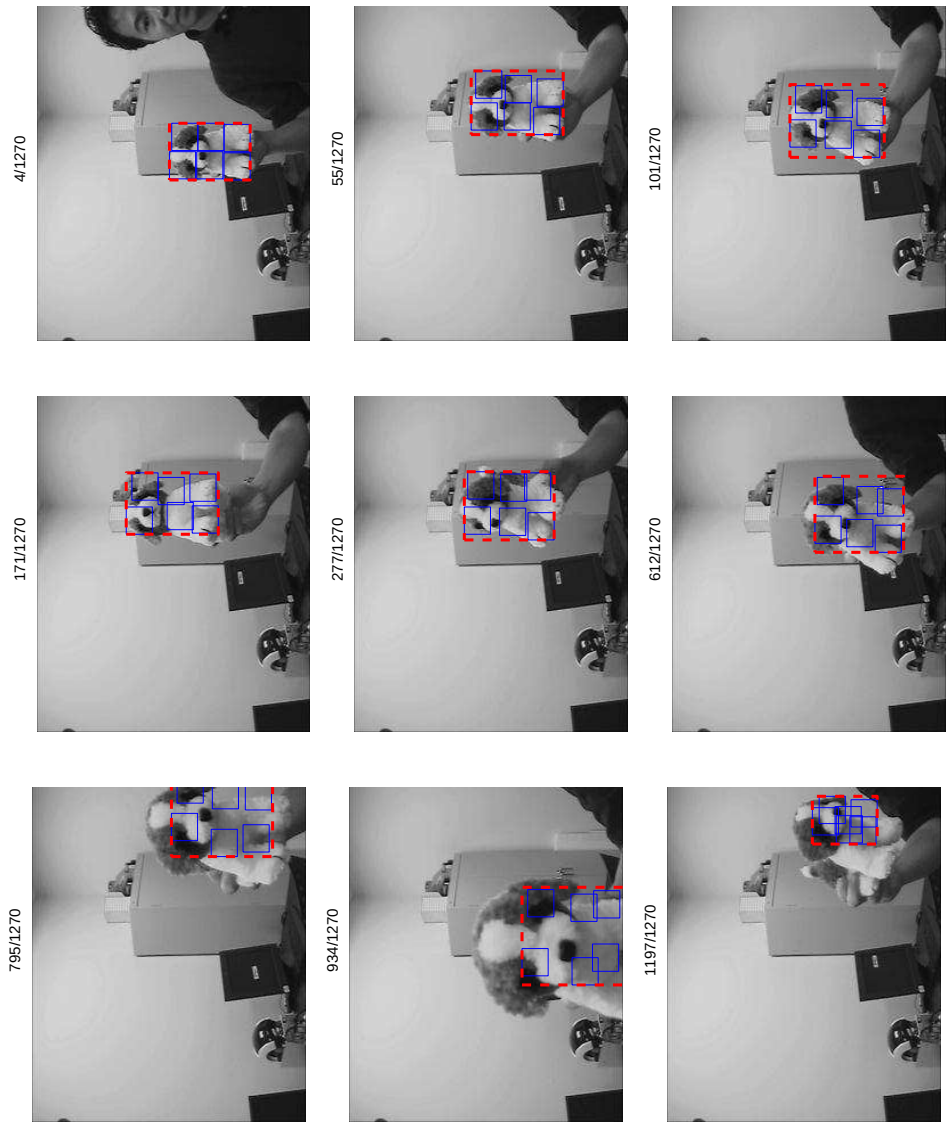


Figure 3.3: A dog doll moving in different pose and scale, dashed (red) box shows the object bounding box and solid (blue) small boxes are the subregions

Experiment 4: The forth experiment is a 360 × 240 grayscale movie recorded by a mobile camera. It contains several moving cars in a cluttered road. Shown in Fig. 3.4, the target car scale and size are changed over time (e.g., 37, 54, 110); however, it is accurately tracked by the proposed

method.

Figure 3.4: A car moving in a cluttered road, dashed (red) box shows the object bounding box and solid (blue) small boxes are the subregions

3.4.1 Qualitative Comparison

The accuracy and robustness of the proposed tracking method has been verified by comparing its performance with several state-of-the-art methods. Also the ground truth data is used to validate the comparison. In Fig. 3.5, the object bounding box obtained by the proposed method (bold dashed red box), the ground truth data (bold dotted yellow box), the Mean-shift [21] (dash-dot cyan box), the Fragment-based Tracker [1] (solid magenta box), the Color-Texture based Mean-shift [67] (dashed green box), and the Scale Adaptive Mean-shift [68] (blue ellipse) for 5 sample frames of different experiment videos are illustrated. Shown in this figure, the target object has been tracked accurately and robustly by the proposed method in different image sequences whereas other methods occasionally failed to locate the target at several frames. For instance, the Mean-shift method failed to track the target in the first experiment at frames 574 and 1120, the Fragment-based tracker failed to track the target in the second experiment at frames 78 and 178, the Color-texture based Mean-shift could not locate the target in the second experiment at frames 78 and 98, the Scale Adaptive based Mean-shift tracker did not track the target in the first and second experiments at frames 1120, 178, and 241 respectively. Based on this comparison, the

proposed method outperformed others in most frames. The Fragment based tracker was the second best tracker; however this method had significant drift at several frames (e.g., frame 1120 in the first movie, frames 78 and 98 in the second movie, frame 949 in the third movie), due to the large object motion, change in the object appearance or lighting. The Mean-Shift tracker and its extensions (i.e., the Color-Texture based Mean-Shift and the Scale Adaptive Mean-Shift) generally performed poorly especially in cluttered scenes where the object and background pixel values are mixed. The videos corresponding to the experimental results can all be found at `http://acis.ok.ubc.ca/~hfirouzi`.

Figure 3.5: A comparison of the proposed tracker (bold dashed red box) with the ground truth (bold dotted yellow box), the Mean-shift (dash-dot cyan box), the Fragment-based Tracker (solid magenta box), the Color-Texture based Mean-shift (dashed green box), and the Scale Adaptive Mean-shift (blue ellipse)

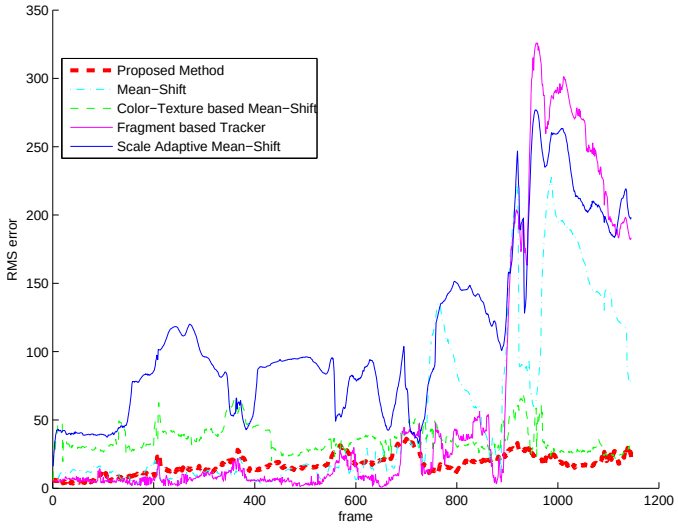


Figure 3.6: The RMS errors of the object bounding box obtained by each tracking method and the ground truth data for the first experiment

3.4.2 Quantitative Analysis

In addition to the qualitative comparison in Section 3.4.1, the manually-labeled "ground truth" object bounding box has been used to evaluate the precision of the proposed tracking method in comparison with the other methods. Shown in Fig. 3.6, 3.7, 3.8, 3.9, the root mean squared (RMS) error between the ground truth object bounding box and the estimated bounding box obtained by the proposed method is less than the others' RMS errors in most image frames. Although the error of the proposed tracker is not the least at all frames (e.g., frames between 400 and 700 in the first experiment or frames between 1190 and 1270 in the third experiment), it is still comparable with the best result obtained by the Fragment based Tracker. Although in the third experiment at frames between 900 and 1200, the proposed method as well as other trackers could not locate the target object precisely due to the significant changes in the object appearance and scale, the proposed tracking method managed to resolve this problem after frame 1200 whereas other methods such as Mean-shift failed to track the object. In Fig. 3.10, the accumulated RMS error between the ground truth data and the trackers' results are illustrated. Based on this figure, the average tracking error obtained by the proposed tracking method is the least in comparison with the other methods.

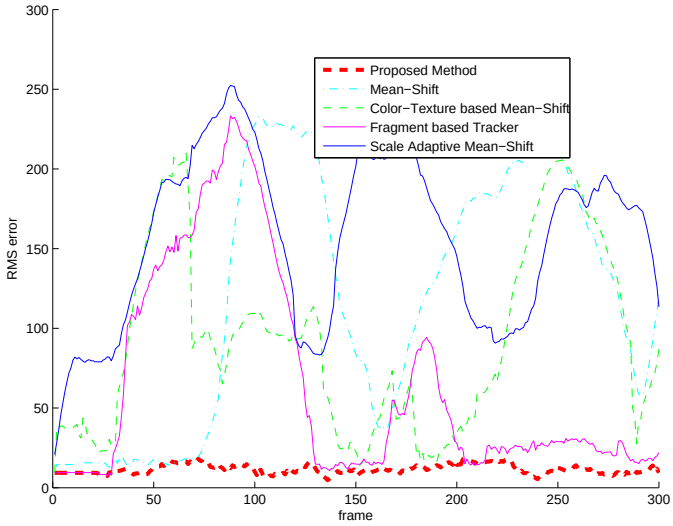


Figure 3.7: The RMS errors of the object bounding box obtained by each tracking method and the ground truth data for the second experiment

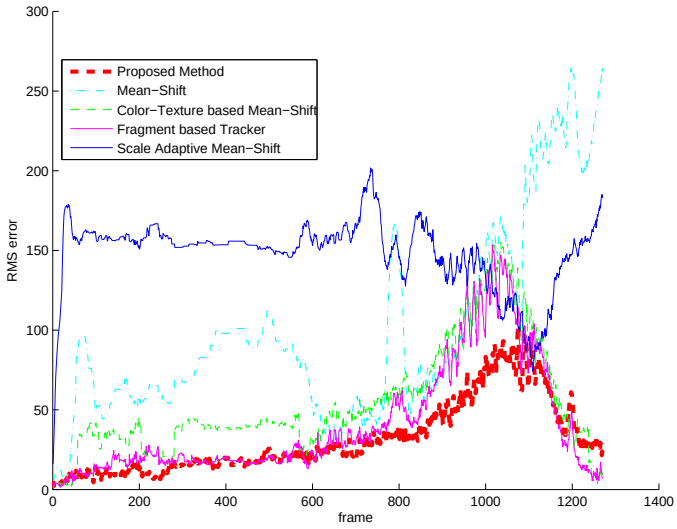


Figure 3.8: The RMS errors of the object bounding box obtained by each tracking method and the ground truth data for the third experiment

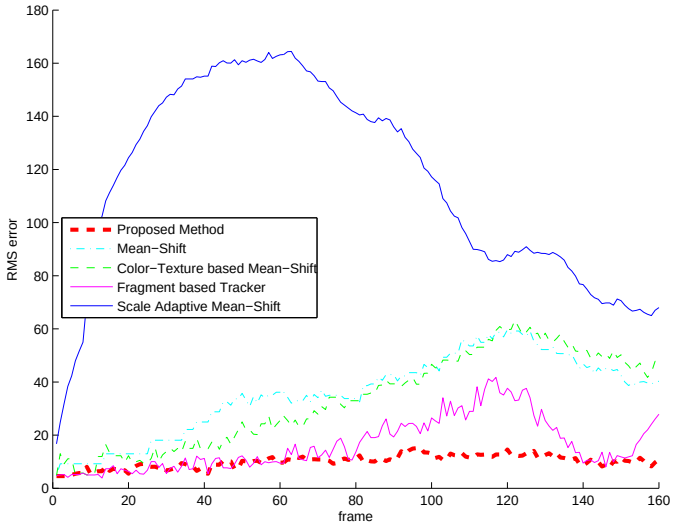
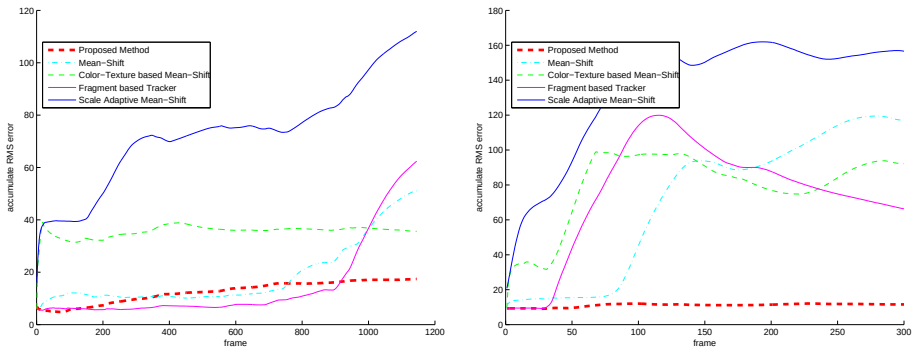
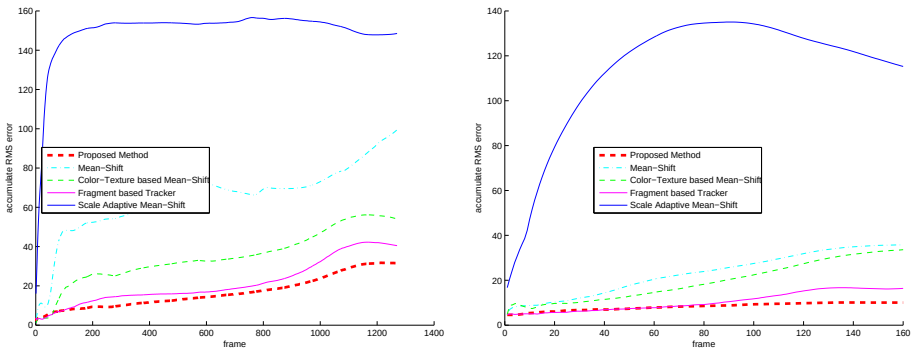


Figure 3.9: The RMS errors of the object bounding box obtained by each tracking method and the ground truth data for the fourth experiment



(a) Experiment 1

(b) Experiment 2

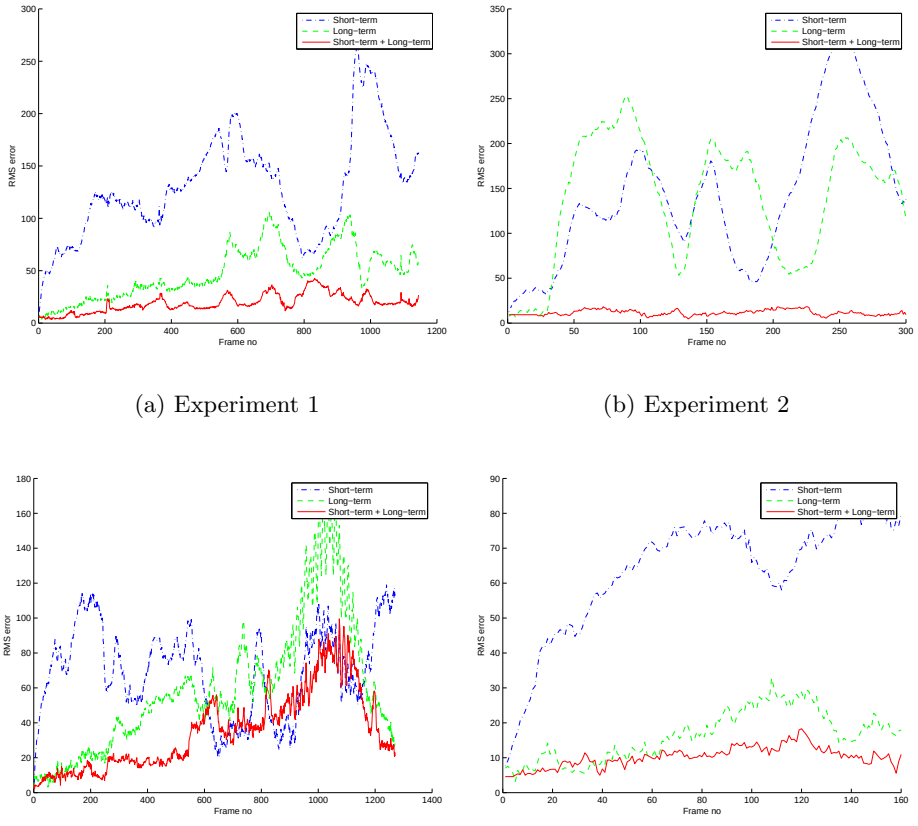


(c) Experiment 3

(d) Experiment 4

Figure 3.10: The accumulated RMS errors of the object bounding box obtained by each tracking method and the ground truth data for each frame in all experiments

In order to closely investigate the contribution of each of the adaptive templates (i.e., short-term and long-term templates) to the overall performance of the proposed method, all experiments have been repeated using only one template at a time. Fig. 3.11 compares the tracking RMS error obtained by the short-term, the long-term, and a combination of short-term and long-term templates in each experiment. This figure shows that the RMS error of the proposed method using both templates is consistently less than that of each template individually. Therefore, combining both short-term and long-term templates, we are able to significantly improve the accuracy of the proposed tracking method.



(a) Experiment 1

(b) Experiment 2

(c) Experiment 3

(d) Experiment 4

Figure 3.11: The RMS error of the proposed method using short-term, long-term, and both short-term and long-term templates in all experiments

3.4.3 Implementation

In regards to the computational cost, it is verified from my experiments that the processing time of the proposed method implemented in Matlab(R) takes 0.12 seconds in average for one image frame on a PC with a 2.93 Ghz Intel(R) Core(TM)2 Due CPU. It is highly expected that the computation cost of the proposed tracker will be significantly improved using C++ implementation and parallel programming. In fact, the proposed object localization algorithm consists of multiple simultaneous optimization steps which can be run in parallel on a multi-thread processor.

More precisely, the computational cost of the proposed method may change based on several parameters including object size, number of subregions, and maximum number of optimization iterations (i_{max}). For instant in the first and the forth experiments, although the image size is the same (i.e., 360×240), the required time to process one image frame is 0.2 and 0.09 seconds respectively. The difference in the processing time is because of having different number of subregions and object size. In the first experiment, the target object with size of 61×61 is partitioned into six subregions, whereas in the forth experiment there are four subregions and the object size is 54×40.

Moreover, by changing the maximum number of optimization iterations, we can improve the computational cost. However, it is observed that the proposed method may occasionally fail to track objects when this parameter is set to a small value. Fig. 3.12 illustrates the required time to process one image frame versus the tracking RMS error. Based on this figure, when the maximum number of optimization iterations is decreased, the processing time is reduced but the tracking RMS error is increased. In this work, the parameter i_{max} is empirically set to 30 so that both the computational cost and the tracking accuracy are proportionally adequate. Likewise, number of objects is also another parameter which can increase the overall computational cost. In the case of multiple targets, the proposed method can be used to track each object independently and the total processing time is the sum of the required processing time by each tracker.

In general, partitioning the target object into several subregions can improve the tracking robustness against shape deformation, non-rigidity, and scale variations. However, it is observed from the experiments that the accuracy of tracking decreases when subregions are either very small or too large with respect to the object size. Fig.3.13 shows the average RMS tracking error and the average processing time of one image frame using the proposed method for different subregion sizes including (5×5), (10×10),

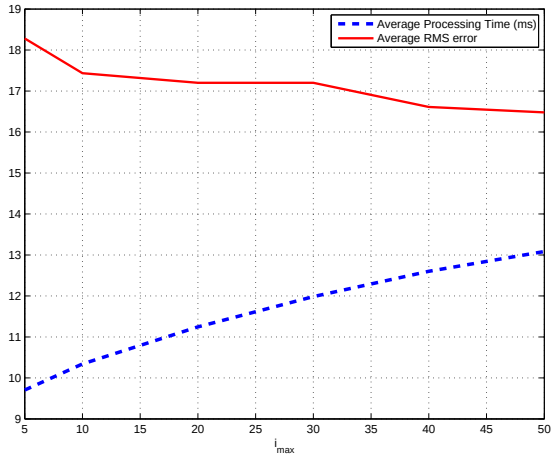


Figure 3.12: Average processing time and average RMS error for different values of the maximum optimization iterations parameter

(20×20), (22×22), (25×25), (30×30), and (50×50). Based on this figure, the RMS error of the proposed method is almost the same for subregion size between (20×20) and (30×30). However, the computational cost of the method is increased when the subregion size is very small and too big. In general, the object region should be partitioned based on the type and size of the target object. In this work, according to the average size of objects (i.e., 51.75×55.5), the subregion with the size of (22×22) has been experimentally used for partitioning.

3.5 Discussions

This chapter presents a component-based tracker which models each component named subregion by two heterogeneous adaptive templates. The new location of each subregion is independently estimated using an EM-like optimization method with multiple initializing points. The object location is obtained by the robust fusion of the subregions locations. Also, the locations of subregions are then corrected based on the final object location. Based on the experimental results shown in Section 4.4, the proposed tracking method is able to track a target object whose pose, appearance, shape, and scale may change over time. In addition, the proposed tracker is robust to temporary occlusion, large and unpredicted motion, and noise. In fact the robustness and accuracy of this method can be attributed to several factors.

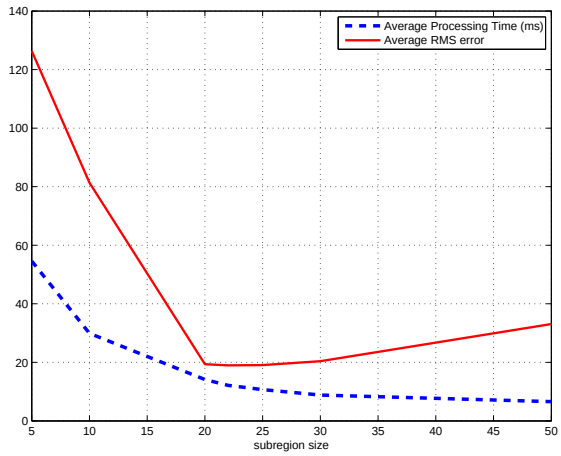


Figure 3.13: Average processing time and average RMS error for different subregion sizes

First, partitioning the object bounding box into several subregions and fusing the subregion locations improve the tracking robustness and performance by managing the shape variations and scale changes in a distributed framework. Also partial occlusion and tracking in a cluttered scene can be handled at the fusion step by rejecting the outliers and invalid subregions. Besides the component-based tracking framework, the proposed EM-like algorithm can efficiently localize the subregion next location by initializing the optimization process with several possible solutions simultaneously. Multiple start points increase the localization robustness especially where the target object motion is large and complex. Finally, the proposed template-based tracking method is robust to the "drift" problem and significant appearance changes because of modeling the object subregions by two different adaptive templates named immediate and delayed templates. The combination of these templates allows the algorithm to handle both short-term and long-term appearance variations and alleviate the drift problem.

Shown in Figures 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10, the proposed tracker performed well in many cases and can fairly handle unpredicted object motion because of using multiple initialization points for the localization method, however it cannot locate the target object at some stages especially where the changes in object deformation and scale are significant. It is also observed from the experiments that the proposed tracker is not always robust against rapid motion. This problem can be solved by increasing the maximum optimization iterations (i_{max}) and the number of initialization

points (δX_l). In fact, more initialization points around the previous object location can improve the tracking accuracy where the object motion is proportionally large. Moreover the inclusion of several overlapping subregions as well as non-overlapping ones can improve the tracking robustness against significant appearance, shape, and scale changes.

As an extension to this work, a more robust fusion method can be used. Also a more general motion model may improve the tracking performance where the object motion is more complex and unpredicted.

Chapter 4

Efficient and Robust Multi-Template Tracking Using Multi-start Interactive Gaussian-based Optimization

Building on the ideas presented in Chapter 3, this Chapter describes an overhaul of the approach in Chapter 3 to improve its performance and robustness for real-time applications. The contributions and differences of the work presented in this Chapter (my second method) in comparison with the work proposed in previous Chapter (my first method) include:

- Target representation model: In the method presented in Chapter 3, the image region of the target object is initially partitioned into several sub-regions, and subsequently each sub-region is represented by two Gaussian-based templates namely immediate and delayed templates. Similarly, in this Chapter the proposed representation model consists of two time-varying templates which can model both short-term and long-term changes in the target appearance. However, other appearance models such as Local Binary Pattern (i.e., a texture descriptor insensible to illumination changes) can be efficiently integrated into the proposed multi-model target representation which is not possible in my first method. Considering the structural differences, the current target model can be customized suitably to obtain a more satisfactory result in comparison with the model presented in previous Chapter.

- Representation model learning: The mean and variance of the templates used in my first method are updated separately based on an updating ratio and the tracking time step, respectively. On the other hand, both the mean and variance of the templates in the second work are adaptively updated based on a forgetting factor, uncertainty margin, and the tracking time step. Therefore, in comparison with the

method presented in Chapter 3 the current proposed template update strategy is not only more adaptive to new appearance changes because of the use of a forgetting factor and the tracking time step, but also more robust against noise and occlusion due to the uncertainty margin used in the learning algorithm.

– Target localization algorithm: The first method uses a predefined multi-start Gradient-based search to estimate the preliminary location of the target sub-regions based on a translational transformation and immediate template. Consequently, the delayed template is employed to correct the preliminary estimation. At the end of this two-step optimization, the target is tracked by fusion of the new sub-region locations. In contrast, the target localization in the current work features an interactive multi-start hybrid search that takes into account generic transformations using a combination of sampling-based and gradient-based algorithms in a unified probabilistic framework. Unlike the two-step optimization used in my first method, in the current method all appearance models (i.e., the short- and long-term templates) are used to find the best location of the target, simultaneously. This approach further increased both the efficiency and accuracy of the proposed tracker.

The rest of this chapter is organized as follows. Related work is reviewed in Section 4.1. In Section 4.2, the proposed appearance model is defined. The formulation and algorithm of the proposed multi-start Gaussian-based template tracking method are explained in details in Section 4.3. In Section 4.4, the proposed tracker is applied on five challenging image sequences and subsequently the results are compared with four state-of-the-art methods as well as the ground truth data. Concluding discussions and potential extensions for future work will be provided in Section 4.5.

4.1 Related Work

Since early template-based tracking methods [59], different algorithms have been proposed to improve the accuracy and efficiency of the tracking; Bergen et al. [11] used a more general motion model e.g., affine transformation, Black and Jepson [12] improved the robustness of the template matching against appearance changes by employing a linear appearance variation, Hager and Belhumeur [38] increased the tracking efficiency by a real-time implementation, and Cootes et al. [22] modeled the object appearance by

Active Appearance Models (AAMs) to handle non-rigid objects. Matthews et al. [62] proposed a method based on an adaptive appearance template which does not suffer from the "drift" problem. Instead of using previous update strategies which involve either no update ($T_{n+1} = T_1$ for all $n \geq 1$) or a naive template update ($T_{n+1} = I_n(W(X; \Theta_n))$ for all $n \geq 1$), they first estimate new transformation parameters Θ_{n+1} based on the naive template update, and then the estimated parameters are used as a starting point to align template T_{n+1} with T_1. This method is relatively stable to the local minimum by reinitializing the gradient-based search. However, the method proposed in [62] cannot handle the occlusion and outliers, and it also fails when it tracks non-rigid objects, especially when the object shape is changing over time. Schreiber [84] presented a robust template matching algorithm to handle partial occlusion and outliers. Unlike other robust template trackers such as [8, 38], in this method the robust weights are adaptively updated only after finding the transformation parameters for a new image to improve the computational efficiency. However, this method is not robust to track non-rigid objects in different lighting conditions. Silveira and Malis [87] used several image transformation models to improve the template-based tracking performance against illumination changes. They proposed a new illumination model which can be used to track a deformable target with illumination change. In this method, a general image formation model which covers both geometric and photometric deformations is defined to track a rigid or deformable object. Although the proposed method is robust to illumination change and general object deformation, it cannot handle large and unpredicted pose and appearance changes due to the gradient-based optimization and a large number of parameters estimated. Also this method is not stable when the target image is not sufficiently textured and unable to track non-rigid objects with variant shape and structure. In the previous chapter, I proposed a component-based template tracking method using two heterogeneous adaptive templates namely short-term and long-term templates. This approach relies on a multi-start EM-like optimization algorithm to estimate the new object transformation parameters. Building on the ideas of the multi-start EM-like localization method, this work describes an overhaul of the approach proposed in Chapter 3 to improve its performance and robustness for real-time applications.

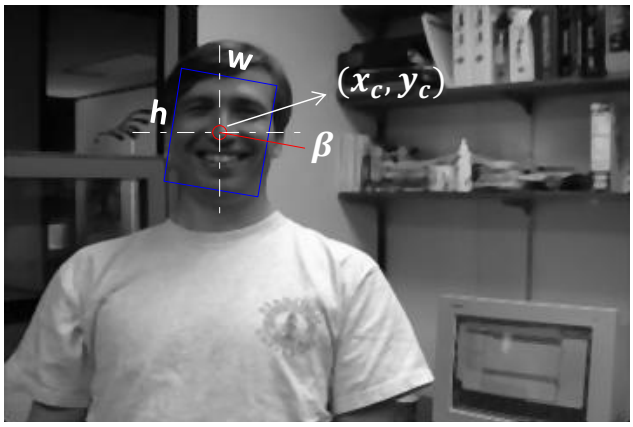


Figure 4.1: Object region parameters [x_c: center x, y_c: center y, w: width, h: height, and β: rotation]

4.2 Adaptive Gaussian-based Appearance Model

In the proposed method, the object appearance is represented by two adaptive templates called *short-term* (T_S) and *long-term* (T_L) templates. Every point $X = \{x, y\}$ inside each template is modeled by a Gaussian function $G_{k(X)} \sim N(\mu_{k(X)}, \sigma_{k(X)})$ where $\mu_{k(X)}$ and $\sigma_{k(X)}$ are the mean and variance of the point X at time step k respectively. Also a 5-DOF[10] transformation function is used to map the point X in the coordinate frame of the template to the point located at $Y = W(X; \Theta)$ in the coordinate frame of the image. The transformation function is composed of translation (d_x and d_y are the object translation along the x and y axes), rotation (β is the object rotation in the image plane), and scale transformations (s_x and s_y are the scale factors along the x and y axes).

$$W(X;\Theta) = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & d_x \\ \sin(\beta) & \cos(\beta) & d_y \end{bmatrix} \begin{bmatrix} s_x \times x \\ s_y \times y \\ 1 \end{bmatrix} \tag{4.1}$$

Figure 4.1 shows the object region, which is specified by a deformable bounding box. This box is defined by five parameters $R = \{x_c, y_c, w, h, \beta\}$ where x_c and y_c are the center pixel coordinates of the object, w and h are the width and height of the region, and β is the object rotation.

The parameters of the Gaussian functions are initialized using the first object image as follows:

[10]Degree of freedom

$$\mu_{0(X)} = I_0(Y_0) \; ; \; \sigma^2_{0(X)} = 1 \tag{4.2}$$

where $Y_0 = W(X, \Theta_0)$ and Θ_0 is the transformation parameters given at the first image frame.

At time step k, the mean and variance of the Gaussian functions are updated every m image frames using a forgetting factor γ and an uncertainty margin σ_0^2.

$$\begin{aligned} \mu_{k(X)} &= \frac{\gamma \times n \times \mu_{k-1(X)} + m \times \widehat{\mu}_{m(X)}}{\gamma \times n + m} \\ \sigma^2_{k(X)} &= \frac{\gamma \times n \times \sigma^2_{k-1(X)} + m \times \widehat{\sigma}^2_{m(X)}}{\gamma \times n + m} \end{aligned} \tag{4.3}$$

where $\widehat{\mu}_{m(X)}$ and $\widehat{\mu}_{m(X)}$ are the approximate mean and variance of the m previous data which are calculated as follows:

$$\begin{aligned} \widehat{\mu}_{m(X)} &= \frac{1}{m} \sum_{i=k-m+1}^{k} I_{i(Y_i)} \\ \widehat{\sigma}^2_{m(X)} &= \frac{1}{m} \left[\sum_{i=k-m+1}^{k} \left(I_{i(Y_i)} - \mu_{k(X)} \right)^2 \right] + \sigma_0^2 \end{aligned} \tag{4.4}$$

In this work, for the short-term and long-term template the following parameters have been used.

Table 4.1: Template updating parameters

	γ	σ_0^2	m (batch size)
short-term	0.85	1	2
long-term	0.97	3	5

Shown in Table 4.1, the forgetting factor parameter is smaller for the short-term template to quickly adapt to the target appearance changes whereas the long-term template is designed to be robust against the outliers, sudden appearance changes, and occlusion. In addition, the short-term template uses a smaller uncertainty margin (σ_0^2) and batch size (m) to allow a quicker update in comparison with that of the long-term template. In the proposed template update strategy, the variance of each Gaussian $\sigma^2_{k(X)}$ is empirically estimated from the error between the template and the

new images. Therefore, those points with comparably higher variance are more likely labeled as outliers in the localization algorithm which will be elaborated in Section 4.3.1.

4.3 Multi-start Interactive Object Tracking

In general, visual tracking consists of two different processes which are (1) Target Representation and Localization (TRL), and (2) Filtering and Data Association (FDA). The latter refers to tracking an object through the estimation of its motion dynamics whereas the former refers to locating the object based on the object appearance and shape. These two processes may be combined with different importance factors depending on the application. For instance in the case of face tracking [14] in a crowded environment, the tracking method is mostly based on TRL rather than FDA because modeling the target appearance is more reliable than predicting the target dynamics. On the other hand, for the applications such as aerial video surveillance where the target motion can be accurately estimated [53], the FDA process is often preferable for target tracking.

In the proposed method, at each localization step, first a certain number of guess points is randomly chosen according to the object motion history. These guess points then initialize several gradient-based optimization processes which interactively find the new transformation parameter to minimize the sum of Gaussian errors between the template and the candidate sub-image. In the following subsections, the proposed localization method is explained in detail. Also for the sake of clarity, the time instant is omitted in the equations.

4.3.1 Object Localization

The localization problem can be viewed as an optimization task. Since the proposed object template is composed of Gaussian functions, at every time step k I optimize the Sum of Gaussian Errors (SGE) between the target representation model and the received image I_k to estimate the changes in the target transformation parameters $\Delta\Theta_k$.

$$\Delta\Theta_k = \underset{\Delta\widehat{\Theta}_k}{\arg\max} \left[\sum_{X \in R} \exp\left(\frac{(I_{k(\widehat{Y}_k)} - \hat{\mu}_{k(X)})^2}{-2\hat{\sigma}^2_{k(X)}} \right) \right] \tag{4.5}$$

where $\widehat{Y}_k = W(X; \Theta_{k-1} + \Delta\widehat{\Theta}_k)$, also, $\hat{\mu}_k$ and $\hat{\sigma}^2_k$ are the prediction of the mean and variance of the Gaussians at time step k. In this work, $\hat{\mu}_k = \mu_{k-1}$

and $\hat{\sigma}_k^2 = \sigma_{k-1}^2$.

Using the Taylor series, we can expand the transformed image as:

$$I_{W(X;\Theta_{k-1}+\Delta\widehat{\Theta}_k)} = I_{W(X;\Theta_{k-1})} + \nabla I_{W(X;\Theta_{k-1})} \frac{\partial W_{(X;\Theta_{k-1})}}{\partial \Theta} \Delta\widehat{\Theta}_k \tag{4.6}$$

where $\nabla I_{W(X;\Theta_{k-1})}$ is the image gradient and taking the derivative of the transformation function W with respect to its parameters Θ, we obtain:

$$\frac{\partial W_{(X;\Theta_{k-1})}}{\partial \Theta} = \frac{\partial Y_{k-1}}{\partial \Theta} = \begin{bmatrix} -xs_xS_\beta - ys_yC_\beta & xC_\beta & -yS_\beta & 1 & 0 \\ xs_xC_\beta - ys_yS_\beta & xS_\beta & yC_\beta & 0 & 1 \end{bmatrix} \tag{4.7}$$

where $Y_{k-1} = W(X;\Theta_{k-1})$, S_β and C_β denote $\sin(\beta)$ and $\cos(\beta)$, respectively.

As a result, by substituting Eq. 4.6 into Eq. 4.5, we obtain:

$$\Delta\Theta_k = \arg\max_{\Delta\widehat{\Theta}_k} \left[\sum_{X\in R} \exp\left(\frac{(I_{k(Y_{k-1})} + \nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \Theta} \Delta\widehat{\Theta}_k - \hat{\mu}_{k(X)})^2}{-2\hat{\sigma}^2_{k(X)}} \right) \right] \tag{4.8}$$

One solution for estimating $\Delta\Theta_k$ is to take the derivative of Eq. 4.8 with respect to $\Delta\Theta$ and set the equation to zero.

$$\Delta\Theta_k = H^{-1} \sum_{X\in R} \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \Theta} \right]^T \left(\hat{\mu}_{k(X)} - I_{k(Y_{k-1})} \right) \tag{4.9}$$

where the Hessian matrix is:

$$H = \sum_X \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \Theta} \right]^T \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \Theta} \right] \hat{\sigma}^2_{k(X)} \tag{4.10}$$

According to Eq. 4.10, the Hessian matrix needs to be calculated in every image frame, which is not possible in many real-time applications. Nevertheless, under the independency assumption, we can calculate the transformation parameters based on Eq. 4.11.

$$\Delta\theta_k^{(i)} = \frac{\sum_X \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \theta^{(i)}} \right]^T \left(\hat{\mu}_{k(X)} - I_{k(Y_{k-1})} \right)}{\sum_X \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \theta^{(i)}} \right]^T \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \theta^{(i)}} \right] \hat{\sigma}^2_{k(X)}} \tag{4.11}$$

where $\Delta\Theta_k = \{\Delta\theta_k^{(i)}\}\ ;\ i \in [1, N_\Theta]$.

Note that in Eq. 4.11, $\frac{\partial Y_{k-1}}{\partial \theta_i}$ and $\nabla I_{k(Y_{k-1})}$ are $[2 \times 1]$ and $[1 \times 2]$ vectors; therefore, no matrix inversion is required, i. e., a significant improvement in the efficiency of the localization task.

In the following subsection, an interactive multi-start EM-like optimization is proposed to estimate the new transformation parameters. In addition to the fact that EM is a powerful optimization algorithm which can manage incomplete data in a Bayesian framework, different starting points increases the robustness of the optimization algorithm against local minima and outliers. In the following subsections, the proposed optimization algorithm is explained in detail.

4.3.2 Interactive Multi-start Optimization

Given L starting points at every tracking step as $\Delta\widehat{\Theta}_{k,l}$; $l \in [1, L]$, we can compute the conditional error density function of the object location using the starting point $\Delta\widehat{\Theta}_{k,l}$ as:

$$P_l(X|\widehat{\Theta}_{k,l}) = \exp\left(\frac{\left(I_{k(Y_{k-1})} + \nabla I_{k(Y_{k-1})}\frac{\partial Y_{k-1}}{\partial \Theta}\Delta\widehat{\Theta}_{k,l} - \hat{\mu}_{k(X)}\right)^2}{-2\hat{\sigma}^2_{k(X)}}\right) \quad (4.12)$$

Accordingly, the conditional error density function of the object location can be defined as a mixture model.

$$P(X|\Phi) = \sum_{l=1}^{L} \alpha_l P_l(X|\widehat{\Theta}_{k,l}) \ , \ \sum_{l=1}^{L} \alpha_l = 1 \quad (4.13)$$

where α_l ; $l \in [1, L]$ are the contributing weights and $\Phi = \{\widehat{\Theta}_{k,l}, \alpha_l\}$ is the parameter set of the mixture model.

Also based on the Bayes's rule we obtain:

$$P(l|X, \Phi) = \frac{\alpha_l P_l(X|\widehat{\Theta}_{k,l})}{\sum_{j=1}^{L} \alpha_j P_k(X|\widehat{\Theta}_{k,j})} \quad (4.14)$$

Similar to the EM algorithm used in [77] for estimating a mixture densities, the multi-start localization problem can be viewed as maximizing the following function.

$$Q(\Phi, \Phi^-) = \sum_{l=1}^{L} \sum_{X \in R} \left[\log(\alpha_l) + \log\left(P_l(X|\widehat{\Theta}_{k,l})\right)\right] P(l|X, \Phi^-) \quad (4.15)$$

where Φ^- is the parameter set obtained from the previous optimization iteration.

Taking the derivative of Eq. 4.15 with respect to α_l and considering the constraint $\sum_{l=1}^{L} \alpha_l = 1$, we obtain:

$$\frac{\partial Q(\Phi, \Phi^-)}{\partial \alpha_l} = 0 \rightarrow \alpha_l = \frac{1}{N_R} \sum_{X \in R} P(l|X, \Phi^-) \tag{4.16}$$

where N_R is the area of the object region R.

Using Eq 4.12 and taking the derivative of Eq. 4.15 with respect to $\Delta\Theta_l$, we obtain:

$$\frac{\partial}{\partial \Delta\hat{\Theta}_{k,l}} \left\{ \sum_{X \in R} \log \Big(P_l(X|\widehat{\Theta}_{k,l}) \Big) P(l|X, \Phi^-) \right\} = 0 \tag{4.17}$$

Similar to Eq 4.11, the changes in transformation parameters $\Delta\widehat{\Theta}_{k,l} = \{\Delta\hat{\theta}_{k,l}^{(i)}\}$; $i \in [1, N_\Theta]$ are computed as the following equation:

$$\Delta\hat{\theta}_{k,l}^{(i)} = \frac{\sum_X \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \theta^{(i)}} \right]^T \big(\hat{\mu}_{k(X)} - I_{k(Y_{k-1})} \big) P(l|X, \Phi^-)}{\sum_X \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \theta^{(i)}} \right]^T \left[\nabla I_{k(Y_{k-1})} \frac{\partial Y_{k-1}}{\partial \theta^{(i)}} \right] \hat{\sigma}^2_{k(X)} P\left(l|X, \Phi^-\right)} \tag{4.18}$$

As a result the transformation parameters are obtained:

$$\widehat{\Theta}_{k,l} \leftarrow \widehat{\Theta}_{k,l} + \Delta\widehat{\Theta}_{k,l} \tag{4.19}$$

4.3.3 Tracking Algorithm

According to the formulation provided in the previous sections, at every tracking step, first a set of L different transformation vectors $\widehat{\Theta}_{k,l}$ is randomly generated based on a Gaussian distribution around the previous values with a diagonal variance matrix $\sigma_\Theta = diag(\sigma_{\theta_1}, .., \sigma_{\theta_K})$. Then the optimization parameters $\Phi = \{\widehat{\Theta}_{k,l}, \alpha_l\}_{l \in [1,L]}$ are iteratively estimated based on equations 4.16 and 4.18 to find the best transformation vector.

In the proposed tracking algorithm, both short-term and long-term templates are optimized simultaneously to find the best transformation parameters. In fact the proposed multi-start search method is capable of efficiently optimizing any arbitrary number of templates at the same time. In this work, the short-term and long-term templates are considered as a $N \times 1$ vector where N is the number of points inside the template and the mean

μ_k and variance σ_k are $(2N) \times 1$ vectors which are the concatenate of the short-term and long-term vectors.

The optimization process is terminated if either it has been performed for a certain number of iterations (it_{max}) or the following inequality is satisfied.

$$\max \left[\alpha_l \times \sum_{i=1}^{N_\Theta} |\Delta \hat{\theta}_{k,l}^{(i)}| \right] < opt_{max\_err} \tag{4.20}$$

where opt_{max_err} is the maximum acceptable error which can be obtained by the proposed optimization method.

Algorithm 3: Multi-start Adaptive Multi-Template Tracking

Require: object region at the first frame ($R_0 = \{x_{c0}, y_{c0}, w_0, h_0, \beta_0\}$)

$\Theta_0 \leftarrow \{\beta_0, s_{x0} = 1, s_{y0} = 1, d_{x0} = x_{c0} - w_0/2, d_{y0} = y_{c0} - h_0/2\}$
$\mu_X \leftarrow I^1(W(X; \Theta^1))$, $\sigma_X \leftarrow \{1\}$
for $k = 1$: end of image sequence **do**
  for all $l \in [1, L]$ **do**
    $\alpha_l \leftarrow \frac{1}{L}$
    $\widehat{\Theta}_{k,l} \leftarrow$ a random sample from distribution $N(\Theta_{k-1}, \sigma_\Theta)$
  end for
  for $it_{opt} = 1 : it_{max}$ **do**
    for all $l \in [1, L]$ **do**
      estimate $P_l(X|\widehat{\Theta}_{k,l})$, $P(l|X, \Phi)$, $\Delta\widehat{\Theta}_{k,l}$, and $\widehat{\Theta}_{k,l}$ using equations 4.12, 4.14, 4.18, and 4.19 respectively
      estimate α_l using Eq. 4.16
    end for
    break if inequality 4.20 is satisfied
  end for
  $\Theta_k \leftarrow \widehat{\Theta}_{k,l^*}$ where $l^* = \arg\max_{l \in [1,L]} [\alpha_l]$
  for each template, update $\mu_{k(X)}$ and $\sigma_{k(X)}$ using Eq. 4.3
end for

A schematic algorithm of the proposed tracking method is illustrated in Algorithm 3. Based on this algorithm, the proposed parallel optimization using multiple templates and starting points can outperform a sequentially execution of an optimization algorithm with different templates or initialization points. Indeed, parallel optimization can improve the accuracy and convergence speed of the overall optimization process by comparing the results at each iteration.

4.4 Experimental Results

In this section, the robustness and accuracy of the proposed tracking method have been experimentally evaluated using five publicly available gray-scale image sequences that contain different challenging situations including significant object pose, appearance, and scale variations, partial and full occlusion, illumination changes and noise, complex and unpredicted motion, and cluttered scene. Also the pixel values in my implementation are normalized to real values in the range [0 1]. The target object region (R) can be initially specified manually or by any object detector. In the following experiments, the maximum error of the localization optimization (opt_{max_err}), the maximum number of optimization iterations, and the number of starting points (L) are set to 0.1, 5, and 50 respectively. Also the object template size is the actual object regions size divided by two and the variance matrix σ_Θ is empirically set to $\{\sigma_{\beta=0.01}, \sigma_{s_x} = 0.01, \sigma_{s_y} = 0.01, \sigma_{d_x} = 5, \sigma_{d_y} = 5\}$. These parameters are fixed for all experiments. It will be shown in Section 4.4 that this selection of parameters provides satisfactory results in term of both accuracy and computational efficiency.

4.4.1 Comparison and Analysis

For each experiment, the result of the proposed tracking method called ERTM (bold dashed red box) has been compared with five of the state-of-the-art trackers including: 1)TLD tracker[47] (dashed green box), 2) Incremental Visual Tracking[82] (solid blue box), 3) Decentralized Template Tracking[31] (dashed black box), 4)Mean-shift[21] (dash-dot cyan box), and 5)Fragment-based Tracker[1] (solid magenta box), as well as the manually labeled ground truth data (bold dotted yellow box) to validate the comparison. Also to establish a quantified comparison, the mean value of the long-term and short-term templates are shown at the top-right of each figure respectively. Moreover, for each image sequence two separate chronological list of images are provided to show the change of the adaptive templates over time.

In addition to the qualitative comparison shown in Figures 4.2, 4.5, 4.8, 4.11, and 4.14, the "ground truth" data has been used to evaluate the precision of the proposed tracking method in comparison with the other methods. According to Figures 4.4, 4.7, 4.10, 4.13, and 4.16, the root mean squared (RMS) of the difference between the ground truth center point and the estimated location obtained from the proposed method is for the most part less than that of the other trackers. In these Figures, the average RMS error

indicates the average of the RMS error from the beginning up to each time step.

Sequence dudek The first image sequence[11], illustrated in Figure 4.2, shows a face subject to different challenging situations including appearance and pose variations, camera movement, partial and full occlusion, scale changes, rotation,and light changes.

Figure 4.2: sequence *dudek*: the proposed tracking result (bold dashed red box) in comparison with TLD (dashed green box), Mean-shift (dash-dot cyan box), Fragment-based tracker (solid magenta box), DRTT (dashed black boc), IVT (solid blue box) and ground truth data (bold dotted yellow box)

Shown in Figure 4.2, the proposed tracker is suitably robust to handle various challenging situations such as different object poses (e.g., frames #2, #189, #1134), scales (e.g., frames #2, #575, #1134), illumination changes (e.g., #836, #920), and temporary occlusion (e.g., frames #208, #211, #367) whereas MS and FT ultimately failed to track the target. Based on Figure 4.4, from frame around #700 to #900, Mean-shift,and Fragment-based trackers started to drift from the target location mostly due to the significant changes in the object appearance and unpredicted motion. Although TLD tracker managed to successfully track the target to the end, it is not robust to occlusion and sudden appearance changes (e.g., frames #226, #920).

[11] `http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html`

Figure 4.3: sequence *dudek*: changes in the mean of the long-term (top) and short-term (bottom) template over time

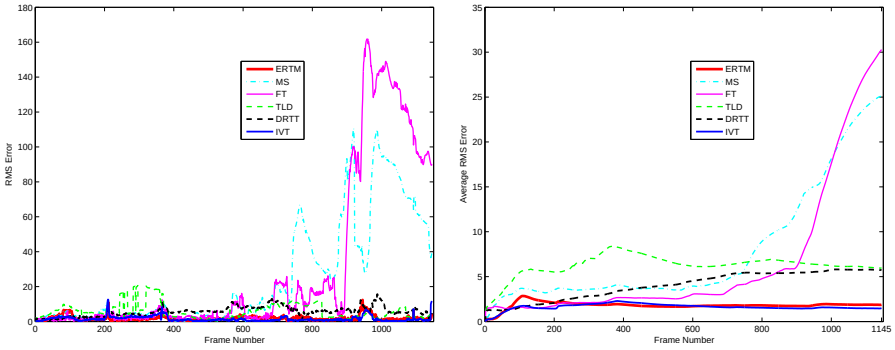


Figure 4.4: sequence *dudek*: the RMS error (left) and the average RMS error (right) between the ground truth data and the result of all trackers

Sequence david The second sequence[12] contains illumination, scale, pose, and appearance changes as well as partial occlusion and out of plane rotation (e.g., #137, #156, #376 #390).

Shown in Figure 4.5, the proposed tracking method can robustly track the target object against pose changes (e.g., frames #137, #156, #171, #376), scale and appearance changes (e.g., frames #137, #456), and partial occlusion (e.g., frames #156, #171, #390). Based on the tracking result error illustrated in Figure 4.7, among other tracking methods, only ERTM (the proposed method), TLD, and IVT tracked the target accurately from the beginning to the end. However TLD failed to locate the target from frame about #110 to #220 mainly because of partial occlusion and out of plane rotation.

Sequence cube The third sequence[13] shows a cubic object which is moved randomly by a person's hands. This video is challenging due to the low

[12] `http://www.cs.toronto.edu/~dross/ivt/`
[13] `http://acis.ok.ubc.ca/~hfirouzi/RDMMATT.html`

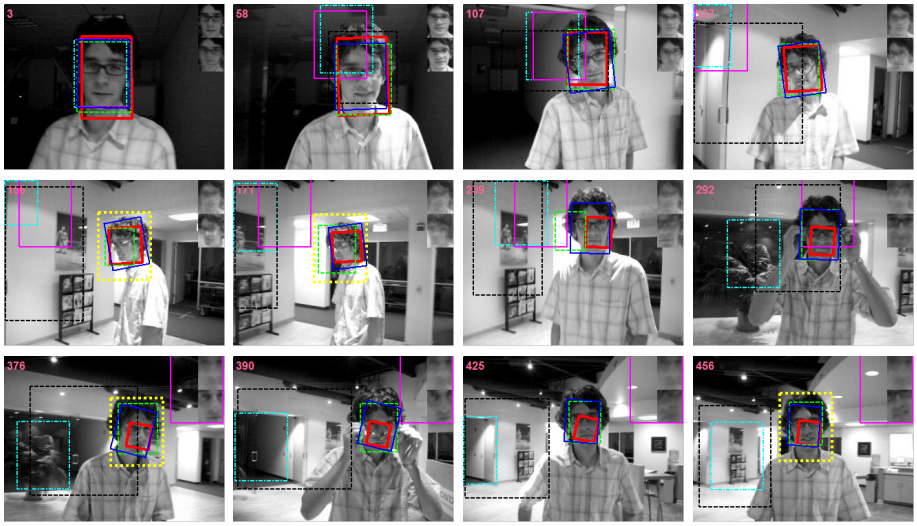


Figure 4.5: sequence *david*: the proposed tracking result (bold dashed red box) in comparison with TLD (dashed green box), Mean-shift (dash-dot cyan box), Fragment-based tracker (solid magenta box), DRTT (dashed black boc), IVT (solid blue box) and ground truth data (bold dotted yellow box)

object image contrast, having just a few unique image features, and unpredicted object motion. Shown in Figure 4.8, at some frames (e.g., #80, #169, #290) the cube is mixed with the person's hand and passing from one hand to the other; however, the proposed method is capable of tracking the object in the low-contrast scene where the object scale is also changing over time (e.g., #32, #270).

In comparison with the other methods shown in Figure 4.10, the proposed method is capable of tracking and locating the target object with more precision and accuracy.

Sequence car The forth video[14] is a low-contrast image sequence of a moving car in a cluttered and dynamic street. Illustrated in Figure 4.11, the proposed method is capable of tracking the target object which is very similar and mixed with the background (e.g., frames #55, #28, #390), whereas, according to Figure 4.13, other tracking methods could not track the target completely because it is poorly textured and very cluttered by

[14] `http://www.cs.toronto.edu/~dross/ivt/`

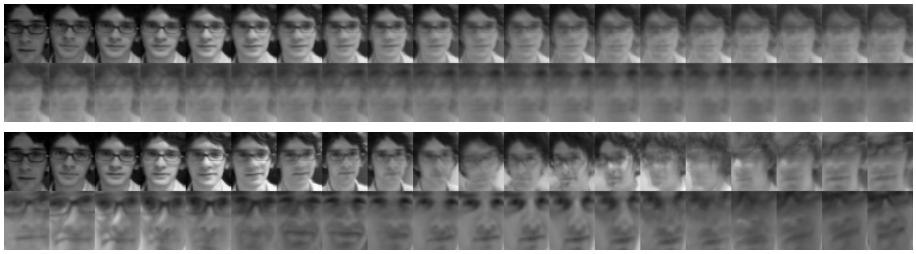


Figure 4.6: sequence *david*: changes in the mean of the long-term (top) and short-term (bottom) template over time

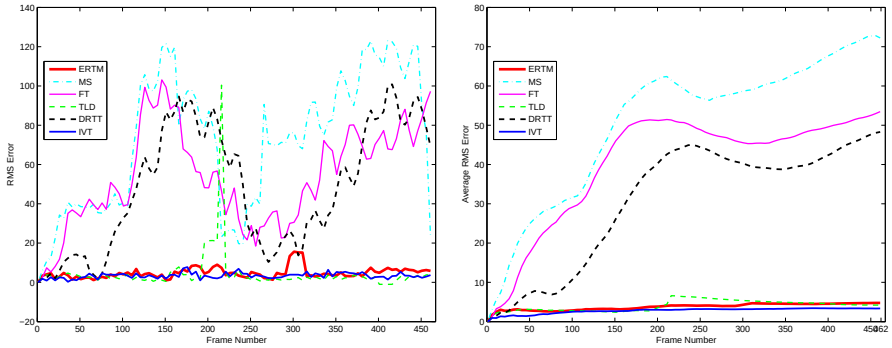


Figure 4.7: sequence *david*: the RMS error (left) and the average RMS error (right) between the ground truth data and the result of all trackers

the background.

Sequence dog The last image sequence[15] shows a doll dog which is moved randomly by a person's hand. Although the object appearance, pose, and scale is significantly changed over time (e.g., frames #927, #1028, #1270), the proposed method can accurately track the target. Other tracking methods are not robust against significant scaling e.g., frames #927, #1028.

4.4.2 Implementation

Based on the experimental results, the proposed tracking method acquired the least RMS error in the image sequences *cube* and *car*. and in the other ones its performance is comparable with the best tracker i.e., IVT (which has obtained the least RMS error).

[15] `http://www.cs.toronto.edu/~dross/ivt/`

Figure 4.8: sequence *hadi*: the proposed tracking result (bold dashed red box) in comparison with TLD (dashed green box), Mean-shift (dash-dot cyan box), Fragment-based tracker (solid magenta box), DRTT (dashed black boc), IVT (solid blue box) and ground truth data (bold dotted yellow box)

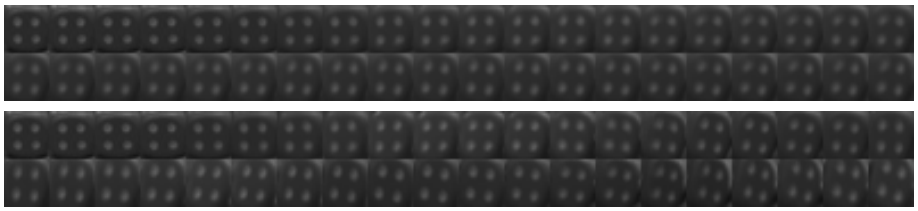


Figure 4.9: sequence *cube*: changes in the mean of the long-term (top) and short-term (bottom) template over time

In general, both Mean-shift and Scale Adaptive Mean-shift can poorly track objects whose appearance and scale are significantly changing during time (e.g., in Figure 4.7 after frame around #40). They are also not robust against illumination changes and outliers. Similarly, Fragment-based tracker cannot handle pose, scale, and illumination changes. TLD tracker, on the other hand, is robust to appearance and pose variations, illumination changes, and non-rigidity. However, it fails when there is either an occlusion, out of plane rotation, or rapid appearance change (e.g., in Figure 4.4 from frame about #250 to #380, in Figure 4.7 from frame about #200 to #220, or in Figure 4.13 after frame about #200). Also, DRTT performed well comparably in the sequences dudek, cube, car, and dog, but it is not robust to significant appearance changes and occlusion (Figure 4.7 after frame about #70).

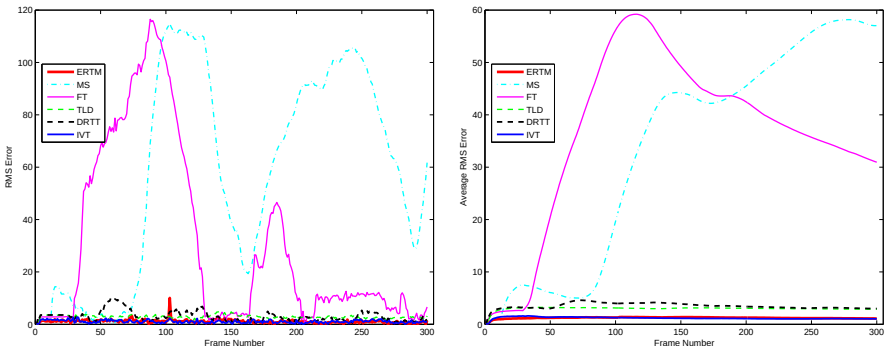


Figure 4.10: sequence *cube*: the RMS error (left) and the average RMS error (right) between the ground truth data and the result of all trackers

In addition, it is verified that the proposed tracker can more efficiently be used in real-time applications in comparison with the other methods. A Matlab(R) implementation of the proposed method can process in average 31.76[16] frames per second(fps) on a laptop with a 2.27 Ghz Intel(R) Core(TM) i3 CPU. However, a Matlab(R) / Mex implementation of the IVT and a matlab implementation of DRTT can only process in average 8.3 and 8.2 frames per second respectively. Low computational cost of the proposed tracker is mainly due to the parallel and interactive optimization algorithm described in Section 4.3.2. Based on my experiment, the average number of optimization iteration is about 1.1 when the number of starting points is set to 50 ($L = 50$). Consequently, the required particles to find the best target location is about 55 which is significantly less than that of a typical particle filter-based visual tracker[17]. Moreover, the proposed object localization algorithm consists of multiple simultaneous optimization steps which can be run in parallel on a multi-thread processor.

The videos corresponding to the experimental results can all be found at `http://www.acis.ok.ubc.ca/~hfirouzi`.

4.5 Discussions

An efficient template-based, or direct, tracking method is presented in this Chapter. The proposed tracker is robust against object appearance,

[16]The number of processed frames in a second for the sequences dudek, david, cube, car, and dog are 20.5, 8.8, 33.7, 73.7, and 21.1 respectively

[17]In general, particle filter-based methods require a minimum 100 samples (i.e., $5 \times 100 = 500$) for each estimating parameter to obtain a satisfactory tracking result.

Figure 4.11: sequence *car*: the proposed tracking result (bold dashed red box) in comparison with TLD (dashed green box), Mean-shift (dash-dot cyan box), Fragment-based tracker (solid magenta box), DRTT (dashed black boc), IVT (solid blue box) and ground truth data (bold dotted yellow box)

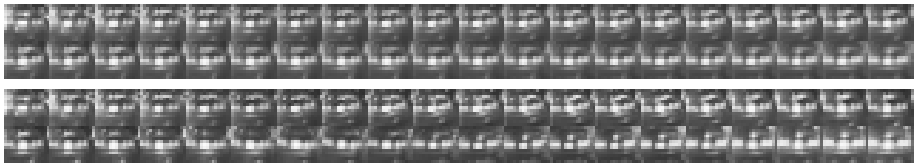


Figure 4.12: sequence *car*: changes in the mean of the long-term (top) and short-term (bottom) template over time

scale, and illumination changes as well as occlusion and cluttered environments. The representation model (i.e., object template) is defined by two heterogeneous adaptive templates consisting of Gaussian functions. A forgetting factor (γ) and an uncertainty margin (σ_0^2) are used to update the Gaussian functions to increase the robustness and adaptability of the appearance model against both long-term and short-term changes over time. In addition, the variance values of the Gaussian functions are updated so that they can reject outliers such as background pixels. At every tracking step, a mixture of Gaussian errors between the object templates and a set of candidate sub-images are minimized using several Gradient-based optimization processes. Each optimization process is initialized with a different starting point to avoid trapping in local minimum and resolve the drift problem. Moreover, the parallel optimization algorithm is designed to improve the overall computational cost of tracking by interactively comparing the

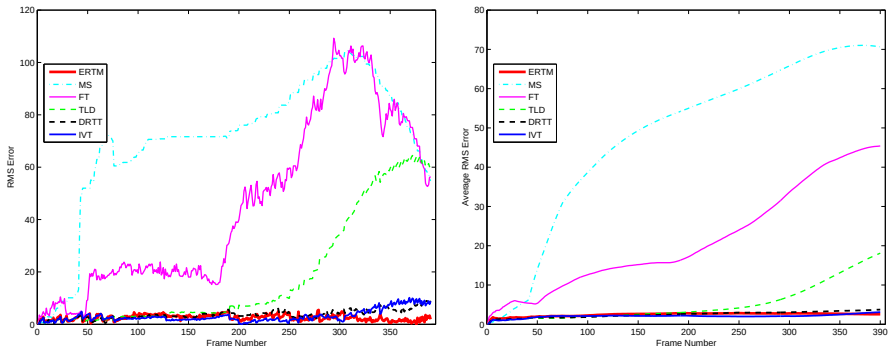


Figure 4.13: sequence *car*: the RMS error (left) and the average RMS error (right) between the ground truth data and the result of all trackers

solutions. It is shown experimentally that a non-optimized Matlab(R) implementation of the proposed tracker can on average process 31.76 frames per second and hence meets the demand of typical real-time applications.

Table 4.2: Challenging situations in different image sequences

Challenging Situation	**Image Sequence**	**Frame Number**
appearance and pose changes	dudek (Figure 4.2)	226, 962, 1134
	david (Figure 4.5)	137, 171, 376
	dog (Figure 4.14)	170, 1152, 1199
scale variations	dudek (Figure 4.2)	575, 1134
	david (Figure 4.5)	58, 171
	cube (Figure 4.8)	32, 270
	dog (Figure 4.14)	67, 1028, 1270
different illumination	david (Figure 4.5)	1, 58, 137, 376
	car (Figure 4.11)	1, 390
occlusion	dudek (Figure 4.2)	208, 367
	david (Figure 4.5)	156, 390
	dog (Figure 4.14)	1028, 1199

Shown in Table 4.2, the proposed method performed well in different challenging situations including appearance and pose changes, scale variations, different illumination, and occlusion, however, it occasionally drifted from the target particularly when the target appearance is significantly changed and it is occluded as well. For instance, in Figure 4.7 around frame #300,

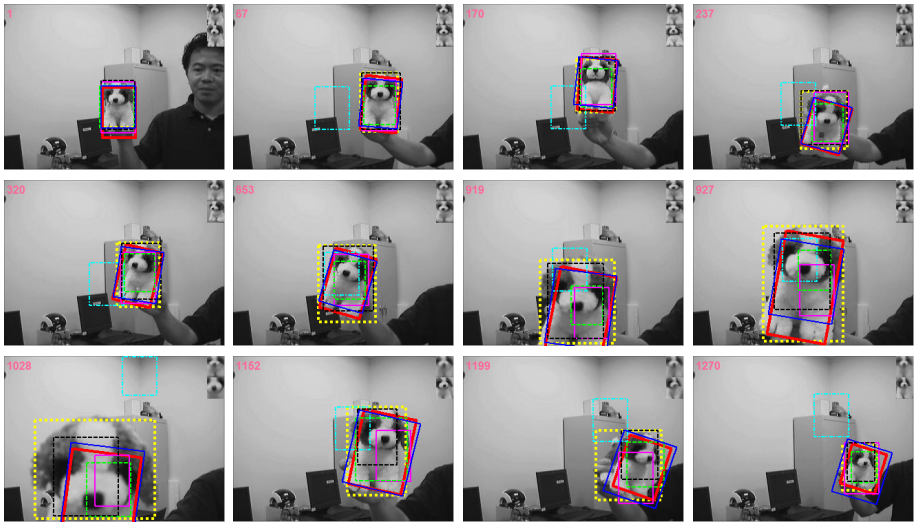


Figure 4.14: sequence *dog*: the proposed tracking result (bold dashed red box) in comparison with TLD (dashed green box), Mean-shift (dash-dot cyan box), Fragment-based tracker (solid magenta box), DRTT (dashed black boc), IVT (solid blue box) and ground truth data (bold dotted yellow box)

Moreover, it has been observed that the accuracy and robustness of the proposed tracker do not significantly depend on its parameter values (the tracking parameters are the same in all experiments, I have observed that the tracking result is fairly robust with different choice of parameters). However, different parameter values may change the overall performance of the proposed tracker. For instance, a smaller value for the maximum error of the localization optimization (opt_{max_err}) or a greater number of starting points can improve the accuracy of the tracking at the expense of an increased computational cost. In general, the tracking parameters constitute a trade-off between accuracy and efficiency.

As a limitation of the proposed tracker, only region-based models can be integrated into the proposed multi-model target representation. However, a modification of the parallel optimization process can be effectively used for both region-based and feature-based multi-model representations. Moreover, it has been observed that this method may fail to re-track the target which has been fully occluded for a long time. To solve this problem, another adaptive template which is robust against the long-term changes can be added to the proposed representation model. As a result, the target

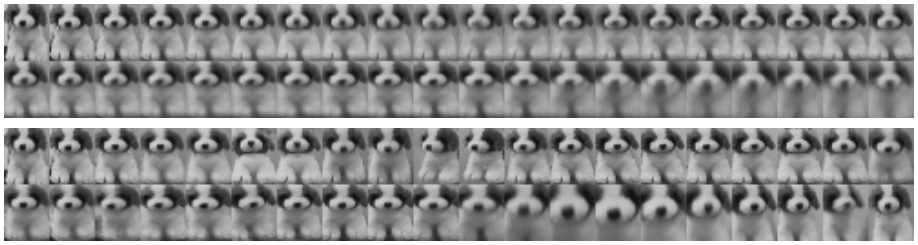


Figure 4.15: sequence *dog*: changes in the mean of the long-term (top) and short-term (bottom) template over time

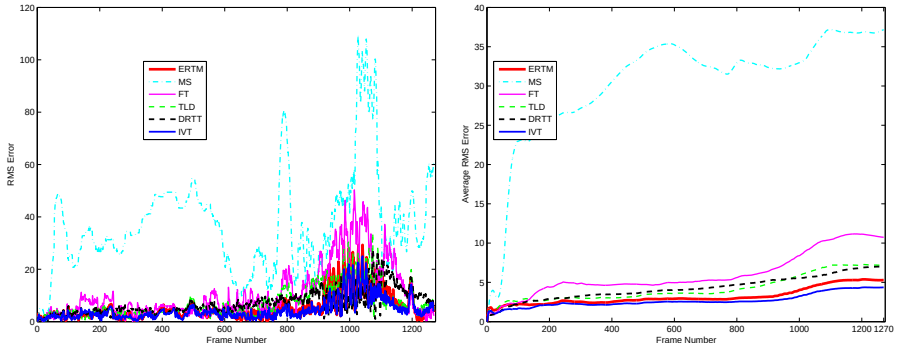


Figure 4.16: sequence *dog*: the RMS error (left) and the average RMS error (right) between the ground truth data and the result of all trackers

can also be tracked under long-term and full occlusion.

Chapter 5

Adaptive On-line Similarity Measure for Direct Visual Tracking

Visual tracking is a fundamental and essential part of many computer vision, robotic, and video analytic applications. In its simplest form, visual tracking is defined as the problem of locating three-dimensional (3D) target objects (such as a human or car) in a two-dimensional (2D) image plane as they move around a scene [99]. Besides other main parts such as target representation model and localization algorithm which have been elaborated in previous chapters, the efficiency and reliability of a tracker is also highly affected by the similarity measure method used. The main goal of a similarity measure is to estimate the distance from the target representation model and the received data or image. Usually a predefined metric such as Euclidean distance is employed to measure the distance. However, these static metrics cannot accurately and robustly estimate the similarity level over time under challenging situations such as long-term occlusion and significant appearance changes.

This chapter presents an on-line adaptive metric to estimate the similarity between the target representation model and new image received at every time instant. The similarity measure, also known as observation likelihood, plays a crucial role in accuracy and robustness of visual tracking. In this work, an L2-norm is adaptively weighted at every matching step to calculate the similarity between the target model and image descriptors. A histogram-based classifier is learned on-line to categorize the matching errors into three classes namely i) image noise, ii) appearance changes, and iii) outliers. A robust weight is assigned to each matching error based on the class label. Therefore, the proposed similarity measure is able to reject outliers and adapt to the target model by discriminating the appearance changes from the undesired outliers. The experimental results show the superiority of the proposed method with respect to accuracy and robustness

in presence of sever and long-term occlusion and image noise in comparison with commonly used robust regressors.

In the following Section 5.1, some relevant previous works are reviewed. Section 5.2 presents the proposed similarity measure where I formulate the metric and describe an on-line algorithm to train a histogram-based classifier. Next in section 5.3, the proposed adaptive metric is used in a typical template matching problem. The results obtained by my metric is compared with several robust regressors as well as manually labeled ground truth data in Section 5.4. Lastly, in Section 5.5 some conclusions and potential future works are discussed.

5.1 Related Work

A primary similarity measure used for the template matching problem is the *Euclidean distance* between the object template and the candidate sub-image. Assume T is the object template, I is the received image frame, and $W(X;P)$ is the warping function which maps every pixel $X = \{x, y\}$ in the image plane to a pixel $X^{'} = W(X;P)$ in the template based on the transformation parameters $P = \{p_1, ..p_k\}$. At every tracking time instant t, the goal of a template-based tracker is to find the best transformation parameters P^t in a way that the distance between the template T^t and the candidate sub-image I^t is minimized. Lucas and Kanade [59] used the *sum of squared difference* (SSD) to measure this distance:

$$P^t = \arg\min_P \sum_X \left[T^t(X) - I^t(W(X;P))\right]^2 \tag{5.1}$$

As illustrated in Eq. 5.1, the SSD measure can be used in conjunction with a gradient based optimization to estimate the transformation parameter. A least squared algorithm to optimize Eq. 5.1 is proposed in [59]. In general, L2 norm of errors is not robust to outliers, severe appearance variations, illumination changes, and occlusion. As a remedy for this problem, a *robust error function*, $\rho(e)$ is used to estimate the error e between the template and the candidate sub-image. Using a robust estimator instead of L2 norm, we obtain:

$$P^t = \arg\min_P \sum_X \rho\left(T^t(X) - I^t(W(X;P))\right) \tag{5.2}$$

Any function which satisfies the following criteria can be considered as a robust estimator [63]:

1. $\forall e \in \Re \ \rightarrow \ \rho(e) > 0$
2. $e_1 > e_2 > 0 \ \rightarrow \ \rho(e_1) > \rho(e_2)$
3. $e_1 < e_2 < 0 \ \rightarrow \ \rho(e_1) < \rho(e_2)$
4. $\rho(e)$ is piece-wise differentiable

A wide variety of robust error functions have been used in the literature. The *Geman-McLure* function is commonly used for the task of visual tracking [12, 82].

$$\rho(e) = \frac{e^2}{e^2 + \sigma^2} \tag{5.3}$$

Another robust estimator used for tracking [38] is the *Huber* function.

$$\rho(e) = \begin{cases} \frac{1}{2}e^2 & if|e| \leq \sigma \\ \sigma|e| - \frac{1}{2}\sigma^2 & otherwise \end{cases} \tag{5.4}$$

where in equations 5.3 and 5.4, σ is a scale parameter.

It has been shown that these functions can improve the robustness of a visual tracker against outliers and occlusion [12]. In general, a robust estimator assigns a weight to each error value based on the magnitude of the error. The weight is less when the error is large. Despite the theoretical benefits, there are two practical problems which may significantly damage the efficiency and robustness of these functions. First the robust estimator is application dependent and has to be picked by a designer for different cases. This can be an acceptable limitation for some application, it is not feasible under general conditions. Also, depending on the distribution of the error a proper scale vector (σ) has to be selected. Moreover, robust regression methods cannot distinguish between outliers and actual significant target appearance changes.

Besides the sum of squared differences and robust estimators, other metrics such as *cross cumulative residual entropy* (CCRE) [95], *mutual information* (MI) [23], *Bhattacharyya coefficient* [21], a convolution of spatial and feature space kernel functions [26], and *sum of conditional variance* (SCV) [78] have been proposed to measure the similarity of the target model and the received images. However, these methods are developed based on static and prespecified measures which cannot sufficiently deal with challenging situations in a visual tracking scenario. One challenge is that the most similar candidate sub-image to the target model may not be the best match using a predefined similarity measure. The mentioned problem mainly rise

when the target appearance changes over time or it is partially occluded by either itself or other background objects. Another phenomena which can cause a tracker to fail is the existence of similar background objects known as distracters in a close proximity to the target object. Therefore, the applicability of these predefined similarity measures are limited to specific cases.

Adaptive similarity measures, on the other hand, can be used to find the best match of the target model over time robustly. Collins et al. [19] proposed a dynamic feature selection method for estimating the similarity of the target model and the candidate image. In this method, the total number of features are fixed and the goal is to adaptively rank these features and use a subset of high ranked ones for matching. Although the method proposed in [19] can select discriminative features properly in some cases, the color features used in this method are not suitable in various applications, and also it is not always feasible to employ a more discriminative feature vector instead of color features due to the used exhaustive search for ranking the features. Recently Jiang et al. [44] proposed a classifier which is learned on-line from the tracking information to find the best match of the target model over time. In this method, an adaptive *Mahalanobis distance* is used to weight each feature in the classification process. According to the experimental results, this adaptive metric performed well in existence of distracters. However, this method may fail in case of occlusion. In general, although adaptive feature selection can improve the target matching by choosing a more discriminative subset, it is not robust against occlusion and significant appearance changes.

My proposed adaptive similarity measure differs from the works in the literature in two ways. Firstly, unlike metrics presented in [19, 44] where a subset of the feature vector is adaptively selected for matching, in my method the distance between the target and the image is modeled on-line by an adaptive hybrid model. This model is designed to reject outliers whereas it deals with appearance changes. Also, my method requires less predefined parameters in comparison with other methods such as robust regression estimation [63] where a scale vector plays a crucial role in the robustness of the regressors.

In the following Section 5.2, first the proposed similarity measure is defined, and then an on-line algorithm to train a histogram-based classifier is described in detail. Next in section 5.3, the proposed adaptive metric is used in a typical template matching problem. The results obtained by my metric is compared with several robust regressors as well as manually labeled ground truth data in Section 5.4. Lastly, in Section 5.5 some conclusions and potential future works are discussed.

5.2 Formulation

From the definition, the goal of a similarity measure is to estimate the distance between a target model and an image. In the proposed adaptive similarity measure, the euclidean distance of the target model and the image is considered as the matching error. However, unlike a typical SSD method, a histogram-based classification is learned on-line to assign a weight to each error based on its error type.

Let $A = \{a_1, .., a_m\}$ and $B = \{b_1, .., b_n\}$ be the features describing the target model and the image. Assuming that the feature space is metric, the number of features of the target and the image are the same (i.e., $m = n$), and if the features have injective relation (i.e., $a_j = b_k \Rightarrow j = k$), we can find the Euclidean error $E = \{e_1, .., e_n\}$ in the features space as:

$$e_j = a_j - b_j \tag{5.5}$$

Inspired by the work proposed in [43], I categorize the matching error E into three classes:

E_i - image noise and/or illumination variations,

E_a - target appearance changes, and

E_o - outliers and occlusion.

The first source of error, E_i, is mainly caused by either small illumination variations or some image noise which is natural in computer vision. Usually the distribution of this type of error can be modeled by a zero-mean Gaussian function as $E_i \sim N(0, \sigma_i)$. In this work, instead of a Gaussian function a symmetrical range is learned from the previous data. Unfortunately, other source of errors (i.e., E_a and E_o) cannot be easily discriminated from each other. The actual appearance changes may cause significant matching errors which are usually considered as outliers or occlusion by the conventional robust estimators [63]. However, a proper similarity measure has to reject outliers while it is adapting to the errors because of actual changes in target appearance and pose. Since in a tracking scenario, the target appearance usually changes smoothly[18] over time, I model the distribution of E_a by two adaptive ranges which are learned on-line from previous errors, and the

[18] In visual tracking, the input images are captured with a high frame per second rate e.g., 15 and also the target is usually a real-world object such as a human face; therefore, it is very unlikely that the target appearance significantly changes between two consecutive images.

outliers are identified if the error type is neither E_i nor E_a. I consider outliers as abnormal matching errors which cannot be easily modeled or predicted. In the following, the algorithm to model error type E_i and E_a are presented in detail.

Assume that each matching error e_j is quantized into Q bins where $b(e_j) \in [1, Q]$ is the bin index in the quantized space. Using k previous matching errors of feature j, I can estimate the number of times that e_j occurred in the bin index q as $h_{j,q} = \sum_{l=k-t+1}^{t} \delta[b(e_j^l) - q]$ where δ is the *Kronecker delta* function. In this work, all the features are first normalized into the range of zero and one, i.e., $\forall j \ ; \ a_j, b_j \in [0, 1]$, and accordingly, the matching errors are in the range of negative one and one i.e., $\forall j \ ; \ e_j \in [-1, 1]$. As a result, $\bar{q} = Q/2$ is the bin index corresponding to the smallest errors i.e., $b(e_j) = \bar{q} \rightarrow |e_j| < 1/Q$.

5.2.1 Finding the Range of Error Types

I propose an iterative algorithm to estimate the ranges of error types E_i and E_a. In this algorithm, the center and radius of each range is estimated in the quantized feature space. For error type E_i, the center μ_{E_i} is fixed and set to $\bar{q}$, and the radius ϵ_{E_i} is iteratively estimated based on the following algorithm. Note that the subscript j is eliminated from the equations for clarity.

Algorithm 4: Error Type E_i Range Estimation

1: $\epsilon_{E_i} \leftarrow 0$, $v \leftarrow 0$, $s \leftarrow 0$
2: **repeat**
3: $\quad v_o \leftarrow v$
4: $\quad \epsilon_{E_i} \leftarrow \epsilon_{E_i} + 1/Q$
5: $\quad s \leftarrow s + h_{\bar{q}+\epsilon_{E_i}} + h_{\bar{q}-\epsilon_{E_i}}$
6: $\quad v \leftarrow s^3/(2Q\epsilon_{E_i})$
7: **until** $v < v_o$
8: $\epsilon_{E_i} \leftarrow \epsilon_{E_i} - 1/Q$
9: $\forall q \in [-\epsilon_{E_i}, \epsilon_{E_i}] \ ; \ h_{\bar{q}+q} \leftarrow 0$

In Algorithm 4, the range of E_i is expanded symmetrically until the ratio of the number of occurrence and the radius are not increased. As it is shown in Algorithm 4 at step 5 and 6, the error range is expanded until the new ratio of third power of the number of points and the error range is not increased.

There are two ranges for the error type E_a which are estimated using the following algorithm. This algorithm is repeated two times to obtain both ranges E_a^1 and E_a^2. Unlike the previous algorithm, here the center of each range is not fixed and set to the maximum occurrence value.

Algorithm 5: Error Type E_a Range Estimation

$q^* \leftarrow \arg\max_q h_q$
$\alpha \leftarrow q^*$, $\beta \leftarrow q^*$, $v \leftarrow 0$, $s \leftarrow 0$
repeat
  $v_o \leftarrow v$
  if $h_\alpha > h_\beta$ **then**
    $\alpha \leftarrow \alpha - 1/Q$
    $s \leftarrow s + h_\alpha$
  else
    $\beta \leftarrow \beta + 1/Q$
    $s \leftarrow s + h_\beta$
  end if
  $v \leftarrow s^3/(Q \times (\beta - \alpha + 1))$
until $v < v_o$
$\mu_{E_a} \leftarrow (\beta + \alpha)/2$
$\epsilon_{E_a} \leftarrow (\beta - \alpha + 1)/2$
$\forall q \in [\alpha, \beta]\ ;\ h_q \leftarrow 0$

Shown in Algorithm 5, in case of error type E_a the number of errors occurred in those bins which are related to each range is set to zero after estimating the center and radius of the range, therefore, the error type ranges are not overlapped. In the next section, these ranges are used to calculate the weight of each matching error $w(e_j)$.

5.2.2 Estimating the Matching Error Weights

At every matching step, the matching error of each feature e_j is compared with the error type ranges and accordingly a weight $w(e_j)$ is obtained.

$$w(e^t) = \begin{cases} 2 & if\ |e_j| < \eta Q \epsilon_{E_i} \\ 1 & if\ |e_j - \mu_{E_a^1}| < \eta Q \epsilon_{E_a^1} \\ 1 & if\ |e_j - \mu_{E_a^2}| < \eta Q \epsilon_{E_a^2} \\ 0 & otherwise \end{cases} \tag{5.6}$$

where $\eta = 2/m \times \sum_j \delta(w(e_j))$ is two times the previous outliers percentage and m is the number of features.

Illustrated in Eq. 5.6, the adaptive weight $w(e^t)$ is robust against outliers and occlusion. Moreover, the errors caused by small illumination variations receive a higher weight in comparison with those of the appearance changes to improve the accuracy of the method.

Obtaining the weights of matching errors, we can calculate the similarity distance $S : \Re^n \times \Re^n \to \Re$ between the target model and an image as:

$$S(A, B) = \sum_j w(e_j) \times {e_j}^2 \tag{5.7}$$

In the following sections, the proposed similarity measure is formulated along with a commonly used template-based tracking.

5.3 Template Tracking using the Adaptive Similarity Measure

In this section, the proposed similarity measure is applied to a typical template-based tracker. In this method, the object is represented by a dynamic template which is updating every k frame using the new received images. As opposed to the conventional template trackers, in this method a *condensation-like* sampling algorithm [42] is used to locate and track the target at every image frame. In the following subsections, the tracking algorithm followed by the representation model are described in detail.

5.3.1 Template Representation

In conventional template-based tracking method, the target object is simply represented by its sub-image region obtained from the first image I^1, i.e., $T^1(X) = I^1(W(X;P));\ X \in R^1$ where $W(X;P)$ is the warping function and R^1 is the object region at time step $t = 1$. The function $W(X;P)$ maps the image pixel at location $X = \{x, y\}$ from the candidate sub-image into the reference model using an affine transformation consisting of six variables $P = \{t_x, t_y, \theta, s, \alpha, \phi\}$ which are x and y translations, rotation angle, scale, aspect ratio, and skew direction, respectively. There are different methods to update the template over time. One option is to not change the template [59] i.e., $T^t = T^1$ which performs poor in case of appearance and illumination changes, the second way is to update the template every frame i.e., $T^t = T^{t-1}$ called *naive* update [62]. This approach is also not stable because of

drift problem[19]. In this work, the template is updated every k frames based on a forgetting factor λ. Therefore, the model can simply represent the target appearance changes while it is robust against the drift problem.

$$T^t(X) = \frac{\lambda(t-k)}{\lambda(t-k)+k} T^{t-k-1}(X) + \frac{k}{\lambda(t-k)+k} \bar{I}(X) \tag{5.8}$$

where λ and k are empirically set to 0.97 and 5 respectively for all experiments, and $\bar{I}$ is the average value of k most recent object image.

$$\bar{I}(X) = \frac{1}{k} \sum_{j=t-k}^{t} I^j(W(X; P^j)) \tag{5.9}$$

5.3.2 Particle Filtering and Tracking

Visual tracking can be viewed as a sequential inference task in a Markov model with hidden state variables P^t describing the object motion parameters at time step t. Given an image sequence $\mathbf{I} = \{I^1, ..., I^t\}$ and reference models (in this case object templates) $\mathbf{T} = \{T^1, ..., T^t\}$, the hidden state variables can be estimated based on Bayes' theorem as follows:

$$p(P^t|I^t; T^t) \propto \; p(I^t|P^t; T^t) \int p(P^t|P^{t-1}) \; p(P^{t-1}|I^{t-1}; T^{t-1}) dP^{t-1} \tag{5.10}$$

where $p(P^t|I^t; T^t)$, $p(I^t|P^t; T^t)$, $p(P^t|P^{t-1})$, and $p(P^{t-1}|I^{t-1}; T^{t-1})$ are the posterior probability, observation likelihood, dynamical or motion model between two states, and prior probability respectively.

The proposed adaptive similarity measure is used to define the observation likelihood.

$$p(I^t|P^t; T^t) = \exp\left(-\frac{S(T^t, \tilde{I}^t)}{\sigma_c}\right) \tag{5.11}$$

where in this work, the condensation algorithm variance σ_c is set to 0.2, and $\tilde{I}^t(X) = I^t(W(X; P^t))$ is the transformed candidate image.

[19] It is the problem of updating the target model using unrelated information such as background pixels [62]

5.3.3 Sampling Algorithm

Modeling the observation likelihood using the proposed similarity measure in the previous subsection, I aim to approximate the posterior distribution $p(P^t|I^t;T^t)$ defined in Eq. 5.10 using a condensation-like sampling algorithm[42].

Assume that the prior distribution $p(P^{t-1}|I^{t-1};T^{t-1})$ is approximated by N samples (or particles) with corresponding weights $(\{P_n^{t-1},\pi_n^{t-1}\}_{n=1}^N)$. The first step is to randomly choose N samples (with replacement) from the set $\{P_n^{t-1}\}$ based on the probability $\{\pi_n^{t-1}\}$. As a result, those samples with high weight may be selected several times. In the next step, known as diffusion, each sample undergoes a Brownian motion using a Gaussian distribution usually with a diagonal covariance matrix. The weights $\{\pi_n^t\}$ of the new sample set $\{P_n^t\}$ are obtained as:

$$\pi_n^t = p(I^t|P_n^t;T^t) \tag{5.12}$$

As a result, the best transformation parameters P^t is thus the particle corresponding to the maximum sampling weight.

$$P^t = \arg\max_{P_n^t}\left[p(P_n^t|I^t;T^t)\right] = \arg\max_{P_n^t}\left[\pi_n^t\right] \tag{5.13}$$

In the next section, the accuracy and robustness of the proposed method is evaluated using several challenging videos.

5.4 Experimental Results

In this section, the performance of the proposed adaptive similarity measure is validated using several experiments. In addition to the ground truth data, my experimental results have been compared with L2-norm measure and four other robust regression methods (i.e., $\sigma = \{0.3, 0.4, 0.5, 0.6\}$). For fair comparison, the same target representation model and tracking algorithm described in Section 5.3 is used with different similarity measures, in addition, the tracking parameters are kept the same in all experiments, and also different scaling vectors are used for the robust regression method to obtain the best result. Comparing with other similarity measures, not only my method is more robust against sever occlusion and outliers, but also it can handle non-uniform illumination and appearance changes.

In the following subsections, the experimental results using five challenging gray-scale image sequences are illustrated. These videos and the related

ground truth data are publicly available and considered as a benchmark in the literature.

5.4.1 Qualitative Comparison

In this section, the tracking result obtained by the proposed similarity measure is qualitatively compared with L2-norm, four other regressors, and ground truth data. In the following figures, the target bounding box obtained by the proposed adaptive measure (AM - red solid box), L2-norm (L2 - green dashed box), robust regression with scale parameter $\sigma = 0.3$ (R3 - pink dashed box), robust regression with scale parameter $\sigma = 0.4$ (R4 - cyan dashed box), robust regression with scale parameter $\sigma = 0.5$ (R5 - yellow dashed box), robust regression with scale parameter $\sigma = 0.6$ (R6 - black dashed box), and the ground truth data (white dotted-dashed box) are illustrated. The object template corresponding to each method is shown at the bottom of each image. The mask image of outliers and appearance changes obtained by the proposed method is also shown at top-right of the image. In the mask image, the black and gray pixels are outliers and appearance changes respectively.

dollar sequence

The first video[20] consists of 326 image frames with the resolution of 320×240, and the target object is a dollar paper moving in a simple background. This video is selected for my experiments because it contains self-occlusion and similar objects, known as distracters.

Illustrated in Figure 5.1, at early stages (e.g., frames #48 and #56) the dollar is bended and a part of it is self-occluded, after that the target is moved in a close proximity of a similar object (e.g., frames #131 and #251). In this experiment, my method performed well against self occlusion and distracter and L2-norm failed to handle outliers. Also, all regression methods could accurately track the target in entire video.

faceocc sequence

Second sequence[21] is a long video containing 884 gray-scale images with the resolution of 352×348 where a human face is occluded several times by a book. This video is considered as one of the challenging benchmarks for occlusion handling task due to the long-term and significant occlusion.

[20] Taken from `http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml`
[21] Taken from `http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml`

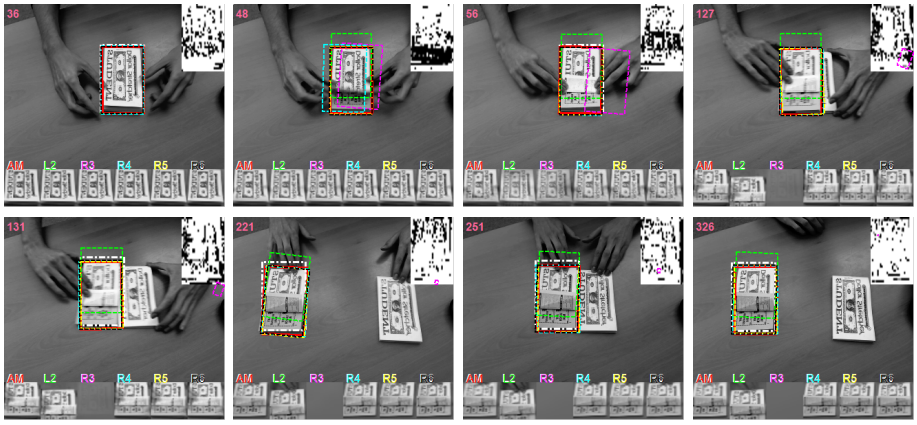


Figure 5.1: sequence *dollar*: the target bounding box obtained by the proposed adaptive measure **AM** (red solid box), L2-norm **L2** (green dashed box), robust regressors with $\sigma = 0.3$ **R3** (pink dashed box), $\sigma = 0.4$ **R4** (cyan dashed box), $\sigma = 0.5$ **R5** (yellow dashed box), $\sigma = 0.6$ **R6** (black dashed box), and the ground truth (white dotted-dashed box)

In *faceocc* sequence, shown in Figure 5.2, the target is a human face in a simple background, however around 80 percent of images contains occlusion and at some frames only a very small part of the target is visible (e.g., frame #571, #711, and #832). Due to the sever and long-term occlusions existed in this sequence, the target model shown in the bottom of each image is most of the time corrupted with irrelevant pixels and cannot correctly represent the human face. However, the proposed method was able to accurately and robustly track the target. The second best is **R5** whose accuracy is far from my method while others largely drift from the target at frame around #581

faceocc2 sequence

In the next image sequence[22] also the target object is a human face and it consists of 812 gray-scale image frames with the resolution of 321×295. The difference between *faceocc2* and *faceocc* sequences is that in the latter the face is almost stationary with a simple background, however it the former, the face is moving in a cluttered background with similar pixel values. In addition, this face appearance and orientation change a lot over time while the target is significantly occluded by different objects. Thus,

[22] Taken from `http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml`

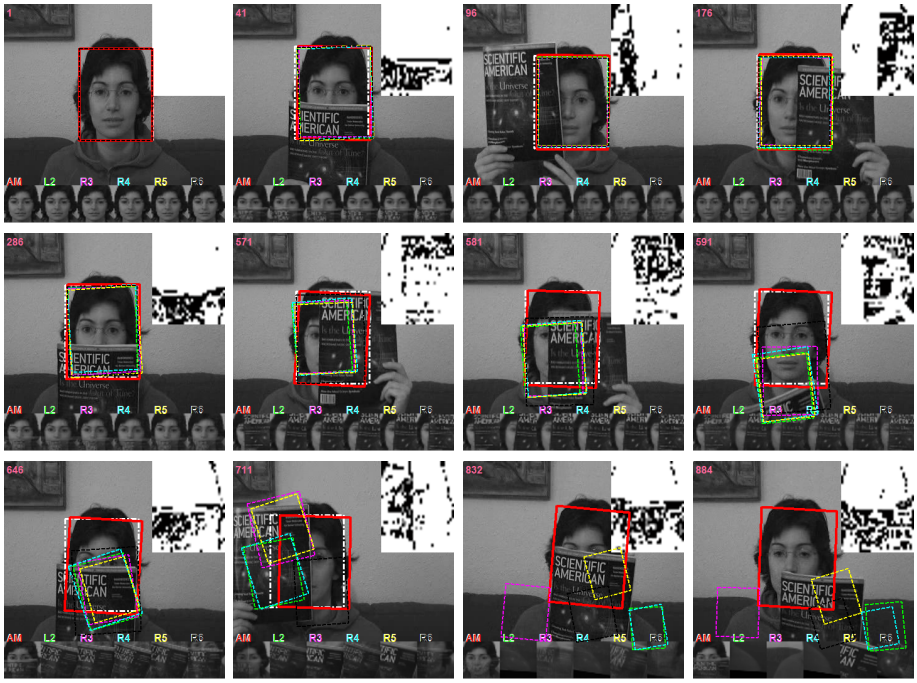


Figure 5.2: sequence *faceocc*: the target bounding box obtained by the proposed adaptive measure **AM** (red solid box), L2-norm **L2** (green dashed box), robust regressors with $\sigma = 0.3$ **R3** (pink dashed box), $\sigma = 0.4$ **R4** (cyan dashed box), $\sigma = 0.5$ **R5** (yellow dashed box), $\sigma = 0.6$ **R6** (black dashed box), and the ground truth (white dotted-dashed box)

the template matching error is not only because of occlusion but also due to the appearance changes.

Illustrated in Figure 5.3, the human face is occluded by a book several times specifically when the target is rotated (e.g., frames #416 and #491). The coexistence of occlusion and appearance variation is also happened at frames #575 and #700. According to the experimental results, my similarity measure outperformed other methods. The regression estimations with $\sigma = [0.4, 0.5]$ (R4 and R5) could also tracked the target up to the end of the sequence, but their accuracy was not as good as the proposed method.

Figure 5.3: sequence *faceocc2*: the target bounding box obtained by the proposed adaptive measure **AM** (red solid box), L2-norm **L2** (green dashed box), robust regressors with $\sigma = 0.3$ **R3** (pink dashed box), $\sigma = 0.4$ **R4** (cyan dashed box), $\sigma = 0.5$ **R5** (yellow dashed box), $\sigma = 0.6$ **R6** (black dashed box), and the ground truth (white dotted-dashed box)

david sequence

This video[23] contains 462 gray-scale images with the resolution of 321×295. The target object is a face which is moved in a cluttered background. The appearance, scale, and orientation of the face are changed during the tracking, and also the target is self occluded at some image frames. The new challenge in this sequence comparing with the previous ones is that in this experiment, the target encounters different lighting conditions as well as appearance variations. Although the simple template tracking method used in this work cannot robustly represent the object appearance, the proposed similarity measure is capable of increasing the tracking robustness and accuracy against illumination, scale, appearance variation as well as outliers and occlusions.

Based in Figure 5.4, the target object, human face, is moved in a room

[23]Taken from `http://www.cs.toronto.edu/~dross/ivt/`

Figure 5.4: sequence *david*: the target bounding box obtained by the proposed adaptive measure **AM** (red solid box), L2-norm **L2** (green dashed box), robust regressors with $\sigma = 0.3$ **R3** (pink dashed box), $\sigma = 0.4$ **R4** (cyan dashed box), $\sigma = 0.5$ **R5** (yellow dashed box), $\sigma = 0.6$ **R6** (black dashed box), and the ground truth (white dotted-dashed box)

under different lighting situations e.g., frame #30 and #86. The half of the target is then occluded by itself at around frame #161. From this time step, all other methods started drifting from the target due to the significant appearance changes and occlusion, but my method could manage to reject outliers and matching errors because of large appearance changes and track the face. However, because of poor performance of template tracking methods in case of non-rigidity and simultaneous appearance and illumination changes, the proposed similarity measure failed to match the template model with the target from frame around #276.

trellis70 sequence

The last sequence[24] is a video containing 501 gray-scale images with the resolution of 321 × 295. The target object is a face which is moved in a cluttered background. The appearance, scale, and orientation of the face are changed during the tracking, and also the target is self occluded at some image frames. The new challenge in this sequence comparing with

[24]Taken from `http://www.cs.toronto.edu/~dross/ivt/`

the previous ones is that in this experiment, the target encounters different lighting conditions as well as appearance variations. Although a typical template tracking method used in this work cannot robustly represent the object appearance, the proposed similarity measure can increase the tracking robustness and accuracy against illumination, scale, appearance variations as well as outliers and occlusions.

Figure 5.5: sequence *trellis70*: the target bounding box obtained by the proposed adaptive measure **AM** (red solid box), L2-norm **L2** (green dashed box), robust regressors with $\sigma = 0.3$ **R3** (pink dashed box), $\sigma = 0.4$ **R4** (cyan dashed box), $\sigma = 0.5$ **R5** (yellow dashed box), $\sigma = 0.6$ **R6** (black dashed box), and the ground truth (white dotted-dashed box)

Illustrated in Figure 5.5, the target appearance is partially changed because of different lightings e.g., frames #38, #78, #123, and #155. Some parts of the target are occluded due to the out of plane rotation of the face in around frame #178. All trackers except my method failed to accurately track the object from frame around #78. However, the proposed similarity measure could find the correct match to the target template in coexistence of severe illumination changes and outliers up to frame #229. It is expected that employing my method along with an illumination invariant representation model can improve the robustness and accuracy of the tracker against significant illumination changes and partial occlusions. Moreover, the target model can be updated considering the outliers detected by the proposed method. This way, irrelevant information such as background pixels and

occluding objects will not damage the target model over time.

In the following subsection, the ground truth data is used to provide a quantitative evaluation of the results obtaining by all methods.

5.4.2 Quantitative Comparison

In addition to the qualitative analysis illustrated in the previous section, the tracking results of all methods are compared with the ground truth data to show the accuracy and robustness of each method. In the following figure, the RMS (root mean squared) error and the average RMS error of the target bounding box obtained by each method and the ground truth data are shown. Similar to the above figures, the results corresponding to my method, L2-norm, robust regression with $\sigma = 0.3$, $\sigma = 0.4$, $\sigma = 0.5$, and $\sigma = 0.6$ are specified by a red solid line, a green dashed line, a pink dashed line, a cyan dashed line, a yellow dashed line, and a black dashed line respectively.

Figure 5.6 shows the RMS error and the average RMS error of each method in comparison with the ground truth data. Illustrated in this figure, all methods expect R3 (robust regression with $\sigma = 0.3$) and L2 could accurately track the target in sequence *dollar*. R3 (pink dashed) and L2 (green dashed) methods drift form the target at around frame #50 because of partial occlusion. In sequence *faceocc*, only my method could track the target up to the end of the sequence. The second best method is R6 which failed to correctly locate the target at frame around #480, other ones lost the face earlier at frame around #520. Similarly in sequence *faceocc2*, the proposed method outperformed other methods. In this experiment, R4 and R5 could also track the target up to the end with less accuracy in comparison with my method. The forth sequence, *david*, involves significant appearance and illumination variations which generally cause a typical template tracker to fail. However, the proposed method could robustly track the target up to frame around #280 while R3 and L2 failed at early stages (i.e., at frames #25 and #90 respectively), also, R4, R5, and R6 started to drift from the target at frame around #160 due to simultaneous illumination changes and out of plane rotation. In the last sequence *trellis70*, similar to the previous one, there are frequent illumination, appearance, scale, and orientation changes which makes it difficult to obtain a precise template of the target object. In this sequence, although none of the methods could accurately track the target in entire image frames, the proposed similarity measure was capable of rejecting outliers whereas adapting the appearance and illumination changes at the same time in several challenging situations

e.g., in Figure 5.5 from frames #145 to #207. All other methods failed to track the target at frame around #145.

5.4.3 Implementation

From the quantitative results illustrated in Figure 5.6, the proposed method outperformed all other similarity measures in most times and could adaptively identify and reject outliers in my experiments. The reason for high accuracy and robustness of the proposed method include using a hybrid model for estimating the matching error distribution, and next, an on-line classification and auto-tuning mechanism used for parameter training. In addition, all parameters of my similarity measure are kept the same in all sequences, as a result, my method is fairly robust to the choice of parameters and can work accurately and robustly without any modification. In general, among different scaling factors, the robust regression with $\sigma = 0.5$ could generally handle outliers more accurately in comparison with other ones. However, in sequence *faceocc* R6 obtained less RMS error than R5. Also as expected, L2-norm performed poorly in all experiments due to its weakness in rejecting outliers and occlusion.

Although the proposed method can accurately find the best match to the target model against outliers, its RMS error is not small at some frames. One reason for this phenomena is that the manually generated ground truth data are approximately precise and subject to the human error, it is expected that the RMS error of the proposed method in comparison with a more accurate ground truth data can be less in several cases. For instance, in Figure 5.4 in frames #161, #176, #181, and #201 the tracking results obtained by my method (solid red box) are obviously more accurate that those of the ground truth data (dotted dashed white box). Moreover, a typical template cannot robustly represent a non-rigid target whose appearance and illumination are significantly changed over time. Therefore, the proposed similarity measure is able to improve the accuracy and robustness of an advanced visual tracking method in case of sever outliers and long-term occlusions.

5.5 Discussions

This chapter presented a robust similarity measure which can adaptively learn the matching error type using on-line classification. The proposed method is capable of categorizing the error into three classes: 1) small variations of the target illumination and appearance, 2) significant changes in

target appearance, and 3) abnormal errors because of outliers and occlusion. According to the error types, an image mask is generated to assign a weight to each matching error. The normalized weighted errors are then used to find the best match to the target model. As an advantage to other comparable methods, my similarity measure is able to adapt its regression parameters over time.

The accuracy and robustness of my method have been compared with several commonly used robust regression methods. The proposed method is able to find the best match to the target template in different challenging situations including partial illumination variations, significant appearance changes, and long-term occlusion. It is observed that my proposed method excels all the regressors including **R5** (regression method with scaling factor 0.5) which has performed better than the others.

A new direction of this work is to use the outliers masking by the proposed similarity measure to update the target representation model. However, a matched sub-image with proportionally high percentage of occlusion may not be a proper information for updating the target model. In addition, using my method, I can approximate the start and end of an occlusion which is useful for generating a temporary representation model.

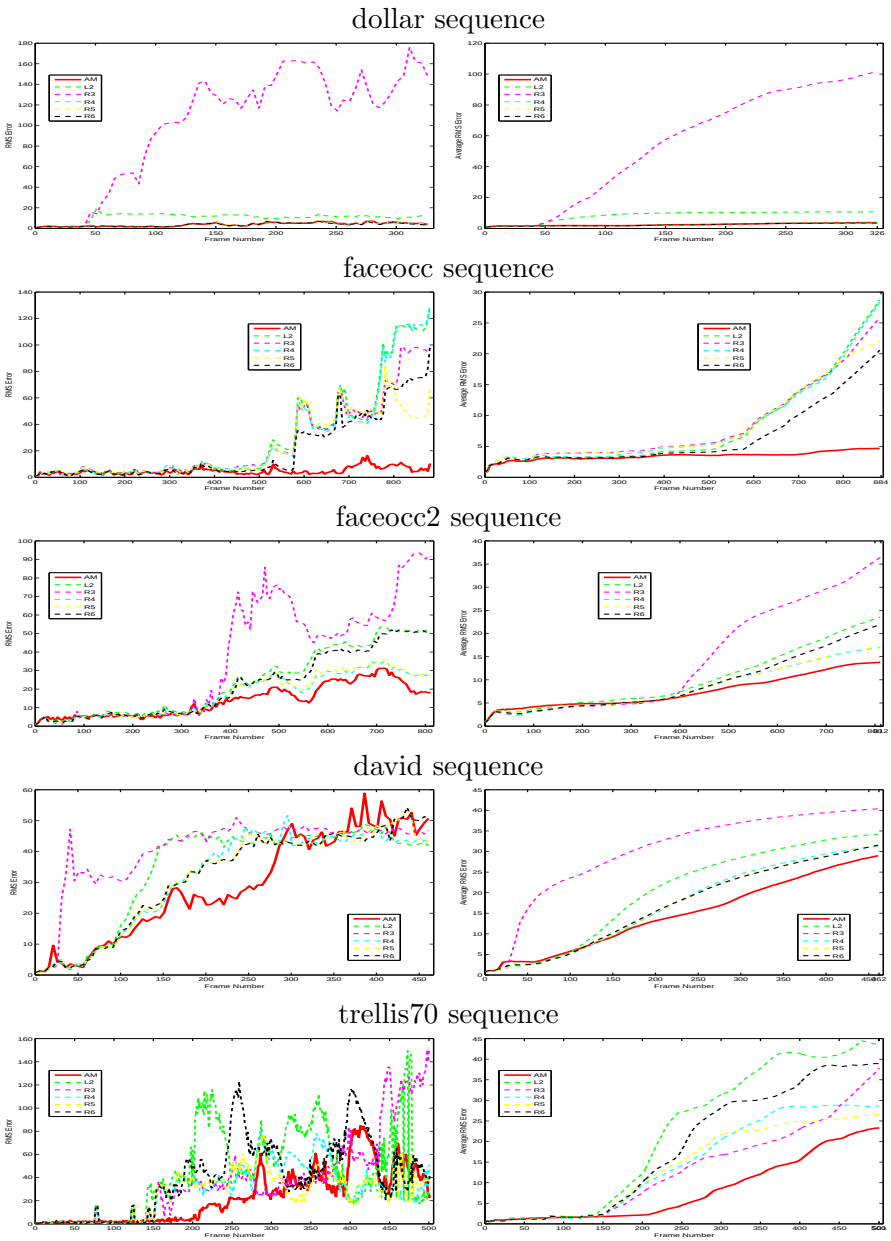


Figure 5.6: the RMS error (left column) and the average RMS error (right column) corresponding to the proposed adaptive measure **AM** (red solid line), L2-norm **L2** (green dashed line), robust regressors with $\sigma = 0.3$ **R3** (pink dashed line), $\sigma = 0.4$ **R4** (cyan dashed line), $\sigma = 0.5$ **R5** (yellow dashed line), and $\sigma = 0.6$ **R6** (black dashed line)

Chapter 6

Conclusions

This thesis presented several computer vision techniques for tracking real-world and non-rigid objects of interest in a video. The focus of this research has been generally on improving the accuracy, robustness, and efficiency of visual tracking by proposing advanced computer vision techniques including decentralized and multi-model representation models, multi-start and interactive localization algorithms, and on-line similarity measure. It is assumed that the target object encounters several challenging situations such as appearance, scale, orientation, shape, and pose changes, illumination variations, outliers, and occlusion. A typical visual tracking method consists of three main components namely the *target representation model*, *localization algorithm*, and *similarity measure*. Based on the model representing the target, visual tracking can be categorized into two wide classes: region-based and feature-based approaches. Although some robust features have been introduced in the literature [10, 58, 66] they are highly generalized and hampered in real-time applications due to their computational complexities. Moreover, feature-based approaches introduce another challenging problem which is known as point correspondence and data association problem. In contrast, in region-based approaches both the visual and spatial information are available for modeling and tracking the target without a time consuming preprocessing step. In this thesis, two robust region-based trackers and an adaptive metric for similarity measurement are proposed. It has been shown that the proposed techniques outperform the commonly used state-of-the-art visual trackers when they applied on several publicly available benchmarking videos.

In this chapter, all the proposed techniques including two visual tracking methods and the similarity measure are first summarized, and then the advantages and limitations are discussed followed by the guidelines for future work.

6.1 Robust Decentralized Multi-Model Adaptive Template Tracking

In the first tracker proposed in Chapter 3, the target object is represented by a decentralized multi-template model. Each part of this model consists of two adaptive templates namely immediate template and delayed template which encode the short- and long-term appearance variations of the corresponding target sub-region. Also a two-step gradient-based optimization and a robust fusion are sequentially used to track the target at every tracking step. In the proposed localization method, first the short-term template is used for coarse sub-region localization, and then its final location is obtained based on the long-term template. Multiple starting points are input to a parallel optimization process to estimate both coarse and fine sub-region location estimations. Lastly, an outlier-resistant method is used to fuse the sub-region locations to track the target.

6.1.1 Research Contributions and Advantages

The specific features of the first proposed method include:

1. Partitioning a non-rigid target into smaller regions known as sub-regions where each region can be tracked efficiently as a rigid body. In general, partitioning the target region can improve the accuracy of appearance modeling by piece-wise representation of a complex and non-rigid object. Decentralized modeling is also robust against outliers and occlusion. For instance, in the case of occlusion the non-occluded sub-regions can be effectively used to track the target.

2. Representing each sub-region by a heterogeneous Gaussian-based multi-template model: The proposed immediate and delayed templates make the representation model valid under different time-varying appearance and shape changes. Compared to the conventional trackers, the proposed method is more robust against significant appearance changes and short-term occlusion.

3. Robust sub-region localization based on a two-step coarse to fine gradient-based optimization: The proposed localization method can solve the drift problem which is a common problem in template tracking.

4. Efficient and simultaneous search using different starting points: Multi-start gradient-based search significantly decreases the probability of

trapping in a local minimum, but at the same it efficiently estimates the target location.

5. Target tracking by robust fusion of new sub-region locations: Multi-layer and decentralized localization improves the accuracy and robustness of the ultimate target location by reducing the noise impact and rejecting the outliers.

6.1.2 Discussions and Future Work

Although the proposed method performed well to a great extent, there are some limitations which need to be considered. First, the size of each partition should be large enough to represent a meaningful part of the target which contains sufficient visual information to track. It is observed that both large and small partition size will compromise the performance of the proposed tracker. Having said that, there is a safe partition size range for many applications. For instance, in the experiments demonstrated in Chapter 3 the partition size is fixed (i.e., 22×22 pixels) for all of the videos. It is also observed that the proposed tracker occasionally drifted from the target when both the appearance and location of the target have changed significantly. This problem can be mainly fixed by increasing the number of starting points used for sub-region localization and number of optimization iterations. It is noted that a large number of optimization steps will increase the computational cost. Thanks to the multi-template model, the proposed tracker is robust to short-term occlusion. However, in the case of long-term and full occlusion when the representation model is updated by invalid information for a proportionally longer period of time the tracker may lose the target. A solution for the latter problem is to decrease the updating rate of the delayed template, so that it represents longer appearance changes.

6.2 Efficient and Robust Multi-Template Tracking Using Multi-start Interactive Gaussian-based Optimization

Similar to the first proposed method, in the second method presented in Chapter 4 the target is represented by short- and long-term Gaussian-based templates. However, the Gaussian functions are adaptively updated based on a forgetting factor and an uncertainty margin considering the tracking time step. In this method, the target is located based on an interactive multi-

start gradient-based search. Unlike the first method where a translational transformation is used, this search uses a more general transformation. Also, in the latter optimization algorithm the starting points are initialized by a sampling-like algorithm in a probabilistic framework. In addition, in this method both short- and long-term templates are used at the same time to estimate the target location which significantly improve the efficiency and accuracy of tracking. This method has been compared with several state-of-the-art trackers as well as ground truth data to evaluate its performance.

6.2.1 Research Contributions and Advantages

The specific features and advantages of the second proposed tracker are as follows:

1. Flexible multi-model target representation: Although in the second tracker only short- and long-term templates are used for target representation, different models can be also added to the proposed representation model. In general, a multi-model representation can accurately and robustly handle challenging situations such as significant appearance and shape changes.

2. Robust template updating algorithm: A combination of tracking time step, a forgetting factor, and an uncertainty margin are used to update the mean and variance of the Gaussian functions. An important advantage is that the algorithm is not sensitive to the parameters.

3. Efficient and interactive multi-start optimization: Parallel search in different time-varying templates using multiple starting points can improve accuracy, robustness, and efficiency of the target localization.

6.2.2 Discussions and Future Work

As a limitation of the second proposed tracker, only region-based models can be integrated into the proposed multi-model target representation. However, a modification of the parallel optimization process can be effectively used for both region-based and feature-based multi-model representations. Moreover, it has been observed that this method may fail to re-track the target which has been fully occluded for a long time. To solve this problem, another adaptive template which is robust against the long-term changes can be added to the proposed representation model. Therefore, the target is tracked under long-term and full occlusion.

6.3 Adaptive On-line Similarity Measure for Direct Visual Tracking

In addition to the previous proposed robust template-based tracking methods, in Chapter 5 an on-line similarity measure for finding the best match of the target model is presented. In this method, a histogram-based classifier is learned on-line to categorize the matching errors into three classes: i) small appearance variations, ii) actual significant appearance changes, and iii) abnormal changes because of outliers and occlusion.

The proposed measure is robust against abnormal matching errors due to the outliers, severe illumination changes, and long-term occlusion. Extensive experimental results verify the accuracy and robustness of the proposed similarity measure in comparison with the other commonly used robust regression methods.

6.3.1 Research Contributions and Advantages

The main contribution of the proposed similarity measure is on-line training of a classifier to assign a robust weight to each matching error. As a result, the proposed adaptive metric is robust against abnormal matching errors due to the outliers, severe illumination changes, and long-term occlusions. In addition, the proposed adaptive metric can be integrated into any commonly used visual trackers to improve their robustness against outliers and long-term occlusions.

6.3.2 Discussions and Future Work

In the proposed similarity measure the robust weights are generated based on the Euclidean distance between the target model and the candidate image, thereby, region-based trackers are the best fit for the proposed metric. However, the on-line classifier training algorithm can be modified to use other distances such as Mahalanobis distance or a non-linear distance. Moreover, the robust weights can be used to filter out irrelevant information when the target model is updated.

6.4 Thesis Impact

In view of recent advances in high-tech areas, computer vision plays a crucial role in accelerating progress towards fully automated and intelligent

systems. The core of many real-world computer vision applications is developed based on a visual object tracking method. In this thesis, two robust and efficient region-based trackers and an adaptive similarity measure are proposed. These methods can provide opportunities for introducing new applications as well as effectively improve the performance of the existing computer vision applications such as automatic visual surveillance, human behavior and activity analysis, vehicle navigation and tracking, medical image processing and diagnostics, and automatic quality assurance and control.

The highlighting contributions of this study can be summarized as follows:

- A robust decentralized multi-model target representation capable of accurately modeling a non-rigid object under various challenging conditions over time was proposed.
- An efficient multi-start interactive target localization algorithm which can search for the best solution in multiple models simultaneously was introduced.
- Last but not the least, an on-line and significantly robust metric to estimate the similarity measure between the target model and new received images was presented.

Bibliography

[1] Adam, A., Rivlin, E., Shimshoni, I., 2006. Robust Fragments-based Tracking using the Integral Histogram. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06). IEEE, pp. 798–805. → pages 23, 42, 63

[2] Agarwal, A., Triggs, B., 2004. Learning to track 3D human motion from silhouettes. In: Proceedings of the twenty-first international conference on Machine learning. ACM, p. 2. → pages 1

[3] Amini, A., Owen, R., Anandan, P., Duncan, J., 1991. Non-rigid motion models for tracking the left-ventricular wall. In: Information Processing in Medical Imaging. Springer, pp. 343–357. → pages 2

[4] Artner, N. N. M., Ion, A., Kropatsch, W. G., Apr. 2010. Multi-scale 2D tracking of articulated objects using hierarchical spring systems. Pattern Recognition 44 (4), 800–810. → pages ix, 9

[5] Arulampalam, M., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Signal Processing, IEEE Transactions on 50 (2), 174–188. → pages 19

[6] Avidan, S., Aug. 2004. Support vector tracking. IEEE transactions on pattern analysis and machine intelligence 26 (8), 1064–72. → pages 1

[7] Babenko, B., Member, S., Yang, M.-h., Member, S., 2011. Robust Object Tracking with Online Multiple Instance Learning. Analysis 33 (8), 1619–1632. → pages 10

[8] Baker, S., Matthews, I., Feb. 2004. Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision 56 (3), 221–255. → pages 55

[9] Bar-Shalom, Y., 1990. Tracking and Data Association. The Journal of the Acoustical Society of America 87 (2), 918. → pages 16, 18, 20

[10] Bay, H., Tuytelaars, T., Gool, L. V., 2006. Surf: Speeded up robust features. Computer VisionECCV 2006, 404–417. → pages 12, 94

[11] Bergen, J. R., Anandan, P., Hanna, J., Hingorani, R., 1992. Hierarchical Model-Based Motion Estimation. In: In Proceedings of the European Conference on Computer Vision. pp. 237–252. → pages 25, 54

[12] Black, M., Jepson, A., 1998. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision 26 (1), 63–84. → pages 15, 25, 54, 76

[13] Bradski, G., 1998. Real time face and object tracking as a component of a perceptual user interface. In: Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on. IEEE, pp. 214–219. → pages 1

[14] Buenaposada, J. M., Muñoz, E., Baumela, L., Apr. 2009. Efficient illumination independent appearance-based face tracking. Image and Vision Computing 27 (5), 560–578. → pages 58

[15] Chen, Q., Sun, Q., Heng, P., Xia, D., Jan. 2008. Parametric active contours for object tracking based on matching degree image of object contour points. Pattern Recognition Letters 29 (2), 126–141. → pages 8

[16] Chen, Y., Rui, Y., Huang, T., 2001. JPDAF based HMM or real-time contour tracking. Computing. → pages 22

[17] Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies 6 (4), 271–288. → pages 1

[18] Collins, R., Lipton, A., Fujiyoshi, H., Kanade, T., 2001. Algorithms for cooperative multisensor surveillance. Proceedings of the IEEE 89 (10), 1456–1477. → pages 1

[19] Collins, R. T., Liu, Y., Leordeanu, M., Oct. 2005. Online selection of discriminative tracking features. IEEE transactions on pattern analysis and machine intelligence 27 (10), 1631–43. → pages 77

[20] Comaniciu, D., Ramesh, V., Meer, P., 2000. Real-time tracking of non-rigid objects using mean shift. In: IEEE International Conference on Computer Vision and Pattern Recognition. Published by the IEEE Computer Society, p. 2142. → pages 22

[21] Comaniciu, D., Ramesh, V., Meer, P., May 2003. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5), 564–577. → pages 3, 7, 10, 22, 42, 63, 76

[22] Cootes, T., Edwards, G., Taylor, C., Jun. 2001. Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6), 681–685. → pages 25, 54

[23] Dame, A., 2011. Video mosaicing using a mutual information-based motion estimation process. Image Processing (ICIP), 2011 18th 2 (x), 1493–1496. → pages 15, 76

[24] Decarlo, D., Metaxas, D., 2000. Optical flow constraints on deformable models with applications to face tracking. International Journal of Computer Vision 38 (2), 99–127. → pages 1, 15

[25] Doucet, A., Godsill, S., Andrieu, C., 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and computing 10 (3), 197–208. → pages 19

[26] Duraiswami, R., Davis, L., 2005. Efficient Mean-Shift Tracking via a New Similarity Measure. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 176–183. → pages 76

[27] Erdem, c. E., Nov. 2007. Video object segmentation and tracking using region-based statistics. Signal Processing: Image Communication 22 (10), 891–905. → pages ix, 13

[28] Fablet, R., Black, M., 2002. Automatic detection and tracking of human motion with a view-based representation. In: Computer VisionECCV 2002. Springer, pp. 476–491. → pages 1

[29] Firouzi, H., Najjaran, H., Dec. 2010. Adaptive Non-rigid Object Tracking by Fusing Visual and Motional Descriptors. In: 2010 International Conference on Digital Image Computing: Techniques and Applications. IEEE, pp. 58–62. → pages 3

[30] Firouzi, H., Najjaran, H., Jul. 2010. Real-time monocular vision-based object tracking with object distance and motion estimation. In: 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Vol. 1. IEEE, pp. 987–992. → pages 3

[31] Firouzi, H., Najjaran, H., Dec. 2012. Robust decentralized multi-model adaptive template tracking. Pattern Recognition 45 (12), 4494–4509. → pages 3, 4, 63

[32] Firouzi, H., Najjaran, H., 2012. Robust gaussian-based template tracking. Intelligent Robotics and Applications. → pages 3

[33] Gai, J., Stevenson, R. L., Jan. 2011. Studentized dynamical system for robust object tracking. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society 20 (1), 186–99. → pages 24

[34] Georgescu, B., Comaniciu, D., Han, T. X. T., Zhou, X. X. S., 2004. Multi-model component-based tracking using robust information fusion. Statistical Methods in Video Processing, 61–70. → pages 25, 26

[35] Gordon, N., Salmond, D., Smith, A., 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Radar and Signal Processing, IEE Proceedings F 140 (2), 107–113. → pages 18

[36] Greiffenhagen, M., Comaniciu, D., Niemann, H., Ramesh, V., 2001. Design, analysis, and engineering of video monitoring systems: an approach and a case study. Proceedings of the IEEE 89 (10), 1498–1517. → pages 1

[37] Grimson, W., 1991. Object recognition by computer: the role of geometric constraints. MIT Press. → pages 2

[38] Hager, G., Belhumeur, P., 1998. Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10), 1025–1039. → pages 25, 54, 55, 76

[39] Hampapur, A., Bobbitt, R., Brown, L., Desimone, M., Feris, R., Kjeldsen, R., Lu, M., Mercier, C., Milite, C., Russo, S., Others, 2009. Video analytics in urban environments. 2009 Advanced Video and Signal Based Surveillance, 128–133. → pages 1

[40] Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., von Seelen, W., 1998. Computer vision for driver assistance systems. In: Proc. SPIE. Vol. 3364. Citeseer, pp. 136–147. → pages 1

[41] Hu, W., Tan, T., Wang, L., Maybank, S., Aug. 2004. A Survey on Visual Surveillance of Object Motion and Behaviors. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 34 (3), 334–352. → pages 1

[42] Isard, M., Blake, A., 1998. Condensation - conditional density propagation for visual tracking. International journal of computer vision 29 (1), 5–28. → pages 81, 83

[43] Jepson, A., Fleet, D., El-Maraghi, T., Oct. 2003. Robust online appearance models for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10), 1296–1311. → pages 7, 37, 78

[44] Jiang, N., Liu, W., Wu, Y., Aug. 2011. Learning adaptive metric for robust visual tracking. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society 20 (8), 2288–300. → pages 77

[45] Joa, M., Vasconcelos, M. J. a. M., Ventura, S. M. R., Freitas, D. R. S., Tavares, J. a. M. R. S., Nov. 2011. Towards the automatic study of the vocal tract from magnetic resonance images. Journal of voice : official journal of the Voice Foundation 25 (6), 732–42. → pages 2

[46] Julier, S., Uhlmann, J., 1997. A new extension of the Kalman filter to nonlinear systems. In: Int. Symp. Aerospace/Defense Sensing, Simul. and Controls. Vol. 3. Citeseer, p. 26. → pages 18

[47] Kalal, Z., Mikolajczyk, K., Matas, J., Dec. 2011. Tracking-Learning-Detection. IEEE transactions on pattern analysis and machine intelligence 34 (7), 1409–1422. → pages 63

[48] Kettnaker, V., Zabih, R., 1999. Bayesian multi-camera surveillance. In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). Vol. 2. IEEE Comput. Soc, pp. 253–259. → pages 1

[49] Khan, Z., Gu, I., Backhouse, A., 2011. Robust Visual Object Tracking using Multi-Mode Anisotropic Mean Shift and Particle Filters. IEEE

trans. Circuits and Systems for Video Technology 21 (1), 74–87. → pages 10, 23

[50] Kitagawa, G., 1987. Non-Gaussian state-space modeling of nonstationary time series. Journal of the American Statistical Association 82 (400), 1032–1041. → pages 18

[51] Kr, B., Zivkovic, Z., Krose, B., 2004. An EM-like algorithm for color-histogram-based object tracking. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. 1, 798–803. → pages 23

[52] Krahnstoever, N., Tu, P., Yu, T., Patwardhan, K., Hamilton, D., Yu, B., Greco, C., Doretto, G., 2009. Intelligent video for protecting crowded sports venues. 2009 Advanced Video and Signal Based Surveillance, 116–121. → pages 1

[53] Kumar, R., Sawhney, H., Samarasekera, S., Hsu, S., Tao, H., Guo, Y., Hanna, K., Pope, A., Wildes, R., Hirvonen, D., Others, 2001. Aerial video surveillance and exploitation. Proceedings of the IEEE 89 (10), 1518–1539. → pages 1, 15, 58

[54] Leone, a., Distante, C., Apr. 2007. Shadow detection for moving objects based on texture analysis. Pattern Recognition 40 (4), 1222–1233. → pages 22

[55] Lim, J., Ross, D., Lin, R., 2004. Incremental learning for visual tracking. Advances in neural information processing systems 17 (1), 793–800. → pages 29

[56] Lowe, D., 1980. Solving for the Parameters of Object Models from Image Descriptions. In: Image Understanding Workshop. pp. 121–127. → pages 2

[57] Lowe, D., 1987. Three-dimensional object recognition from single two-dimensional images. Artificial intelligence. → pages 2

[58] Lowe, D. D. G., Nov. 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110. → pages ix, 6, 10, 12, 94

[59] Lucas, B., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: International joint conference

on artificial intelligence. Vol. 3. pp. 674–679. → pages 15, 20, 25, 54, 75, 81

[60] Mathes, T., Piater, J., 2006. Robust non-rigid object tracking using point distribution manifolds. Pattern Recognition. → pages 7

[61] Matthews, I., Baker, S., Nov. 2004. Active Appearance Models Revisited. International Journal of Computer Vision 60 (2), 135–164. → pages 25

[62] Matthews, I., Ishikawa, T., Baker, S., Jun. 2004. The template update problem. IEEE transactions on pattern analysis and machine intelligence 26 (6), 810–5. → pages 11, 26, 27, 55, 81, 82

[63] Meer, P., Dec. 2004. Robust techniques for computer vision. Emerging topics in computer vision 17 (6), 555–80. → pages 75, 77, 78

[64] Messelodi, S., Modena, C., Segata, N., Zanin, M., 2005. A kalman filter based background updating algorithm robust to sharp illumination changes. Image Analysis and ProcessingICIAP 2005, 163–170. → pages 22

[65] Moon, T. T., Dec. 1996. The expectation-maximization algorithm. Signal Processing Magazine, IEEE 13 (6), 47–60. → pages 21

[66] Morel, J.-M., Yu, G., Jan. 2009. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. SIAM Journal on Imaging Sciences 2 (2), 438–469. → pages 94

[67] Ning, J., Zhang, L., Zhang, D., Wu, C., 2009. Robust Object Tracking Using Joint Color-Texture Histogram. International Journal of Pattern Recognition and Artificial Intelligence 23 (07), 1245. → pages 23, 42

[68] Ning, J., Zhang, L., Zhang, D., Wu, C., 2011. Scale and Orientation Adaptive Mean Shift Tracking. IET Computer Vision. → pages 23, 42

[69] Olson, C., 2001. Image registration by aligning entropies. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. Vol. 2. IEEE Comput. Soc, pp. II–331–II–336. → pages 5

[70] Papadakis, N., Bugeau, A., Jan. 2011. Tracking with occlusions via graph cuts. IEEE transactions on pattern analysis and machine intelligence 33 (1), 144–157. → pages 8

[71] Papanikolopoulos, N. N., Khosla, P. P., 1993. Adaptive robotic visual tracking: Theory and experiments. IEEE Transactions On Automatic Control 38 (3), 429–445. → pages 2

[72] Ponsa, D., López, A. M., Nov. 2009. Variance reduction techniques in particle-based visual contour tracking. Pattern Recognition 42 (11), 2372–2391. → pages 22

[73] Rabiner, L., Juang, B., Jun. 1986. An introduction to hidden Markov models. IEEE ASSP Magazine 3 (1), 4–16. → pages 18

[74] Ramesh, V., Comaniciu, D., Genc, Y., Paragios, N., Zhu, Y., Mittal, A., Zoghlami, I., Gao, X., Tsin, Y., 2005. Real-time vision at siemens corporate research. In: Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, pp. 300–305. → pages 1

[75] Rand, D., Kizony, R., Weiss, P., 2008. The Sony PlayStation II EyeToy: low-cost virtual reality for use in rehabilitation. Journal of Neurologic Physical Therapy 32 (4), 155. → pages 1

[76] Rasmussen, C., Hager, G., 1998. Joint probabilistic techniques for tracking multi-part objects. In: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on. IEEE, pp. 16–21. → pages 25

[77] Redner, R. R., Walker, H. H., 1984. Mixture densities, maximum likelihood and the EM algorithm. SIAM review 26 (2), 195–239. → pages 31, 60

[78] Richa, R., Sznitman, R., Taylor, R., Hager, G., Sep. 2011. Visual tracking using the sum of conditional variance. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. No. 2. IEEE, pp. 2953–2958. → pages 15, 76

[79] Rius, I., Gonzàlez, J., Varona, J., Xavier Roca, F., Nov. 2009. Action-specific motion prior for efficient Bayesian 3D human body tracking. Pattern Recognition 42 (11), 2907–2921. → pages 1

[80] Roberts, L., 1965. Machine perception of three-dimensional solids. MIT Press. → pages 2

[81] Roche, A., Malandain, G., Ayache, N., 2000. Unifying maximum likelihood approaches in medical image registration. International Journal of Imaging Systems and Technology 11 (1), 71–80. → pages 5

[82] Ross, D. a., Lim, J., Lin, R.-S., Yang, M.-H., Aug. 2007. Incremental Learning for Robust Visual Tracking. International Journal of Computer Vision 77 (1-3), 125–141. → pages ix, 7, 8, 11, 24, 63, 76

[83] Ross, D. a., Tarlow, D., Zemel, R. S., Mar. 2010. Learning Articulated Structure and Motion. International Journal of Computer Vision 88 (2), 214–237. → pages ix, 9

[84] Schreiber, D., Sep. 2007. Robust template tracking with drift correction. Pattern Recognition Letters 28 (12), 1483–1491. → pages 55

[85] Shafique, K., Shah, M., Jan. 2005. A noniterative greedy algorithm for multiframe point correspondence. IEEE transactions on pattern analysis and machine intelligence 27 (1), 51–65. → pages 7

[86] Shan, C., Tan, T., Wei, Y., Jul. 2007. Real-time hand tracking using a mean shift embedded particle filter. Pattern Recognition 40 (7), 1958–1970. → pages 23

[87] Silveira, G., Malis, E., 2010. Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. International journal of computer vision 89 (1), 84–105. → pages 26, 55

[88] Stauffer, C., Grimson, W., 1999. Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. Vol. 2. IEEE, pp. 246–252. → pages 22

[89] Stauffer, C., Grimson, W., 2000. Learning patterns of activity using real-time tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (8), 747–757. → pages 1

[90] Sun, W., Yang, X., Feb. 2011. Nonrigid image registration based on control point matching and shifting. Optical Engineering 50 (2), 027006. → pages 5

[91] Torresani, L., Yang, D., Alexander, E., Bregler, C., 2001. Tracking and modeling non-rigid objects with rank constraints. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. IEEE Comput. Soc, pp. I–493–I–500. → pages 2

[92] Vasconcelos, M., Tavares, J., 2008. Methods to automatically build Point Distribution Models for objects like hand palms and faces represented in images. Computer Modeling in Engineering & Sciences 36 (3), 213–241. → pages 25

[93] Vasconcelos, M. J. M., Ventura, S. M. R., Freitas, D. R. S., Tavares, J. M. R. S., Oct. 2010. Using statistical deformable models to reconstruct vocal tract shape from magnetic resonance images. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine 224 (10), 1153–1163. → pages 2

[94] Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. IEEE Comput. Soc, pp. I–511–I–518. → pages 5, 12

[95] Wang, F., Vemuri, B. C., Jan. 2007. Non-Rigid Multi-Modal Image Registration Using Cross-Cumulative Residual Entropy. International Journal of Computer Vision 74 (2), 201–215. → pages 15, 76

[96] Wang, Q., Chen, F., Xu, W., Feb. 2011. Adaptive multi-cue tracking by online appearance learning. Neurocomputing 74 (6), 1035–1045. → pages ix, 10, 12, 24

[97] Wang, Q., Chen, F., Xu, W., Apr. 2011. Tracking by third-order tensor representation. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society 41 (2), 385–96. → pages ix, 13

[98] Yang, M., Kriegman, D., Ahuja, N., 2002. Detecting faces in images: A survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (1), 34–58. → pages 1

[99] Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: A survey. ACM Computing Surveys 38 (4), 13. → pages 1, 74

[100] Zhao, G., Pietikainen, M., 2006. Local binary pattern descriptors for dynamic texture recognition. Pattern Recognition, 2006. ICPR 2006. . . . , 18–21. → pages 11

[101] Zhou, H., Yuan, Y., Zhang, Y., Shi, C., Jan. 2009. Non-rigid object tracking in complex scenes. Pattern Recognition Letters 30 (2), 98–102. → pages 23