DYNAMIC MOTION AND APPEARANCE MODELING
FOR ROBUST VISUAL TRACKING

A Thesis in
Electrical Engineering
by
Hwasup Lim

© 2007 Hwasup Lim

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2007
The thesis of Hwasup Lim was reviewed and approved* by the following:

Octavia I. Camps  
Adjunct Professor of Electrical Engineering  
Thesis Adviser  
Co-Chair of Committee

Constantino M. Lagoa  
Associate Professor of Electrical Engineering  
Co-Chair of Committee

Mario Sznaier  
Adjunct Professor of Electrical Engineering

William E. Higgins  
Professor of Electrical Engineering

Sunil K. Sinha  
Assistant Professor of Civil and Environmental Engineering

W. Kenneth Jenkins  
Professor of Electrical Engineering  
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School.
Abstract

Visual tracking is one of the most active areas in computer vision and it has many promising applications such as human motion capture, human computer interface, and visual surveillance. The performance of visual tracking systems is often severely affected by appearance changes of the target, occlusion, scene clutter and sensor noise. Conventional approaches address these challenges by introducing simple dynamic models such as constant velocity of the target, and adaptive appearance models based on the assumption that the target features are continuous in the spatial and time domains. These approaches, however, have been shown in the literature to be fragile to occlusion and susceptible to incorrect observations. In addition, modeling dynamic target appearance is often faced with difficulties due to its high dimensionality and nonlinearity.

This thesis presents efficient frameworks for modeling dynamic motion and appearance of the target object by employing tools from robust system identification theory and nonlinear dimensionality reduction techniques. The target motion and appearance are modeled as unknown operators that satisfy certain interpolation conditions. The unknown operators can then be identified by solving a convex optimization problem where high dimensionality and nonlinearity in appearance changes are addressed by efficiently projecting high dimensionality descriptors into low dimensional manifolds. The learned dynamic models are then used to accurately predict the location and appearance of the targets in future frames, thus preventing tracking failures due to model drifting, target occlusion, and scene clutter.

The advantages of this approach are multiple: 1) It allows to treat the tracking problem from an input/output point of view, requiring very little a priori knowledge about the target. At the same time, it allows to naturally incorporate as much prior knowledge as it is available. 2) It decouples nonlinear appearance changes from linear dynamics facilitating the identification process. 3) It provides mechanisms to invalidate a priori assumptions and worst-case estimates of the identification error that can be used to determine how long the predictions are valid. 4) Finally, the use of reliable models allows for simpler computational complexity algorithms.

The proposed approach was validated with several experiments conducted under various scenarios such as single and multiple target tracking, and static and moving camera environments, and finally, single and multiple camera environments. The advantages of the proposed approach are shown with illustrated examples in which robustness to severe occlusion and efficiency of appearance modeling are compared to conventional approaches.
Table of Contents

List of Tables ........................................................................ vii
List of Figures ........................................................................ viii
Acknowledgments ................................................................... xii

Chapter 1. Introduction ........................................................... 1
  1.1 Visual Tracking ............................................................... 1
  1.2 Dynamic Elements in Visual Tracking ............................ 3
    1.2.1 Motion Change ....................................................... 3
    1.2.2 Color Change ....................................................... 3
    1.2.3 Appearance Change .............................................. 4
  1.3 Dynamic Modeling for Robust Visual Tracking .............. 4
    1.3.1 Problem Statement ............................................... 4
    1.3.2 Proposed Approach .............................................. 5
    1.3.3 Assumptions ....................................................... 5
  1.4 Contributions ............................................................... 6
  1.5 Thesis Organization ..................................................... 6

Chapter 2. Visual Tracking Techniques ................................. 7
  2.1 Introduction ............................................................... 7
  2.2 Visual Tracking Techniques ........................................ 7
    2.2.1 Template Matching .............................................. 7
    2.2.2 Mean Shift Algorithm .......................................... 9
  2.3 Visual Tracking with Dynamics .................................... 10
    2.3.1 Dynamics Model ................................................. 11
    2.3.2 Kalman Filter .................................................... 12
    2.3.3 Particle Filter .................................................... 13
  2.4 Discussion ............................................................... 15

Chapter 3. Robust System Identification for Dynamics Identification ........................................ 16
  3.1 Introduction ............................................................... 16
  3.2 Preliminary ............................................................... 17
  3.3 Background on Robust System Identification ................. 18
    3.3.1 Information Consistency and Interpolatory Algorithms 19
3.3.2 Caratheodory-Fejer Based Interpolatory Identification Algorithms ........................................... 19
3.4 Multiframe Tracking as an Interpolation Problem ................................................................. 21
  3.4.1 Model (in)validation ............................................. 23
  3.4.2 Worst-case estimates of the prediction error ................................................................. 24
3.5 Discussion .................................................................................................................. 25
  3.5.1 Applications to Robust Visual Tracking ................................................................. 25
  3.5.2 Selection of the Input and Output of the System .................................................. 26
  3.5.3 Selection of the Parametric Component and the Nonparametric Component ................. 27
  3.5.4 Accuracy and Computation Time ........................................................................ 28

Chapter 4. Dynamic Motion Modeling .................................................................................. 31
  4.1 Introduction ............................................................................................................. 31
  4.2 The Proposed Approach .......................................................................................... 32
  4.3 Illustrative Examples ............................................................................................... 34
    4.3.1 Experimental Setups .................................................................................. 34
    4.3.2 Single Camera Tracking ........................................................................... 37
    4.3.3 Multiple Camera Tracking ....................................................................... 43
  4.4 Discussion ............................................................................................................. 45

Chapter 5. Dynamic Homography Modeling ........................................................................ 48
  5.1 Introduction ............................................................................................................. 48
  5.2 Homography Estimation .......................................................................................... 49
  5.3 The Proposed Approach .......................................................................................... 52
  5.4 Illustrative Examples ............................................................................................... 54
    5.4.1 Experimental Result .................................................................................. 54
  5.5 Discussions ............................................................................................................ 58

Chapter 6. Dynamic Color Modeling .................................................................................. 60
  6.1 Introduction ............................................................................................................. 60
  6.2 The Proposed Approach .......................................................................................... 60
  6.3 Illustrative Examples ............................................................................................... 62
    6.3.1 Experimental Setups .................................................................................. 62
    6.3.2 Experimental Results ................................................................................. 63
  6.4 Discussion ............................................................................................................. 68
Chapter 7. Dynamic Appearance Modeling .................................. 72
  7.1 Introduction ................................................................. 72
  7.2 The Proposed Approach .................................................. 75
  7.3 Dynamic Appearance Modeling ........................................... 75
      7.3.1 Nonlinear Dimensionality Reduction for Dynamic Appearance
            Modeling ............................................................. 76
      7.3.1.1 Locally Linear Embedding (LLE) for Dimension Reduction ....... 76
      7.3.2 Robust Dynamics Identification on the LLE manifold ............... 81
      7.3.3 Recovering Dynamic Appearance from the Manifold Space .......... 82
            7.3.3.1 Radial Basis Function ..................................... 82
  7.4 Dynamic Motion Modeling .................................................. 85
  7.5 Human Tracking using Dynamic Appearance and Motion Modeling .... 85
      7.5.1 Occlusion Handling ............................................... 85
  7.6 Illustrative Examples ....................................................... 86
      7.6.1 Implementation Details ........................................... 86
      7.6.2 Tracking Results .................................................. 88
  7.7 Discussion ................................................................. 90

Chapter 8. Conclusions .......................................................... 91
  8.1 Related Publications ....................................................... 91

Appendix. Notation ............................................................. 92

References ................................................................. 93
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td><em>A priori</em> information for the jumping person sequence shown in Figure 4.2</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td><em>A priori</em> information for the paper airplane sequence shown in Figure 4.3</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td><em>A priori</em> information for the bouncing ball sequence shown in Figure 4.4</td>
<td>42</td>
</tr>
<tr>
<td>4.4</td>
<td><em>A priori</em> information for the car sequence shown in Figure 4.5</td>
<td>43</td>
</tr>
<tr>
<td>5.1</td>
<td><em>A priori</em> information for the moving camera sequence shown in Figure 5.4</td>
<td>56</td>
</tr>
<tr>
<td>6.1</td>
<td><em>A priori</em> information for the rolling ball sequence shown in Figure 6.2</td>
<td>65</td>
</tr>
<tr>
<td>6.2</td>
<td><em>A priori</em> information for the flying ball sequence shown in Figure 6.7</td>
<td>68</td>
</tr>
<tr>
<td>7.1</td>
<td><em>A priori</em> information for the walking sequences shown in Figures 7.12, 7.14, and 7.15.</td>
<td>87</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 The proposed approach of dynamic modeling for robust visual tracking.  

2.1 Visual tracking using template matching. The first image of the top left hand corner is a template and the second image is the warped target taken from the image. The affine parameters are calculated minimizing the error between the template and the warped target.  

2.2 The procedure of the mean shift algorithm. The location of the target region is iteratively calculated by moving the search window to the local mean.  

3.1 Basic transfer functions for the parametric part $F_p$. (a) step function $u(t) \leftrightarrow \frac{z}{z-1}$, (b) ramp function $t \leftrightarrow \frac{z}{(z-1)^2}$, (c) parabolic function $t^2 \leftrightarrow \frac{z(z+1)}{(z-1)^3}$, (d) exponential function $e^t \leftrightarrow \frac{z}{z-a}$, (e) sine function $\sin(t) \leftrightarrow \frac{z \sin(\omega)}{z^2 - 2z \cos(\omega) + 1}$, and (f) cosine function $\cos(t) \leftrightarrow \frac{z(\cos(\omega))}{z^2 - 2z \cos(\omega) + 1}$.  

4.1 Past observations of the target locations are used for dynamic motion modeling using the CF-based identification. The prediction of future locations is produced based on the identified dynamics.  

4.2 Comparison between constant velocity dynamics and CF-identified dynamics of the jumping person sequence where the person becomes occluded while jumping. The green line in the images of (b) represents the predicted target positions by CF-identified dynamics, which accurately predict the target positions even during occlusion.  

4.3 Comparison between constant velocity dynamics and CF identified dynamics of the paper airplane sequence where the paper airplane becomes occluded, changing its direction. The green line in the images of (b) represents the predicted target positions by CF-identified dynamics, which accurately predicts the target positions even during occlusion.  

4.4 Comparison between constant velocity dynamics and CF-identified dynamics of the bouncing ball sequence where the ball becomes occluded, changing its direction. The green line in the images of (b) represents the predicted target positions by CF-identified dynamics, which accurately predicts the target positions even during occlusion.
4.5 Comparison between constant velocity dynamics and CF-identified dynamics of the car sequence where the car becomes occluded, changing its size and direction. ......................................................... 44

4.6 Triangulation method with CF approach to occlusion handling. CF-identified dynamics accurately predicts the position of the target in the right camera view when the target is occlusion by the wall. ....... 46

5.1 Homography method for multiple cameras. ................................. 49

5.2 Comparison between the Kalman filter with constant velocity model and homography matrix for occlusion handling. The first rows of (a) and (b) were taken by Camera 1 and the second rows were taken by Camera 2. Even though the target object is occluded, the position can be estimated from the tracked position in the other camera view using the homography matrix. .............................................................. 53

5.3 The images in the first row come from the static camera for reference and the images in the second row come from the moving camera. ....... 55

5.4 Dynamic homography modeling using the CF-identified dynamics. The blue lines are the ground truth measurements for 15 frames and the red lines are the estimations by the CF approach for 20 frames where the last 5 frames are predicted by the identified dynamics of each parameter. 57

5.5 The images in the first row come from the static camera for reference and the images in the second row come from the moving camera. The blue rectangles are the ground truth measurements of the corresponding points of Figure (a) and the red crosses are the estimated positions by the estimated from the dynamic homography matrix by the CF approach. 58

6.1 Past observations of the target locations and color are used for dynamic motion and color modeling using CF-based identification. The prediction of future locations and color histogram are produced based on the identified dynamics. .............................................................. 61

6.2 (a) Frame 1 (first row) and Frame 10 (second row) of a multicolored ball. (b) to (d) probability of a pixel belonging to the ball, according to the histogram from the (b) first frame (constant color model), (c) previous frame (adaptive color model), and (d) predicted by the CF-based dynamic color model. .............................................................. 65

6.3 Measurements and CF estimated values of the (a) orange, (b) blue, and (c) red bin counts of the hue histogram of the ball show in 6.2 (a) as it rolls in front of the camera. .............................................................. 66
6.4 Tracking results of constant and adaptive color modeling. All the above methods fail to track a red and blue ball due to color change or occlusion.

6.5 Measurements and CF estimated values of the (a) blue and (b) red bin counts of the hue histogram of the ball shown in Figure 6.7 (a) and (b) as it rolls in front of the camera.

6.6 Measurements and CF estimated values of the (a) x and (b) y positions of the ball shown in Figure 6.7 (b) as it rolls in front of the camera.

6.7 CF-based dynamic color model. Tracking using dynamic color model alone (top), and both dynamic color model and dynamic motion model (bottom) of a red and blue ball is successful, even in the presence of occlusion. Conjunction with dynamic motion model provides accurate position predictions during occlusion.

7.1 Generated walking sequence after the learning step, $t = 38, 39, 40, \ldots, 48$.

7.2 Dynamic appearance and occlusion by a background object. The white box is the predicted appearance and the black box is the probability map based on the predicted appearance.

7.3 A generative model for learning and tracking appearance. $z_t, y_t, x_t$ represents the appearances in the image, the manifold, and the state of the manifold respectively.

7.4 Steps of Locally Linear Embedding [58]. (1) Find the neighbors of each data point, $X_i$. (2) Compute the weights $W_{ij}$ that best reconstruct $X_i$ from its neighbor. (3) Compute the vectors $Y_i$ best reconstructed by $W_{ij}$.

7.5 Example of Locally Linear Embedding algorithm [58]. A. three dimensional data. B. sampled data from a two dimensional manifold of A. C. two dimensional representation by LLE.

7.6 Dynamic appearances of a walking motion for 37 frames, $t = 0, 6, 12, \ldots, 36$.

7.7 Low dimensional representation of the walking sequence by LLE.

7.8 Three dimensional representation of the walking sequence by LLE. Periodicity of walking cycle is observed in the manifold space.

7.9 Dynamics Identification by CF interpolation and prediction for next 38 frames.

7.10 Center reduction in RBF interpolation.

7.11 Generated walking sequence for 76 frames by RBF, $t = 0, 3, 6, \ldots, 75$.

7.12 Handling occlusion caused by the background object, $t = 60, 75, 90, \ldots, 105$. The white box in the second row is the predicted appearance and the black box is the probability map based on the predicted appearance.
7.13 Layer selection by the likelihood to the observed appearance in the overlapped region by two templates. The higher likelihood represents the front layer and vice versa. ................................................................. 88

7.14 Handling occlusion caused by other moving people. The layer is automatically selected based on the likelihood of each template to the observed appearance with the overlapped regions. ................................. 89

7.15 Moving camera after learning appearance. The tracker robustly tracks the target person even though the camera moves. ......................... 89
Acknowledgments

Most of all, I thank God for letting me finish my dissertation. I am most grateful and indebted to my thesis advisor, Dr. Octavia I. Camps, for the large doses of guidance, patience, and encouragement that she has shown me during my time here at the Pennsylvania State University. I am especially thankful for her financial support that has been provided to me over the years. I thank my other committee members, Dr. Mario Sznaier, Dr. William E. Higgins, Dr. Constantino Lagoa, and Dr. Sunil Sinha for their insightful commentary on my work. I am especially thankful for Dr. Sznaier’s valuable advice and suggestions that were significant in their guidance for doing my research work. I am also grateful and indebted to Ben Fransen and Roberto Lubliner man for inspiration and enlightening discussion on a wide variety of topics.

Lastly, I would like to thank my family including my wife Sukyoung Choi for her unlimited love, patience, encouragement, and support throughout my graduate program, my daughter Yumin who was born last year and who adds so much joy to my life, my mother, my sister, and my brother for their continued love and prayer, and my parents in law for their encouragement and support even from a distance.
Chapter 1

Introduction

1.1 Visual Tracking

One of the main activities of humans is to track objects from the images obtained with the human vision system. This instinctive process of tracking objects can be divided into three steps: target object separation from the background, continuous localization of the target object, and finally analysis and recognition from its motions. Since the beginning of computer vision, researchers have been pursuing the goal that computers imitate this process, using digital images obtained from digital imaging devices.

Visual tracking is one of the most active computer vision areas and embodies such a goal using mathematical tools. Since the notion of visual tracking is similar to the actual human vision system, the techniques for visual tracking have very similar procedures as found in humans. In the literature, visual tracking sometimes concentrates only on segmenting (detecting) target objects. Generally speaking, visual tracking can be defined as “the repeated process of localizing the target object from a sequence of images in digital videos,” including other accompanied processes to achieve this goal.

Recently, visual tracking has attracted increasing attention due to the ease of accessibility to inexpensive and accurate digital imaging equipment. It has a potential and promising wide range of applications. The following examples show several application areas using visual tracking techniques.

1. **Motion Capture**: Once human motions are captured using a single or multiple cameras, the motion records of a moving person can be used for rendering 3D motion in computer graphics. Manipulating the obtained motion records, one can make thousands of virtual characters that have slightly different motions. Another example in the same context is that the captured motions in sports videos can also be used for performance analysis of athletes such as a pose analysis of swinging a golf club or batting a baseball.

2. **Human Computer Interaction**: Human gestures often express more directly and efficiently a user’s demands beyond pressing buttons or typing on computer
keyboards. The captured motion of human gestures using video cameras can be analyzed and then interpreted as signals or commands to machines. This interaction between a human and a computer using gestures provides more freedom of action to the user while transferring their commands. In particular, this capability can be very useful for gaming, allowing the player to make real actions for the actions in the game.

3. **Surveillance**: Surveillance is one of the most promising and practical applications of visual tracking. For the purpose of security, many CCTV or digital video cameras are installed in streets, buildings, or airports to monitor traffic states or human activities. Knowing what the objects are doing is of great interest in monitoring security videos. However, monitoring all videos to detect objects doing unwanted actions is very time consuming work. Visual tracking techniques can efficiently provide pre-screening of normal actions and can then detect suspicious actions from the videos, greatly reducing human efforts.

Several different approaches to perform visual tracking have been developed, starting from the early tracking algorithms that concentrated on tracking feature points [45], image templates [24], contours [35], and object regions [16]. Various modifications or extensions of such approaches have been built upon their original work for more accuracy and robustness, and new algorithms have been continuously attempted to increase the tracking performance.

The variety of visual tracking techniques comes from methods for target representation, coherence estimation between frames, multiple cameras, etc. Regardless of the types of methods, the main interest of visual tracking is how well the target is continuously localized in a sequence of images. Note that due to the continuous nature of motion, the visual information of moving objects is also continuous over time\(^1\) and highly correlated with adjacent frames. This fact allows for the use of a prediction method using several filtering tools under assumptions of noisy measurements. Many parametric [34] and nonparametric [29] approaches have been developed, mitigating strict assumptions such as linear or Gaussian model. While successful in many scenarios, these approaches still fall short of describing dynamic variation due to target motion.

Refer [7] for further research in this topic. A roadmap of visual tracking and some of the above approaches will be briefly introduced in the next chapter.

---

\(^1\)Strictly speaking, it is not continuous because the image is sampled by 25~30 frames per second in digital video cameras.
1.2 Dynamic Elements in Visual Tracking

The main challenge in visual tracking is the inevitable variation of motion, color, or appearance of the target object shown in the image due to internal or external factors. Internal factors originate from the target motion, structure, or self-occlusion, and external factors originate from camera motion, illumination change, or occlusion. Successful modeling of such variations often directly affects the tracking performance. While there are many studies focusing on specific variations, the following variations are considered as research topics in this thesis.

1.2.1 Motion Change

What makes the tracking problem difficult is the fact that the motions of the target objects are, in general, not identical in every situation unless those are repeated motions by certain mechanical connections. In addition, the transition process of the target object and the measurement process by image-capturing devices are also affected by unpredictable noise. In this context, the past observations are the only useful cues for predicting the future location of the target object. Various filtering techniques, for example, the Kalman filter [23], the particle filter (Condensation) [29], or their extensions, are utilized to produce better predictions. Most of these approaches, however, assume that targets move with simple dynamics such as constant velocity. This assumption is often fatal in the presence of clutter or occlusion because it cannot produce an accurate prediction due to a lack of capturing realistic motion dynamics. Therefore, accurate motion modeling is essential for robust visual tracking.

1.2.2 Color Change

Color information, for example, a color histogram, is important for identifying the target region, which is often converted to a probability distribution using the color histogram to apply tracking algorithms. The color histogram of the target object can also dynamically vary over time due to viewpoint changes when the object is multicolored. To overcome such variations, an adaptive approach that incorporates past color observations into the current color histogram was introduced in [73]. This approach, however, still remains susceptible to clutter or occlusion, incorporating incorrect color information into the current color histogram. Such incorrect updates will lead the tracker to fail to identify the target due to false color information. This problem can be also addressed by a lack of accurate dynamics in color modeling that can prevent the target color model from being corrupted by false information.
1.2.3 Appearance Change

Even though a color histogram of the target object allows the tracker to distinguish the target object from the background, the color histogram is not a feature that can describe the entire visual information of the target region. The most desirable strategy is to model the target appearance segmented from the image, containing all visual information taken by digital video cameras. To track the target using its appearance, constant or adaptive image templates of the target are commonly used for appearance verification. However, appearance variation due to, for example, articulated motion is a challenging problem because such methods allow for only one parametric transformation of all the pixels in the template. Moreover, adaptation of appearance variation does not provide any predictions for future appearance. Adaptive approaches also have incorrect update problems as a result of false information. This problem necessitates the need for dynamic appearance modeling with a predictive framework. However, direct modeling of such appearance is often a difficult problem due to high dimensionality and nonlinearity of appearance variations. An alternative approach needs to address these two challenging problems.

1.3 Dynamic Modeling for Robust Visual Tracking

As discussed in the previous section, visual tracking is unavoidably involved in the dynamic modeling of the target features. Accurate dynamic models directly determine the robustness of the tracker to various changes. Therefore, the aim of this thesis is to provide a framework that can handle temporal variation of the target features with a predictive mechanism.

1.3.1 Problem Statement

The goal of the research presented in this thesis is:

To identify accurately the dynamics of the target motion, color, and appearance of the target object based on the past observations, and to design dynamic motion, color, and appearance models that can make accurate predictions using the identified dynamics, even in the presence of clutter or occlusion.

This thesis focuses on developing dynamic models of target features rather than naive visual tracking techniques. The solution to the above problem will increase robustness of the existing visual tracking techniques. In this thesis, natural indoor or outdoor scenes taken in real world are considered for the experiments instead artificial video scenes.
1.3.2 Proposed Approach

In this thesis, we advocate the use of robust system identification to identify the dynamics and to make accurate predictions of motion, color, and appearance. As shown in Figure 1.1, the procedure consists of three steps: 1) a collection of the past measurements of motion, color, and appearance, 2) dynamics identification, and 3) prediction of motion, color, and appearance using the identified dynamics. The obtained dynamic models will be combined with existing tracking techniques for target identification.

![Diagram of Proposed Approach]

Fig. 1.1. The proposed approach of dynamic modeling for robust visual tracking

Note that a method for target representation should be defined prior to the proposed approach in order to discriminate the target object from the background. Existing tracking algorithms then localize the target using the representation. There are many possible options to represent or segment the target; for example, probability distribution, image template, or contour. Probability distributions using color histograms and image templates as an appearance are used in this thesis.

Unlike modeling motion or color, modeling appearance is often a difficult problem due to high dimensionality and nonlinearity. These problems are addressed in Chapter 7, where efficient solutions are provided.

1.3.3 Assumptions

To specify the objective and to confine the coverage of the research, the following assumptions will be made in the context of this thesis.
• The initial position of the target object at the beginning of tracking is manually initialized by the user. Even though this initialization is an important problem to start tracking, this topic is not our focus and can be supported by existing algorithms in the literature.

• A small set of past observations is used under the assumption that there were no clutter or occlusion during these observations.

• A small set of a priori information on dynamics is given for dynamics identification while still assuming a causal system\(^2\).

1.4 Contributions

Modeling variations of motion, color, or appearance are challenging problems that computer vision researchers have extensively studied. However, dynamics of such variations are often overlooked in the literature. This thesis specifically utilizes these dynamics and has several contributions to visual tracking, which have not been attempted before.

• Dynamic modeling is presented against adaptive modeling, which is a popular method to handle temporal variation of target motion, color, and appearance. However, adaptive modeling does not have a predictive mechanism and remains susceptible to incorrect measurements. In contrast, dynamic modeling provides prediction based on the identified dynamics to prevent the tracker from being set adrift due to scene clutter and occlusion.

• A robust system identification technique is utilized to identify the dynamics of target motion, color, and appearance, providing model invalidation and worst-case estimates. A method for dynamics identification is an important part to achieve dynamic modeling.

• Nonlinear dimensional reduction and robust system identification are combined for dynamic appearance modeling. Direct modeling dynamic appearance is often a difficult problem due to the high dimensionality and nonlinearity. In this thesis, we provide an efficient framework that uses nonlinear dimensional reduction and robust system identification for dynamic appearance modeling.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 briefly reviews popular techniques for visual tracking. Chapter 3 discusses the Caratheodory-Fejer approach. Chapter 4, 5, 6, and 7 present the topics of dynamic motion modeling, dynamic homography modeling, dynamic color modeling, and dynamic appearance modeling respectively. Finally, Chapter 8 concludes the thesis and provides ideas for future work.

\(^2\) A priori information for dynamics identification will be discussed in Chapter 3.
Chapter 2

Visual Tracking Techniques

2.1 Introduction

Visual tracking is the repeated process of localizing a particular point, object, or region of interest in consecutive frames of a video sequence. Many visual tracking techniques have been developed for various purposes. While the early tracking algorithms have concentrated on tracking feature points [45], recent approaches have been extended to a region, or a contour, or a 3D surface of the target. In this chapter, we briefly introduce several visual tracking techniques used for this thesis.

2.2 Visual Tracking Techniques

While the image itself, in general, is widely used as target information such as in template matching algorithms, different forms of information are often required for target representation. One popular representation is to use a probability distribution, which describes the probability that the pixel belongs to the target region using color, texture, or other forms of target information.

The following two techniques use popular two-frame tracking, which means only the previous frame and current frame are used for computation of the target position. In this framework, the tracker is initialized using the estimate in the previous frame.

2.2.1 Template Matching

The simplest method of visual tracking is to match a template $T(x)$ onto an image $I(x)$ minimizing the Sum of Squared Difference (SSD) under translation.

$$D = \sum_i \| I(x) - T(x + u) \|^2$$  \hspace{1cm} (2.1)

where $x$ denotes the image coordinate $[x, y]^T$ and $u$ denotes a translation vector $[\Delta x, \Delta y]^T$.

Based on the first order approximation of the Taylor series, the translation $u$ can be iteratively calculated with two steps [45].
1. Compute the offset \( \bar{u} \)

\[
\bar{u} = \sum_i (g_i g_i^T)^{-1} \sum_i g_i e_i
\]  

(2.2)

where \( i \) denotes the index of pixels of the template \( T(x) \), \( g_i \) the gradient vector of \( I(x_i) \) and \( e_i \) the error between \( I(x_i) \) and \( T(x_i + u) \).

2. Update \( u_{j+1} \)

\[
u_{j+1} = u_j - \bar{u}
\]

(2.3)

This approach can be extended to, for example, an affine transformation [24], projective transformation [63][10], and 3D template matching [33]. The well known Kanade-Lucas-Tomasi(KLT) tracker[61] combines texture information of the template as good features.

---

Fig. 2.1. Visual tracking using template matching. The first image of the top left hand corner is a template and the second image is the warped target taken from the image. The affine parameters are calculated minimizing the error between the template and the warped target.

---

Figure 2.1 illustrates template matching using an affine transformation. In this example, a color image was first transformed to a gray image by averaging RGB pixel values and a rectangular template that contained the target object, a book held by a person, manually extracted from the first frame of a sequence of the images. The tracking process by template matching was then started from the initial position where
the target template was extracted. The tracking process for a frame is continued until the error between the template image and the captured image by affine transformation of the template image is smaller than a user-defined threshold. As shown in Frame 460, in which the warped target image in the second row of the top left hand corner is slightly brighter than the template target image in the first row of the top left hand corner, this method can handle illumination changes because it uses gradient information of the template image. However, a constant template image throughout the image sequence may fail to track the target due to its appearance change by viewpoint change or articulated body motion. This problem will be discussed in Chapter 7, in which an efficient approach for handling appearance variations will be provided.

2.2.2 Mean Shift Algorithm

An alternative approach for localizing the center of the target region is the mean shift tracking algorithm \[16][14], which progressively updates the object position only using the pixels in the target model (for example, search window). The estimated locations \( x \) of the object are iteratively calculated as follows:

\[
x_{j+1} = \frac{\sum_i t_i w(x) g(\|x_j - t_i\|^2)}{\sum_i w(x) g(\|x_j - t_i\|^2)}
\]

(2.4)

where \( i \) represents the pixel index of the target model such as a search window, \( w(\cdot) \) is a weight function, \( g(\cdot) \) is the derivative of a particular kernel function, and \( t_i \) is the pixel location of the target model in an image. The weight function can be the probability that the pixel belongs to the target object. Color information of the target is generally used to calculate the weight at the pixel location. This method will be discussed further in 4.3.1. A popular choice of a kernel function is the Epanechnikov kernel

\[
K_E = \begin{cases} 
\frac{1}{2} c^{-1}(1 - \|x\|^2), & \|x\| \leq 1 \\
0, & \text{otherwise} 
\end{cases}
\]

(2.5)

Since the derivative of the Epanechnikov kernel profile is constant, then Equation (2.2.2) reduces to a weighted distance average

\[
x_{j+1} = \frac{\sum_i t_i w(x)}{\sum_i w(x)}
\]

(2.6)

This result indicates that the local mean of the target model is the estimated position of the target object. The schematic procedure is shown in Figure 2.2. The
position in the previous frame is used as the initial position of the search window in the beginning of the tracking process. The search window is iteratively moved to the local mean calculated within the search window. The tracking process is continued until the difference between the previous position and the translated current position of the search window is smaller than the user-defined threshold. The size of the target object can then be estimated by the area of the probability (weight) distribution captured by the search window. If the size of the search window is smaller than that of the target object, for example, the area of the probability distribution is almost the same as that of the search window. Then the size of the search window can be increased to capture the whole area of the probability distribution. This method is known as the continuously adaptive mean shift (Camshift) algorithm [8].

![Fig. 2.2. The procedure of the mean shift algorithm. The location of the target region is iteratively calculated by moving the search window to the local mean.](image)

The mean shift algorithm is a very attractive method, because it is a simple, nonparametric method for searching for the peak of a probability density function. Thus it is widely used in visual tracking when the target object is represented by a probability distribution.

### 2.3 Visual Tracking with Dynamics

Note that an initial guess of the target position for the above techniques is very important for accurate estimates and computational costs. The visual tracking techniques, using a template or search window, strongly depend on a local search so that the
produced estimate can be detained in trivial local minima. This problem happens often because the previous estimate is, in general, used as the initial guess in their iterative methods. When the displacement of the target object is bigger than the search region, the tracker completely fails to track the target object. Therefore, the use of dynamics has been attempted, with the conjunction of several filtering techniques to predict the future location of the target.

2.3.1 Dynamics Model

In general, a linear dynamic system can be described by a state space model:

\[ x_k = Ax_{k-1} + Bu_k \]
\[ y_k = Cx_k + Du_k \]

(2.7)

where \( x \) represents the state of the system at time \( t \), \( y \) the output (measurement) of the system, \( u \) the input of the system, \( A \) is the state matrix, \( B \) is the input matrix, \( C \) is the output (measurement) matrix, and \( D \) is the feedthrough matrix. From the viewpoint of visual tracking, the state \( x \) can be the position and velocity of the target object and the measurement \( y \) can be the position of the target object, only when the location of the target object is concerned during the tracking process. These parameters, however, can be extended to any dynamic variables that describe visual or latent information of the target object. In general visual tracking systems, the feedthrough matrix \( D \) is set to zero for simplicity. In this thesis this parameter is also used for accurate dynamics modeling by dynamics identification, which will be described in the next chapter.

The conventional dynamics model for visual tracking is the constant velocity dynamics, which assumes that the target object moves with constant velocity. The state space representation of constant velocity dynamics is given as follows.

\[
\begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_k \\
y_k
\end{bmatrix} =
\begin{bmatrix}
x_{k-1} \\
y_{k-1}
\end{bmatrix}
\]

(2.8)

where the state \( x = \{x, y, v_x, v_y\} \) denotes the position and velocity of the target object and the measurement \( y = \{x, y\} \) denotes the position only. The sampling time is assumed
1 for simplicity. This dynamics represents that the current position can be predicted by using the previous position and velocity. The predicted position can then be used for the initial estimate of the two frame tracking methods described in the previous section.

In general, the process of the state transition and measurement acquisition is corrupted by external noise, for example, external forces to the target object and noisy image acquisition. To overcome these difficulties, the Kalman filter [34] and the particle filter [29] are generally utilized by assuming a Gaussian noise distribution and a non-Gaussian noise distribution, respectively.

2.3.2 Kalman Filter

The Kalman filter [34] is an optimal recursive filter which estimates the state of a dynamic system from a series of incomplete and noisy measurements. It supports estimations of past, present, and even future states. Thus, the position and velocity of the target are often estimated by the Kalman filter in consecutive frames.

The Kalman filter is based on two models, the system model and the measurement model. The system model is an equation describing the evolution of the state in time and the measurement model represents the relation between the noisy measurement and the true system state:

\[
\begin{align*}
    x_k &= Ax_{k-1} + Bu_k + w_k, \quad w_k \sim N(0, Q) \\
    y_k &= Cx_k + v_k, \quad v_k \sim N(0, R)
\end{align*}
\]  

(2.9)

where \(x_k\) denotes a state of system at step \(k\), \(A\) a state matrix, \(u_k\) a known input, \(y_k\) a measurement, \(C\) a measurement matrix, \(w_k\) and \(v_k\) represent the process noise and the measurement noise. The state estimation is calculated by using a recursive form. The equations for the Kalman filter are divided into two stages: time update and measurement update. The time update equations project the current state forward to obtain a prior estimate (“Predict”). The measurement update equations incorporate a new measurement into the prior estimate to calculate an improved a posteriori estimate (“Correct”). The Kalman filter algorithm is summarized as follows:

- Time Update (“Predict”)

\[
\begin{align*}
    \hat{x}_k^- &= Ax_{k-1} + Bu_k \\
    P_k^- &= AP_{k-1}A^T + Q
\end{align*}
\]  

(2.10)
• Measurement Update ("Correct")

\[
K_k = P_k^T C^T [CP_k^T C + R]^{-1}
\]

\[
\hat{x}_k = \hat{x}_k^- + K_k [y_k - C\hat{x}_k^-]
\]

\[
P_k = [I - K_k C] P_k^-
\]

(2.11)

where \(y_k - C\hat{x}_k^-\) is the measurement innovation (the discrepancy between the predicted measurement, \(C\hat{x}_k^-\), and the actual measurement), \(K_k\) is the Kalman gain that minimizes the \textit{a posteriori} error covariance, and \(P_k\) is the \textit{a posteriori} estimate error covariance.

When the state of the system is defined as \(x = \{x, y, v_x, v_y\}\) (position and velocity of the target object). The predicted state \(\hat{x}_k^-\) can then be used as an initial guess of two-frame tracking techniques in each frame and a posterior state \(\hat{x}_k\) can be calculated by incorporating the measurement provided by two-frame techniques.

In general, tracking objects involve nonlinear and non-Gaussian system models. The Kalman filter, however, is limited to linear and Gaussian assumption. To overcome these problems, the Kalman filter has been extended for nonlinear function, using the first order approximation of the Taylor series (extended Kalman filter: EKF) or using the unscented transformation [32] (the unscented Kalman filter: UKF [31][67]). While these extensions allow for nonlinear functions, EKF requires the liberalization of the nonlinear functions in every time instant with expensive computational cost and UKF requires predefined nonlinear dynamics to propagate their sample points. Moreover, both methods are still restricted to the Gaussian model.

2.3.3 Particle Filter

In contrast to the Kalman filter, the state of the tracked object can be estimated by using a non-parametric method such as the particle filter which approximates the posterior distribution of the state using a set of weighted samples (particles). Because of its non-parametric nature, it provides a probabilistic framework capable of handling nonlinear dynamics as well as non-Gaussian noise models. In the pioneer work of Condensation [29], factored sampling was used to generalize a particle filter. More recently, Merwe and Doucet have formulated a general particle filter, which uses Monte Carlo simulations with sequential importance sampling and resampling to estimate the state distribution [65].
The particle filter recursively approximates the posterior distribution using a finite set of weighted samples. It consists of essentially two steps: prediction and update. Given all available observations $z_{1:k-1} = \{z_1, \ldots, z_{k-1}\}$, the prediction stage uses the probabilistic state transition model $p(x_k|x_{k-1})$ to predict the posterior at time $k$ as

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (2.12)$$

At time $k$, the observation $z_k$ is available, the state can be updated using Bayes’ rule

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (2.13)$$

where $p(z_k|x_k)$ is described by the observation equation. In the particle filter, the posterior $p(x_k|z_{1:k})$ is approximated by a finite set of $N$ samples $\{x^i_k\}_{i=1}^N$ with importance weights $w^i_k$. The candidate samples $x^i_k$ are drawn from an importance distribution $q(x_k|x_{1:k-1}, z_{1:k})$ and the weight of the samples are

$$w^i_k = w^{i-1}_k \frac{p(z_k|x^i_k)p(x^i_k|x_{k-1})}{q(x^i_k|x_{1:k-1}, z_{1:k})} \quad (2.14)$$

The samples are resampled to generate an unweighted particle set, according to their importance weights to avoid degeneracy. In the case of the conventional particle filter - Condensation, the state transition model $p(x_k|x_{k-1})$ is used as the importance distribution $q(x_k|x_{1:k-1}, z_{1:k})$ and the weight then becomes the observation likelihood $p(z_k|x_k)$.

When the particle filter is applied to visual tracking, the probabilistic state transition model $p(x_k|x_{k-1})$ can be defined as a linear state transition model as follows.

$$x_k = Ax_{k-1} \quad (2.15)$$

where $x = \{x, y, v_x, v_y\}$ and a popular choice of $A$ is constant velocity dynamics. Even though the particle filter supports a nonlinear state transition model, obtaining an exact motion model at every frame is not always possible; therefore, a simple linear model is often used. The observation model is used to measure the observation likelihood of the samples; it is an important issue for visual tracking. Many observation models have been built for particle filter-based tracking. In [29], a contour based appearance template was chosen to model the target. When the particle filter is combined with the template matching method, the observation likelihood can be calculated by the sum of square differences (SSD) between the template and the image region that is captured by the template image. The position of the template image is determined by the sample $\tilde{x}^i_k$. 
2.4 Discussion

The combination of visual tracking techniques and several prediction methods are common methods used to estimate the state of the target object under noisy measurements. However, a drawback of all the above methods is the use of simplistic dynamics such as constant velocity. While constant velocity dynamics is a very simple and efficient way to estimate a single step of the state, it cannot generate reliable estimates beyond further steps. Thus, the tracker might fail to track the target object in the presence of clutter or occlusion due to inaccurate predictions.

Camps et al. [11] and Lim [38] have shown that these problems are explained by a lack of accurate dynamics. In their work, identified dynamics from real data successfully generated long-term predictions during occlusion. These predictions allowed the tracker to follow the target object even without observations. While the motion was the only approach for dynamics modeling in their work, the subject can be extended to any dynamic elements of the target object as discussed in Chapter 1.

The following chapters will introduce the robust system identification theory and its application to dynamic modeling will be illustrated with examples.
Robust System Identification for Dynamics Identification

3.1 Introduction

Most trackers assume a simple dynamics model such as a system moving with constant velocity. While successful in many scenarios, this approach suffers from the fact that the tracker must rely on the assumed model of the target dynamics to produce estimates of its future information, for example, position, color, or appearance, which introduce a potential source of fragility. A mismatch between this model and the actual dynamics will lead to an incorrect prediction\(^1\). This lack of robustness is illustrated with several examples in [11].

Note that more precise dynamic models have been tried in the particular case of human motion tracking. For example, models based on biomechanics have been successfully used to produce computer-based animations [9] and track an articulated human body [70]. However, biomechanical models are usually very complex and difficult to estimate from visual data alone. An alternative approach was proposed in [52] in which models were learned from a training corpus of observed state space trajectories. A drawback of this approach is that the states of the system must be specified \textit{a priori}, which requires \textit{a priori} knowledge of the order of the model.

In addition, stochastic approaches can neither provide a region guaranteed to contain the target nor falsify the information about the distribution. That is, if this information is rendered obsolete, for instance, by the target entering a region where the clutter is substantially different from the one used in training, a probabilistic approach would typically not be able to establish correspondences with certainty. Then the present scenario is no longer compatible with the assumptions.

In this chapter we show that all of the above issues can be addressed by \textit{identifying} the relevant dynamics of the target from a small set of frames. This method is accomplished by reducing the tracking problem to that of establishing the existence of an \(\ell_2\) to \(\ell_2\) operator that satisfies certain interpolation conditions. This scheme allows for exploiting convex analysis and integral quadratic constraints methods, which were recently developed (mostly in the control community). These methods recast the problems into a Linear Matrix Inequality (LMI) optimization form. A LMI optimization problem can be efficiently solved using commercially available tools.

The benefits of using this new framework are multiple:

\(^1\) This is the well known divergence phenomenon, see for instance [1]
It allows to approach the tracking problem from an input-output point of view. Thus, it does not require prior knowledge of a state space realization of the system, or even its order. Moreover, as we will illustrate with an example in the sequel, it can be used to model not only the motion of the target, but also its dynamic appearance as well.

- It allows to naturally incorporate the knowledge of the system dynamics whenever it is available.
- It can learn the relevant dynamics from a few initial set of frames in the video sequence.
- It provides mechanisms to invalidate a priori assumptions about the dynamics of the target and the noise characterization.
- It provides worst-case estimates of the identification error that can be used both to determine for how long the prediction of the target position (appearance) are valid, in the context of robust filter such as $\mathcal{H}_2/\mathcal{H}_\infty$ [57], and to improve tracking robustness.
- Finally, as we illustrate with several examples in the next chapters, the proposed method can be combined with existing Kalman [34], particle filter [29], and unscented particle filtering techniques [65] leading to algorithms capable of robustly tracking targets in the presence of severe occlusion. When compared to existing techniques, this combination allows for improving robustness significantly, while at the same time reducing the computational complexity of the resulting algorithm.

### 3.2 Preliminary

In this section, dynamic properties of targets, such as time evolution of features or appearance, will be represented using linear operators responding to some input signal $u$, with output $y$. The outputs are sequences of time values of measured quantities, such as position, size, pixel value, or other available information of the moving target.

From an input-output viewpoint any linear operator of interest $S$ will be represented by its convolution kernel $\{s_{i,j}\}$ or by an infinite lower triangular matrix $T_S$ mapping (scalar) sequences in $\ell_2$:

$$
\begin{bmatrix}
  y_0 \\
  y_1 \\
  y_2 \\
  \vdots
\end{bmatrix} = 
\begin{bmatrix}
  s_{0,0} & 0 & 0 & \cdots \\
  s_{1,0} & s_{1,0} & 0 & \cdots \\
  s_{2,0} & s_{2,0} & s_{2,0} & \cdots \\
  \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\begin{bmatrix}
  u_0 \\
  u_1 \\
  u_2 \\
  \vdots
\end{bmatrix}
\quad (3.1)
$$
When dealing with input-output sequences on the horizon \([0, n - 1]\), we will use the finite upper left submatrix of \(n \times n\), \(T_{S}^{n}\), obtained from the infinite matrix above.

In the sequel, we will also represent a finite dimensional Linear Time Invariant (LTI) operator by using either a minimal state-space realization:

\[
\begin{align*}
x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k + Du_k
\end{align*}
\]  

or a (rational) complex-valued transfer function:

\[
S(z) = \sum_{k=0}^{\infty} s_k z^{-k}
\]

Refer to Appendix A for notations used in this thesis.

### 3.3 Background on Robust System Identification

In this section we summarize, for the sake of completeness, the background results on robust identification used in this thesis. A comprehensive treatment of the subject, as well as an extensive reference list, can be found for instance in the textbooks [50][13].

The field of system identification concerns itself with mechanisms and algorithms that process finite, partial, and corrupted data to yield abstract mathematical descriptions of real world systems.

Traditional identification approaches [44][43] assume that the data are corrupted by a stochastic process with known statistical properties and that the system to be identified has a prescribed model structure. Most of these identification procedures are based on least squares methods that estimate the parameters of the hypothesized models from the corrupted measurements. In these approaches the only source of uncertainty is the noise in the measurements while the prescribed model is assumed to be an accurate representation of the real system.

In many situations, for example, when measurements are known within an accuracy range or when the available statistical information might be questionable, deterministic bounded noise descriptions are a practical and sound alternative to stochastic ones. Using this approach, the problem of system identification can be formulated as finding the sets of parameter values that are consistent with the known noise bounds. A survey of set membership formulations for system identification can be found in [48].
Noise description is only one of the factors affecting the quality of an identified model. Perhaps a more important factor is the unrealistic presumption that a fixed model structure may fully represent the system to be identified: in practice, since only partial information of the physical system is available, model parameters might change due to first principles. These issues are addressed by robust system identification \cite{50,13}, which departs from traditional approaches by using a deterministic worst-case approach with no prior assumption about the order of the system. Instead, robust identification procedures are based on a priori assumptions on the class of systems and noise, and on the a posteriori experimental data. Using this information robust system identification algorithms find nominal models based on the experimental data, and worst-case identification error bounds over the set of models defined by the a priori information.

3.3.1 Information Consistency and Interpolatory Algorithms

Because the assumed a priori information is, in general, a quantification of the engineering common sense or simply a ‘leap of faith,’ there is no guarantee that it will be coherent with the a posteriori experimental data. Thus, robust identification procedures must always first test the consistency of both types of information. Once consistency has been established, the computation of a nominal model and a valid model error bound can be attempted. In this thesis we will concentrate on a specific class of algorithms, interpolatory \cite{12}, that allow for efficiently accomplishing all of these tasks. In addition, these algorithms are always guaranteed to converge as the information is completed and they are optimal to a factor of 2, in the sense that their worst-case error is never larger than twice the minimum achievable error over the set of all identification algorithms.

3.3.2 Caratheodory-Fejer Based Interpolatory Identification Algorithms

In this section we briefly review the properties of the specific interpolatory identification algorithm used in the thesis. This algorithm uses time-domain data and is based on the Caratheodory-Fejer lemma, which considers the problem of the existence of a bounded, causal, discrete, linear time invariant operator such that the first $n$ elements of its impulse response sequence are given:

\[
\text{Lemma 3.1 (Caratheodory-Fejer). Given a matrix valued sequence } \{L_i\}_{i=0}^{n-1}, \text{ there exists a causal, discrete-time, LTI operator } L(z) \in BH_{\infty} \text{ such that:}
\]

\[
L(z) = L_0 + L_1 z + L_2 z^2 + \ldots + L_{n-1} z^{n-1} + \ldots
\] (3.4)
if and only if
\[(T^n_L)^T T^n_L \leq 1\] (3.5)

where 1 denotes the identity matrix of compatible dimension.

Proof: See for instance [22], Chapter 1.

In the sequel we consider operator families of the form
\[S(z) = F_{np}(z) + F_p(z)\] (3.6)

where operator \(S(z)\) is described in terms of a nonparametric component \(F_{np}(z) \in BH_\infty(K)\) and a parametric component \(F_p(z)\). We will further assume that the parametric component \(F_p(z)\) belongs to the following class \(P\) of affine operators:

\[P = \{P(z) = \sum_{i=1}^{N_p} p_i F_i(z) = p^T \Phi(z), \ p \in \mathbb{R}^{N_p}\}, \] (3.7)

where the \(N_p\) components \(F_i(z)\) of vector \(\Phi(z)\) are known, linear independent, rational transfer functions.

**Lemma 3.2.** Given a scalar \(K\), and two vector sequences \((u, v)\), there exists an operator \(S \in S\) such that \(y = Su\) if and only if there exists a pair of vector \((\nu, p)\) satisfying:

\[M(\nu) = \begin{bmatrix} 1 & (T^N_L)^T \\ T^N_L & \frac{1}{K^2} \end{bmatrix} \geq 0 \] (3.8)

\[y = T_u p + T_u \nu\]

where \((P)_k = [F_1^k \ F_2^k \ \cdots \ F_{N_p}^k]_i\) denotes the \(k\)-th Markov parameter of the \(i\)-th transfer function \(F_i(z)\), \(\nu_k\) the \(k\)-th Markov parameter of the nonparametric component \(F_{np}\), respectively, and the scalar \(K\) is an upper bound of the \(\ell_2\) induced norm of \(F_{np}\).

Moreover, in this case, all such operators \(S\) can be parameterized in terms of a free parameter \(Q(z) \in BH_\infty\). In particular, the choice \(Q(z) = 0\) leads to the “central” model:

\[S_{central}(z) = p^T \Phi(z) + F_{np_0}(z)\] (3.9)
where an explicit state-space realization of $F_{npo}(z)$ is given by:

$$F_{npo}(z) = C_F (zI - A_F)^{-1} B_F + D_F$$  \hfill (3.10)

with

$$A_F = \{ A - [C_-^T C_- + (A^T - I)]^{-1} C_-^T C_- (A - I) \}^{-1}$$

$$B_F = [C_-^T C_- (A^T - A - I) - (A^T - I)A]^{-1} C_-$$

$$C_F = KC_+ - KC_+ \{ A - [C_-^T C_- + (A^T - I)]^{-1} C_-^T C_- (A - I) \}^{-1}$$

$$D_H = KC_+ \{ (C_-^T C_- + (A^T - I)A - C_-^T C_- (A - I))^{-1} C_-^T, \}$$

and

$$A = \begin{bmatrix} 0 & \mathbf{I}_{N \times N} \\ 0 & 0 \end{bmatrix}, \quad C_- = [\underbrace{1 \ 0 \ \ldots \ 0}_{N+1}]^T, \quad C_+ = \frac{\nu^T}{K}$$  \hfill (3.11)

Proof: See Theorem 18.5.2 in [2][51].

Finally, the following corollary addresses the issue that real plants are subject to some unknown but bounded noise.

**Corollary 3.1.** [51] Consider the problem of identifying an operator $S \in S$ from measurements of its output $y$ to a known input $u \in \ell_2[0,N]$, corrupted by additive bounded noise $\eta$ in a given set $sN$:

$$y_k = (S \ast u)_k + \eta_k, \quad k = 0, 1, \ldots, N.$$  \hfill (3.13)

Then there exists $S \in S$ that satisfies (3.13) if and only if there exists a pair of vectors $(\nu, p)$ such that $M(\nu) > 0$ and $y - T_u p - T_u \nu \in N$. In that case, one such operator is given by $S_{central} = p^T \Phi + F_{npo}$, where $F_{npo}$ has the state-space realization (3.11).

### 3.4 Multiframe Tracking as an Interpolation Problem

In this section we show that the problem of robustly tracking an object in a sequence of frames is equivalent to finding an $\ell_2$ to $\ell_2$ operator that satisfies certain interpolation conditions. This approach allows for appealing to integral quadratic constraint (IQCs) [47], convex analysis and interpolation concepts to recast these problems into a tractable LMI optimization form.
As mentioned in the introduction, in principle, the location of the target can be predicted using a combination of its (assumed) dynamics, empirically learned noise distributions and past observations [29]. As shown in [11] this process is far from trivial in a cluttered environment. As shown there, both Kalman-based and unscented particle filter-based trackers perform poorly and lose the target, due to a combination of occlusion and the use of dynamics that do not exactly match the exact dynamics of the target. As we show next, these difficulties can be solved by modeling the motion of the target as the output of an ARMA model and by using the results in Section 3.3 to identify the relevant dynamics.

Specifically, assume that the present value of a given target feature, such as position or size, \( f_k \) is related to its past \( N \) values by\(^2\):

\[
\begin{align*}
    f_k &= \mathcal{A}f + Bu \\
    y_k &= f_k + \eta_k
\end{align*}
\]  

(3.14)

where \( f = (f_{k-1} f_{k-2} \ldots f_{k-N})^T \) contains the past observation of the feature, \( u = (u_{k-1} u_{k-2} \ldots u_{k-N})^T \) represents a stochastic input, \( y_k \) denotes the available measurement of the feature, corrupted by noise \( \eta_k \), and where \( \mathcal{A} \) and \( B \) are suitable LTI operators. Alternatively, (by taking \( z \)-transforms in the equation above), one can use the description:

\[
y(z) = F(z)u(z) + \eta(z)
\]  

(3.15)

where the operator \( F \) is not necessarily \( \ell_2 \) stable. In the sequel, we will assume that the following \( \text{a priori} \) information is available:

1. A set membership description of the process noise and the measurement: \( \eta_k \in \mathcal{N} \) and \( u_k \in \mathcal{U} \). These sets can be used to impose correlation constraints.

2. The operator \( F \) admits a finite expansion of the form \( F = \sum_{i=1}^{N_p} p_i F^i + F_{np} \). Here \( F^i \) are known, given, not necessary an \( \ell_2 \) stable operator that contains all the information available about possible modes of the feature of the target\(^3\). An example of

\(^2\)For simplicity, we consider a single feature, but the equations generalize trivially to the multiple-feature case.

\(^3\)If this information is not available the problem reduces to purely non-parametric identification by setting \( F^i \equiv 0 \).
this situation is tracking the position of moving persons, where $F^d$ can be obtained off-line by training with a representative set of motions [4], [21].

3. The residual operator $F_{np} \in \mathcal{B} \mathcal{H}_{\infty, \rho}(K)$ for some known $\rho \leq 1$. That is, a worst case estimate is available of how fast the approximation error of the finite expansion $F_p = \sum_{j=i}^{N_p} p_i F^d$ diverges.

In this context, the next value of the target feature $f_k$ can be predicted by first identifying the relevant dynamics $F$ and then using it to propagate its past $N$ values. In turn, identifying the dynamics entails finding an operator

$$F(z) \in \mathcal{S} \doteq \{ F(z) : F = F_p + F_{np} \}$$

such that

$$y - \eta = Fu$$

precisely the class of interpolation problem addressed in Corollary 3.13 in Section 3.3. By noticing that

$$F_{np}(z) = \mathcal{B} \mathcal{H}_{\infty, \rho} \iff F_{np}(\frac{z}{\rho}) = \mathcal{B} \mathcal{H}_{\infty},$$

it follows that such an operator exists if and only if the following set of equations is feasible:

$$M_{R}(\nu) = \begin{bmatrix} R_{\rho} & T_{\nu}^T \\ T_{\nu} & K^2 \nu^{-2} R_{\rho}^{-1} \end{bmatrix} \geq 0$$

$$y - T_u P p - T_u \nu \in \mathcal{N}$$

where $R_{\rho} \doteq \text{diag}[1 \rho \cdots \rho^N]$.

This is a linear matrix inequality (LMI) problem that can be efficiently solved using commercial software such as the LMI Matlab Toolkit. In addition to providing an estimate of the feature value of the target, this approach provides, as discussed next, mechanisms for model (in)validation and worst-case bounds on the prediction errors.

### 3.4.1 Model (in)validation

Assume that the set $\mathcal{N}$ is described by a set of LMIs of the form

$$\mathcal{N} \doteq \{ \eta \in \mathbb{R}^N : \Gamma(\eta) = \Gamma_0 + \sum_{k=1}^{N} \Gamma_k \eta_{k-1} \geq 0 \}$$

(3.19)
where $\Gamma_k$ are given real-valued symmetric matrices. This noise set is a generalization of the set $\{\eta \in \mathbb{R}^N : |\eta_k| \leq \epsilon\}$ usually considered [12], which allows for taking into consideration correlated noise (see [51] for details).

Then equations (3.18)-(3.19) reduce to a set of LIMs in the variable $\nu, \eta$, and $K^2$. This allows for finding the minimum value of $K$, which is an upper bound of the $\ell_2$ induced norm of the non-parametric part of the operator $F_{np}$, such that the LMI (3.18)-(3.19) are feasible.

In turn, this value $K$ can be used as a "sanity check" to assess the quality of the approximation. A large value of $K$ indicates that the parametric portion of the model $F_p$ does not provide a good description of the motion of the feature, hence the need for a large non-parametric part, indicating that it may be necessary to re-identify the set $\{F_i\}$.

On the other hand, infeasibility of the LMI's indicates that the experimental data is not compatible with the a priori assumptions, possibly an indication either that: (i) a new target activity not described by elements of the set $\{F_i\}$ or (ii) the target entering a region where the noise and clutter models are no longer compatible with the description (3.19). Either case points to the need for re-assessing the a priori information.

### 3.4.2 Worst-case estimates of the prediction error

By construction, the operator found from the solution of the LMI's (3.18) that its response to the input $u$ interpolates within the experimental noise level $\eta_k$, the given value of the feature $f_k$, $k = 0, 1, \ldots, N - 1$. However, when used to predict the future value of the feature, it is of interest to obtain bounds on the worst case prediction error. Next we describe how this can be accomplished.

Given a sequence $\{y_k\}_{k=0}^{N-1}$ of measurements of the value $f_k$ of the feature, consider its consistency set:

$$
\mathcal{T}(y) = \left\{ F \in \mathcal{S} : \{y_k - (F \ast u)_k\}_{k=0}^{N-1} \in \mathcal{N} \right\}
$$

(3.20)

i.e, the set of all models is consistent with both the a priori information and the experimental data. Since the proposed method is interpolatory, from Lemma 3 in the appendix, it follows that a bound on the worst case prediction error over the horizon $[0, M - 1], M > N$, given by:
\[ \| \hat{f} - f \|_{\ell_\infty[0,M-1]} \leq \sup_{F_1,F_2 \in T(y)} \| F_1 * u - F_2 * u \|_{\ell_\infty[0,M-1]} \]
\[ = d[T(y)] \]
\[ \leq \sup_y d[T(y)] = D(I) \] (3.21)

where \( d(\cdot) \) and \( D(I) \) denote the diameter of the set \( T(y) \), in the \( \ell_\infty[0, M - 1] \) metric and the diameter of information as defined in the appendix, respectively. Moreover, since the \( a \ priori \) sets \( (S, N) \) are convex and symmetric, with points of symmetry \( F_s = 0 \) and \( \eta_s = 0 \) respectively, from Lemma 4 in the appendix, it follows that:

\[ D(I) \leq 2 \sup_{F \in T(0)} \| F \|_{\ell_\infty[0,M-1]} \] (3.22)

where \( T(0) \) indicates the set of operators compatible with the zero outcome: \( y_k = 0, k = 0, 1, \ldots, N - 1 \). Computing this bound reduces to a convex optimization problem.

### 3.5 Discussion

#### 3.5.1 Applications to Robust Visual Tracking

The most critical factor in visual tracking is loss of reliable sources from visual information due to clutter or occlusion, leading the trackers to lose their vision such that they fail to track the targets. Even though many filtering techniques, such as Kalman and particle-based filters [34][28][31][65] support prediction of future values based on the past observations, these techniques are restricted to the fact that the measurements are always available to be incorporated into their estimation.

In addition, many tracking techniques assume that the object follows the constant velocity assumption for simplicity. While this assumption is effective in many scenarios, it does not contain the specified dynamics of the target feature; thus, the prediction is no longer valid after one frame in which observation is not found. The characterized dynamics, especially for certain motions such as acceleration and harmonic motion, for example, can produce more predictions. These assumed dynamics, however, should be predefined before tracking is started. Thus, they do not have any information for the exact motion of the target.

If the visual observation is not available due to long term occlusion, the only reliable information for prediction of the target feature is the observed dynamics of the
target before occlusion. Therefore, an ultimate solution to the problem of the absence of relevant dynamics is to identify the dynamics of the target feature based past observations. As discussed in the previous sections, the CF-based system identification provides a useful tool for finding relevant dynamics of the target feature, including additional benefits. To apply the CF approach to robust visual tracking, two types of selections should be preceded before tracking the targets and performing identification. The next sections will describe the requirements to run the CF-based identification.

3.5.2 Selection of the Input and Output of the System

First of all, the input and output of the system should be defined before the identification process. In most cases the input of the system is assumed as an impulse signal at the beginning of the measurement \cite{11}\cite{39}\cite{40}. This assumption is simple and useful when the target features are not affected by external factors, for example, sudden direction change by a force or an impact. The external factor, however, is not easily detected from visual information without physical sensing from the actual target object. Also the false visual information due to the presence of clutter or occlusion makes its detection difficult. Therefore, it is assumed that no external factor exists during the identification and tracking processes.

In other cases when additional information of the target from a different source is available, it can be used as an input of the system. For instance, if the visual information of the target is highly correlated with other visual representation such as multiple camera tracking where the only difference is the viewpoint of the target, the input and output can be defined using each image view of the cameras, respectively \cite{66}. In this thesis, since we concentrate on robust tracking using only visual information of the target itself from a single camera, we assume that the system is initialized by an impulse input.

The output of the system is often selected directly from visible information of the tracking results, such as position or color variation \cite{11}\cite{39}. The output of the system, however, can be extended to any dynamical feature of the target or even latent variables that describe visual variation of the target appearance. In \cite{40} variables of the embedded manifold are used as the output for modeling dynamic appearance.

The number of measurements used for the output of the system depends on the complexity of the motion observed in the measurements. For instant, constant velocity motion can be easily identified with a very small number of measurements. Periodic motion needs to be observed at least one period of the motion. The more complex motion the more number of measurements are required for dynamic identification. In this thesis, all the measurements from the first frame to the frame when the target is occluded are used for the identification process.
3.5.3 Selection of the Parametric Component and the Nonparametric Component

As discussed before, the operator $F$ consists of parametric component $F_p$ and nonparametric component $F_{np}$. The parametric parts are given by the user before identifying the system as a priori information and the non-parametric parts are obtained as the byproduct of the identification process. The parametric components play an important role because they often affect the feasibility of the identified system and the accuracy of identification. Note that it is also possible to identify the system only with non-parametric parts if the system is assumed to be stable. The parametric parts, however, can more efficiently explain the dynamics of the system using well defined parametric representation so that the valid prediction may last longer than by using only the non-parametric parts.

The next question is: What type of and how many parametric parts are necessary for the identification process? The type of parametric component can be chosen based on the type of motions that can be represented by transfer functions. In other word, any type of motion can be a candidate for the parametric components, if it has a certain transfer function. In other assumed dynamics, such as constant velocity or acceleration, only one model can be used as the dynamic model. However, with the CF-based identification, many possible candidate transfer functions can be used as parametric components. The coefficient of each parametric component is automatically calculated in the process of identification such that only crucial components for interpolation remains as significant factors for future prediction.

In general, the more parametric parts, the more feasible identification. However, the following set of transfer functions are used initially for the parametric part for generality.

$$F_p \in \text{span} \left\{ \frac{z}{z - 1}, \frac{z}{(z - 1)^2}, \frac{z(z + 1)}{(z - 1)^3}, \frac{z}{z - a}, \frac{z \sin(\omega)}{z^2 - 2z \cos(\omega) + 1}, \frac{z(z - \cos(\omega))}{z^2 - 2z \cos(\omega) + 1} \right\}$$  

(3.23)

The transfer functions with a very small coefficient in Equation (3.4) are excluded from the set if the system is still feasible without such transfer functions. The above transfer functions were empirically chosen from extensive experiments and proven to be appropriate for dynamics identification of target motion, color, and appearance as shown in the following chapters.

Figure 3.1 illustrates the time domain representation of the transfer functions given in Equation (3.23) when an impulse signal is used as an input of the system. The
step function $\frac{z}{z-1}$ is used for an initial state of the model such as the initial position of the target object. The ramp function $\frac{z}{(z-1)^2}$ is used for monotonous increasing or decreasing states such as constant velocity of the target object. The parabolic function $\frac{z(z+1)}{(z-1)^3}$ and the exponential function $\frac{z}{z-a}$ are used an accelerating state such as constant acceleration of the target object. Finally, the sine function $\frac{z\sin(\omega)}{z^2-2z\cos(\omega)+1}$ and the cosine function $\frac{z(z-\cos(\omega))}{z^2-2z\cos(\omega)+1}$ are used for a periodic state such as a periodic motion of the target object. The fast Fourier transform can be utilized specifically for periodic motions [40], which will be given in Chapter 7. Small variations of the above transfer functions and other transfers can be used for better identification. Throughout the thesis, the parametric components were selected manually based on the feasibility of the system. Further research needs to be conducted for automatic choices for parametric parts.

The nonparametric parameters $K$ and $\rho$ in Equation (3.4) are also important to obtain a feasible system with given transfer functions and measurements. A given scalar $K$ represents the portion of the nonparametric part in the interpolation of the measurements. If the dynamics of the measurements can be well represented with the parametric components, the amount of the nonparametric component needs to be decreased with a small value of $K$. On the other hand, if the parametric components are not sufficient to represent the dynamics of the measurements, a larger value of $K$ is chosen to increase the amount of the nonparametric component. The stability parameter $\rho$ of the nonparametric component should be less than one, assuming that the nonparametric component does not increase above the noise bound. In this thesis, $\rho$ is set to 0.99 in all the experiments, which was chosen experimentally.

### 3.5.4 Accuracy and Computation Time

The CF approach interpolates the measurements within the given noise bound $\mathcal{N}$ in Equation (3.4). Thus, if the system is feasible, it is guaranteed that the identified dynamics produces at most $\mathcal{N}$-valued errors between the estimated measurements and the actual measurements. If the dynamics of the measurements does not change after the dynamics identification, the error also does not increase over time. If the dynamics changes after the identification process, it can be detected by the model (in) validation mechanism in Section 3.4.1 and then the identification process needs to be conducted again.
Fig. 3.1. Basic transfer functions for the parametric part $F_p$. (a) step function $u(t) \leftrightarrow \frac{z}{z-1}$, (b) ramp function $t \leftrightarrow \frac{z}{(z-1)^2}$, (c) parabolic function $t^2 \leftrightarrow \frac{z(z+1)}{(z-1)^3}$, (d) exponential function $e^t \leftrightarrow \frac{z}{z-a}$, (e) sine function $\sin(t) \leftrightarrow \frac{z \sin(\omega)}{z^2 - 2z \cos(\omega) + 1}$, and (f) cosine function $\cos(t) \leftrightarrow \frac{z(z-\cos(\omega))}{z^2 - 2z \cos(\omega) + 1}$.
The computation time of the identification process depends on the number of the measurements, the number of transfer functions for the parametric components, and the parameters of the nonparametric component. For example, the more number of the measurements, the more computation time and the more transfer functions, the less computation time. Finally, the larger $K$, the more computation time. In real time visual tracking systems with 25 fps, all processes from target segmentation to target localization should be completed within 40 milliseconds. Even though the CF approach can be easily solved by the LMI toolbox in Matlab, the computation time of the CF approach still does not satisfy this requirement. Therefore, further work will include faster computation of the CF approach by using C or C++ languages for real time dynamics identification.

In the following chapters, we will show CF-based robust visual tracking techniques with several illustrative examples.
Chapter 4

Dynamic Motion Modeling

4.1 Introduction

The nature of motion by living creatures, for example, human or animal, or any moving object in real world is in general unpredictable from only their visual information unless physical analysis is not actually accomplished. One reliable belief, however, is that the observations from frame to frame, taken by visual sensing devices, are highly correlated so that such an assumption makes the prediction possible.

Such correspondences between individual frames are usually integrated over time to improve robustness by exploiting the dynamic properties of the target. Kalman filter-based trackers use a model of the target dynamics as well as the probability distribution of the process and measurement noise. This model produces estimates of the future positions of the target based on (noisy) measurements of its past locations. Condensation trackers [28] and unscented Kalman Filters [31] generalize Kalman filter based ones by allowing more general (multimodal, nonlinear) models. The unscented particle filter [65][59] (UPF) is also one of those extensions made by combining a particle filter (condensation) and the unscented Kalman filter. In this case, analytical propagation is no longer possible and numerical methods must be used instead.

In general, most trackers employing the above techniques use a simple dynamic model such as constant velocity motion. This approach, however, can introduce a potential source of fragility in the presence of clutter or occlusion. This is due to the fact that the tracker must depend on the assumed model of the target dynamics to produce estimates of its future positions regardless of various motion changes. A mismatch between this model and the actual dynamics will lead to incorrect predictions such that the tracker fails to track the target.

In this chapter we show that the above issues can be addressed by using the Caratheodory-Fejer (CF) based interpolation. As discussed in the previous chapter, modeling an $\ell_2$ to $\ell_2$ unknown operator that integrates past observation provides accurate predictions to future estimations of the target feature. A CF-based approach was initially introduced in the computer vision community for gait recognition [46], using its ability
to model (in) validation. Later, the application to visual tracking was attempted in our previous work, [38][11], in which the CF-based approach was used for dynamic motion modeling while the observation was no longer available during occlusion. The CF-based modeling is shown to model the target dynamics robustly and successfully cope with severe occlusion. In our previous work, the dynamic motion modeling was conducted off-line in such a way that the objective motion is not individualized for new motion. In this chapter, we extended dynamic motion modeling to online modeling in order to produce more accurate estimates.

As we illustrate with several examples, the proposed method can be combined with existing Kalman filtering techniques, leading to algorithms capable of robustly tracking targets in the presence of severe occlusion. When compared to existing techniques, this combination allows for significantly improving robustness, while at the same time reducing the computational complexity of the resulting algorithm.

4.2 The Proposed Approach

Figure 4.1 shows the procedure of the proposed approach to dynamic motion modeling. The color histogram of the target is first calculated after initializing the position and size of the search window in the first frame. Using the back projection of the obtained color histogram to the entire image, the image is converted to a probability distribution, where each pixel value represents the probability that the pixel belongs to the target region based on color similarity.

The mean shift algorithm [16] is then applied to search the peak of the distribution that represents the center of the target object. The object size can be estimated by the area of the probability distribution within the search window. The position obtained by the mean shift algorithm is then corrected by the Kalman filter under assumption of constant velocity dynamics. For later use, the N numbers of past estimated positions are always kept in the memory.

When the occlusion is detected, using the threshold of the size change of the target object, the stored past positions (measurements) are used for dynamics identification using CF interpolation. The tracked positions are used as the output of the system and the system is activated by an impulse signal. The identified dynamics are then replaced with the constant velocity dynamics in the Kalman filter. The following examples will illustrate that the combination of the Kalman filter and the identified dynamics produces accurate predictions, preventing the tracker from drifting due to occlusion.
Fig. 4.1. Past observations of the target locations are used for dynamic motion modeling using the CF-based identification. The prediction of future locations is produced based on the identified dynamics.
4.3 Illustrative Examples

The experiments illustrated in this chapter are divided into two categories: single camera tracking and multiple camera tracking, which will be presented in the following sections.

4.3.1 Experimental Setups

In the following experiments the initial position and size of the target object are manually selected in the first frame of each video sequence. The color information of the target is obtained from the inside region of the search window by applying RGB to HSV (Hue, Saturation, Value [Brightness]) transformation as follows.

Let $\text{MAX}$ equal the maximum of the (R,G,B) values, and $\text{MIN}$ equal the minimum of those values.

$$H = \begin{cases} 
\text{undefined}, & \text{if } \text{MAX} = \text{MIN} \\
60 \times \frac{G-B}{\text{MAX} - \text{MIN}} + 0, & \text{if } \text{MAX} = R \text{ and } G \geq B \\
60 \times \frac{G-B}{\text{MAX} - \text{MIN}} + 360, & \text{if } \text{MAX} = R \text{ and } G < B \\
60 \times \frac{B-R}{\text{MAX} - \text{MIN}} + 120, & \text{if } \text{MAX} = G \\
60 \times \frac{R-G}{\text{MAX} - \text{MIN}} + 240, & \text{if } \text{MAX} = B
\end{cases}$$

(4.1)

$$S = \begin{cases} 
0, & \text{if } \text{MAX} = 0 \\
1 - \frac{\text{MIN}}{\text{MAX}}, & \text{otherwise}
\end{cases}$$

$$V = \text{MAX}$$

The hue information (the color types such as red, green, and blue) of the search window region is quantized into 10 bins and the hue histogram of the target object is then obtained by counting the number of pixels that belong to each bin. The obtained histogram is then normalized such that the sum of all bins is equal to 1 (probability) or 255 (gray value). This hue information is known to be robust to illumination change making. Thus, it is widely used for color modeling of the target object.

Then, using the normalized color histogram, the probability that a pixel belongs to the target object can be obtained by looking up the normalized value of the bin that the hue of the pixel belongs. This process is called the histogram backprojection [62]. Note that the hue histogram of the target is assumed to be constant in the experiments conducted in this chapter. The dynamic modeling of a variable color histogram will be presented in the next chapter.
After the probability distribution of the target object is calculated by the histogram backprojection, the mean shift algorithm is then applied to find the centroid of the probability distribution. This process is performed in every frame with the obtained centroid being used as the measurement of the target position. As mentioned earlier, the initial position of the search window plays an important role in finding an accurate position of the target object and the obtained measurement is assumed to be noisy. Due to these facts, the Kalman filter is employed to predict the initial position and to estimate the posterior position.

Note that while obtaining the measurements using the Kalman filter before occlusion is detected, the constant velocity dynamics is used because the target dynamics are unknown. Also, the Kalman filter with the constant velocity dynamics performs well, unless the target object is occluded by the background object or other moving targets. The $N$ posterior positions estimated by the Kalman filter are always kept in the memory for later use in dynamics identification.

When the occlusion is detected using abrupt size change of the target object, the stored past posterior positions are used for dynamics identification by the CF interpolation in which the past measurements are used as an output of the dynamic system that needs to be identified. Occlusion detection is in general a difficult problem in visual tracking when the target object is slowly occluded by the background object or other moving objects. In this case, abrupt size change is not detected such that the tracker slowly reduces its velocity while tracking the unoccluded target region. Finally, the tracker remains stationary or nearly stationary at the point where the remained target region that is not occluded is completely occluded.

To handle this problem, two options are used in the following experiments by assuming constant size and varying size of the target object. If the target size is assumed to be constant, the difference between the initial size of the target and the tracked size is used for detecting occlusion. If the target size is assumed to be variable, the difference between the previous size at $t-1$ and the current size at $t$ is used for detecting occlusion. In both cases occlusion is detected when the absolute difference is bigger than a certain threshold.

Occlusion detection by size change is generally used when the color histogram is used as a target appearance model. However, if an exact appearance model exists, for example, an image template, occlusion can easily be detected by measuring the occluded region of the template image. This method is used in Chapter 7. The following condition shows the rule of occlusion detection.

Let $w_t$ and $h_t$ be the width and height of the target at time $t$, and $T_w$ and $T_h$ be a user-defined threshold for the width and height respectively.

\[
\begin{align*}
\text{constant target size} & : \\
& \begin{cases} 
\text{unoccluded,} & \text{if } w_0 - w_t \leq T_w \text{ and } h_0 - h_t \leq T_h \\
\text{occluded}, & \text{otherwise}
\end{cases} \\
\text{variable target size} & : \\
& \begin{cases} 
\text{unoccluded,} & \text{if } w_{t-1} - w_t \leq T_w \text{ and } h_{t-1} - h_t \leq T_h \\
\text{occluded}, & \text{otherwise}
\end{cases}
\end{align*}
\]
Regarding choosing a priori information for dynamics identification by the CF approach, as described in Section 3.5.3, there is no explicit method that automatically selects the parameters and the parametric component. However, it should be noted that the coefficient $p$ of the parametric components given as a priori information is computed in Equation (3.4) such that the significance of the parametric components is automatically determined during the convex optimization process. Even though unnecessary parametric components are given, those are virtually excluded by having a very small value of coefficients.

However, those unnecessary parametric components increase the order of the identified dynamic system because the number of the parametric components is directly related to the order of the system. Therefore, it is important to use a minimal set of the parametric components that results in a feasible system, because the order of the identified dynamics affects the computation of Kalman filtering process.

In all of the following experiments, the transfer functions for the parametric part were manually chosen until the system was feasible with minimum requirements. The input of the system was set to an impulse signal, assuming that there was no additional input to the system. The stability parameter $\rho$ of the non parametric component was set to 0.99, assuming that the resultant transfer function of non parametric component should be stable throughout interpolation and prediction. By setting $\rho \leq 1$ (inside of the unit circle), it is guaranteed that the predicted value by the non parametric component does not increase above the given noise bound. Also, worth mentioning is that the higher noise bound can produce the easier feasible solution. However, the higher noise bound affects the accuracy of the prediction of the future state. Thus if the system was feasible with given parametric and non-parametric components, a lower noise bound was tested to obtain better prediction accuracy.

After the system is identified by modeling an unknown operator $F$ with a priori information and the past measurements, this operator is in turn converted to $A, B, C,$ and $D$ matrices, such that the new Kalman filter with the identified dynamics model replaces the old Kalman filter with the constant velocity model. In the following experiments, the state $x$ is defined as $\{x, y, v_x, v_y\}$, position and velocity, and the process noise and the measurement noise are defined as an identity matrix for simplicity. In the beginning of the Kalman filtering, the posterior state covariance $P_0$ in Equation (2.3.2) is set to high so that the Kalman filter significantly incorporates the measurement to correct the posterior state.

Note that the CF-identified dynamics can be replaced with the constant velocity dynamics after occlusion to reduce the computational cost. In general, the order of the
CF-identified dynamics is much higher than that of the constant velocity dynamics due to incorporation of parametric and non-parametric components into its dynamics model. The computation cost during Kalman filtering with CF-identified dynamics is also higher than that with the constant velocity dynamics. Therefore, from the viewpoint of efficiency, the CF-identified dynamics needs to be used for occlusion handling in the above experiments. However, these experiments were aimed to demonstrate predictability of the CF-identified dynamics during occlusion. Switching from the CF-identified dynamics to the constant velocity dynamics was not considered throughout the experiments.

In summary, using the CF-identified dynamics only for occlusion handling, when the constant velocity dynamics can track the target object using the observed measurement, is more efficient.

In the following experiments, the backprojection method, the mean shift algorithm, and the Kalman filter were implemented using C and C++ languages. The CF approach was implemented using the LMI toolbox in Matlab. Even though two different platforms were used for implementation, all the processes were conducted online by communicating each other. In addition, the implemented tracking system is real time except the identification process.

### 4.3.2 Single Camera Tracking

In this section we illustrate how the proposed approach can be used to improve the robustness of Kalman trackers that rely on a combination of past measurements and the dynamics of the target to estimate its future location. Proceeding as in the previous work in [11] where the dynamics is modeled offline, we learn the online dynamic model that uses the sequence of positions of the target object in the scene. The obtained dynamics is then used in conjunction with a Kalman filter, leading to results shown in Figures 4.2 to 4.6.

Table 4.1 shows a combination of *a priori* information for the experiment shown in 4.2 and the *a posteriori* measurements of target locations from past 17 frames, where the target was not occluded, were used to estimate its dynamics. In this experiment, sinusoidal transfer functions with $\omega = 0.7$ were used for the parametric component to model periodic jumping by the target person. The noise bound was set to 15 to compensate the difference between the generated periodic motion by the sinusoidal transfer functions and the actual periodic motion by the jumping person.

In Figure 4.2 the blue color of the target person in the first frame was used as a target color model and the mean shift algorithm was used to find the centroid of the blue color. The reason why the captured size by the mean shift algorithm is smaller
Parameters | Conditions
--- | ---
Noise Bound | $\mathcal{N} = \{ \eta \in \ell_\infty, \| \eta \|_{\ell_\infty} \leq 15 \}$, i.e. in pixel unit.
Input of System | $u = \delta(t)$, i.e. motion of the target was modeled as the impulse response of the unknown operator $F$.
Output of System | $y = 17$ number of past positions $x$ and $y$ before occlusion.
Parametric Part | $F_p \in \text{span} \left\{ \frac{1}{z-1}, \frac{z}{z-1}, \frac{z^2 - \cos \omega z}{z^2 - 2 \cos \omega + 1}, \frac{\sin \omega z^2}{z^2 - 2 \cos \omega + 1} \right\}$ where $\omega = 0.7$
Nonparametric Part | $F_{np} \in \mathcal{BH}_{\infty, \rho}(K)$, with $\rho = 0.99$, i.e. unmodeled dynamics.

Table 4.1. *A priori* information for the jumping person sequence shown in Figure 4.2.

Frame 15 | Frame 20 | Frame 25 | Frame 28
--- | --- | --- | ---
(a) Kalman filter with constant velocity dynamics.

(b) Kalman filter with CF-identified dynamics.

Fig. 4.2. Comparison between constant velocity dynamics and CF-identified dynamics of the jumping person sequence where the person becomes occluded while jumping. The green line in the images of (b) represents the predicted target positions by CF-identified dynamics, which accurately predict the target positions even during occlusion.
than that of the target person is that the upper side dark blue region of the pants was initially selected in the first frame. The occlusion was detected by the size change of the target object, assuming that the size of the target person is constant.

As shown in Frame 20, when the target person is occluded by the other moving person, the Kalman tracker with constant velocity dynamics generated prediction based on the right previous velocity such that the tracker moved to the upper left side. However, the Kalman tracker with CF-identified dynamics changed its direction from the upper left to the lower left because the oscillating motion was captured by the sinusoidal transfer functions given in Table 4.1.

In Frame 25, the Kalman tracker with constant velocity dynamics completely lost the target person, while the Kalman tracker with CF-identified dynamics predicted the position where the target person jumped up again. However, it is shown that the predicted position by CF-identified dynamics is slightly different from the actual position of the target person. This disparity is due to the fact that the frequency of the sinusoidal transfer functions was not exactly same as the actual frequency of the jumping motion, because the frequency was manually selected in this experiment. However, in Chapter 7 the peak frequency of the fast Fourier transform was used for the frequency of the sinusoidal transfer functions when an oscillatory motion was observed from the measurement.

In Frame 28, the Kalman tracker with CF-identified dynamics successfully re-captured the target person when the blue colored region reappeared on the left side of the other person. Note that during occlusion the mean shift algorithm did not find any peak of probability distribution within its search window; thus, no measurement was produced. Therefore, the prediction process was continued until the measurement by the mean shift algorithm was observed again.

Figure 4.3 shows an example of tracking a paper plane in the lab and \textit{a priori} information is shown in Table 4.2. In this experiment, the \textit{a posteriori} measurements of target locations from the past 10 frames, where the target was not occluded, were used to estimate its dynamics. The ramp functions and sinusoidal functions with $\omega = 0.2$ for the parametric component were used to model a parabolic motion of the paper airplane that was flying down to the ground.

The orange color of the paper plane was modeled as a target color model and the position and size of the search window were manually chosen in the first frame. Occlusion was detected by the size change of the paper airplane. Then the paper airplane was thrown to the left side at the right top in the image. Due to the air resistance force that occurred in the wing, the paper airplane flew to the left bottom side, reducing its vertical velocity to the ground side and drawing a parabolic trajectory as shown in Frame 10.
Table 4.2. *A priori* information for the paper airplane sequence shown in Figure 4.3

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Bound</td>
<td>$\mathcal{N} = { \eta \in \ell_\infty, | \eta |<em>{\ell</em>\infty} \leq 10 }$.</td>
</tr>
<tr>
<td>Input of System</td>
<td>$u = \delta(t)$.</td>
</tr>
<tr>
<td>Output of System</td>
<td>$y = 10$ number of past positions $x$ and $y$ before occlusion.</td>
</tr>
<tr>
<td>Parametric Part</td>
<td>$F_p \in \text{span} \left{ \frac{1}{z-1}, \frac{z}{z-1}, \frac{z^2}{(z-1)^2}, \frac{z^2 - \cos \omega z}{z^2 - 2 \cos \omega z + 1}, \frac{\sin \omega z^2}{z^2 - 2 \cos \omega z + 1} \right}$ where $\omega = 0.2$.</td>
</tr>
<tr>
<td>Nonparametric Part</td>
<td>$F_{np} \in \mathcal{B} \mathcal{H}<em>{\ell</em>\infty}, \rho(K)$, with $\rho = 0.99$.</td>
</tr>
</tbody>
</table>

Fig. 4.3. Comparison between constant velocity dynamics and CF identified dynamics of the paper airplane sequence where the paper airplane becomes occluded, changing its direction. The green line in the images of (b) represents the predicted target positions by CF-identified dynamics, which accurately predicts the target positions even during occlusion.
When the airplane was not occluded, the Kalman tracker with the constant velocity dynamics successfully tracked the paper airplane because the measurement by the mean shift algorithm was available. When the airplane was occluded by the chair, the Kalman tracker with the constant tracker seemed to track the airplane. However, as shown in Frame 15, the Kalman tracker with the constant velocity dynamics predicted a higher and more right position than that of the Kalman tracker with the CF-identified dynamics. This result was due to the fact that the Kalman tracker with the constant velocity dynamics used the right previous velocity to predict the future position, without capturing the acceleration to the left side and deceleration to the ground side of the airplane motion. Nevertheless, the Kalman tracker with the CF-identified dynamics predicted the position of the airplane during occlusion by identifying the actual motion dynamics and successfully recaptured the airplane when it reappeared in the left of the chair in Frame 20.

Figure 4.4 shows an example of tracking a bouncing ball on a stair and a priori information is shown in Table 4.3. In this experiment, the a posteriori measurements of target locations from past 20 frames, where the target was not occluded, were used to estimate its dynamics. The ramp functions and sinusoidal functions with $\omega = 0.45$ for the parametric component were used to model a bouncing motion of the ball. The yellow color of the ball was modeled as a target color model and the position and size of the search window were manually chosen in the first frame. Occlusion was detected by the size change of the ball captured by the search window.

Similar to the results shown in Figure 4.2 and 4.3, the Kalman tracker with the constant velocity model incorrectly predicted the position of the ball during occlusion as shown in Frame 25, depending on the assumed dynamics. However, the Kalman tracker with the CF-identified dynamics successfully recaptured the ball when it reappeared in the left lower side of the trash bin, generating accurate predictions during occlusion. Even though the periodic motion of the ball is similar to the periodic motion by a person shown in Figure 4.2, the lower noise bound than that of Table 4.1 resulted in a feasible solution. This disparity is because the bouncing motion of the ball was closer to the actual periodic motion, unlike the intentional jumping motion by a person.

In Figure 4.5, the proposed method was used to learn the dynamics of the size of the target in a sequence where a car approaches the camera. Table 4.4 shows the priori information to model the size of the car. In this experiment, the a posteriori measurements of target locations and sizes from past 20 frames, where the target was not occluded, were used to estimate its dynamics of motion and size. The power functions with a coefficient $a$ and ramp functions for the parametric component were used to
Parameters | Conditions
---|---
Noise Bound | $\mathcal{N} = \{\eta \in \ell_\infty, \|\eta\|_{\ell_\infty} \leq 10\}$
Input Signal | $u = \delta(t)$
Output of System | $y = 20$ number of past positions $x$ and $y$ before occlusion.
Parametric Part | $F_p \in \text{span} \left\{ \frac{1}{z-1}, \frac{z}{z-1}, \frac{z^2}{(z-1)^2}, \frac{z^2 - \cos \omega z}{z^2 - 2 \cos \omega z + 1}, \frac{\sin \omega z^2}{z^2 - 2 \cos \omega z + 1} \right\}$ where $\omega = 0.45$.
Nonparametric Part | $F_{np} \in \mathcal{BH}_{\ell_\infty, \rho}(K)$, with $\rho = 0.99$.

Table 4.3. *A priori* information for the bouncing ball sequence shown in Figure 4.4

Frame 19 | Frame 25 | Frame 30 | Frame 35
---|---|---|---
(a) Kalman filter with constant velocity dynamics.

(b) Kalman filter with CF-identified dynamics.

Fig. 4.4. Comparison between constant velocity dynamics and CF-identified dynamics of the bouncing ball sequence where the ball becomes occluded, changing its direction. The green line in the images of (b) represents the predicted target positions by CF-identified dynamics, which accurately predicts the target positions even during occlusion.
model an accelerating motion and size change of the car. In this case, the appearance descriptors used were the vertical and horizontal sizes of the search window around the target. These sizes were calculated by comparing the hue histogram of the target against the initial hue histogram with a mean shift algorithm.

Unlike the previous experiments, occlusion was detected by the absolute difference between the size change of the target at time $t-1$ and the size change at time $t$ as given in Equation (4.3.1). As shown in Frame 27 and 30, the tracker with CF-identified dynamics accurately predicted the size change during occlusion based on the past observation, while the tracker with the constant velocity dynamics lost the target in the beginning of occlusion. This figure shows how the tracking algorithm recovers the target with the right size after the occlusion.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Bound</td>
<td>$\mathcal{N} = { \eta \in \ell_\infty, | \eta |<em>{\ell</em>\infty} \leq 5 }$</td>
</tr>
<tr>
<td>Input Signal</td>
<td>$u = \delta(t)$</td>
</tr>
<tr>
<td>Output of System</td>
<td>$y = 20$ number of past positions $x,y$, width $w$, and height $y$ of the search window before occlusion.</td>
</tr>
<tr>
<td>Parametric Part</td>
<td>$F_p \in \text{span}\left{ \frac{1}{z-1}, \frac{z}{z-a}, \frac{z}{(z-1)^2}, \frac{z^2}{(z-1)^2} \right}$ where $a \in {1.2, 0.9, 2, 1.3}$</td>
</tr>
<tr>
<td>Nonparametric Part</td>
<td>$F_{np} \in \mathcal{BH}_\infty,\rho(K)$, with $\rho = 0.99$.</td>
</tr>
</tbody>
</table>

Table 4.4. A priori information for the car sequence shown in Figure 4.5

### 4.3.3 Multiple Camera Tracking

The common method for locating the ground position from multiple images is triangulation. This method uses rays from the optical centers of the cameras to centroids of the objects. The object positions can be found by the intersection of such rays. In this experiment, the objects are detected by background subtraction and centroids of the objects are tracked by the Camshift algorithm [8] with the Kalman filter. The
Fig. 4.5. Comparison between constant velocity dynamics and CF-identified dynamics of the car sequence where the car becomes occluded, changing its size and direction.

(a) Kalman filter with constant velocity dynamics.

(b) Kalman filter with CF-identified dynamics.
top view position is estimated by triangulation and predicted by the Kalman filter. The assumption is made that the locations and orientations of the cameras are known. Because multiple rays do not exactly intersect at one point in general, we use the point that gives minimal distances to each ray. When multiple objects exist, there are more intersects than the number of objects. The true locations of the objects can be refined from spurious intersections by using color correspondences of the objects shown in each camera view [49]. Other information such as velocity or size of the object can also reinforce the true position.

Figure 4.6 demonstrates the triangulation method combined with the CF approach. When the target person in the right camera is occluded behind the wall, the predicted position in the right camera view is estimated by the identified dynamics modeled by the CF approach. Its top view position is calculated by the measured position in the left view and the predicted position in the right view by triangulation. In this experiment, the Kalman filter with constant velocity dynamics failed to generate accurate predictions during occlusion even though the observed motion before occlusion was a constant velocity. This result was due to the fact that the target person was slowly occluded by the wall, such that the occlusion was not detected correctly. However, using the CF approach, the incorrect measurement right before occlusion was considered to be corrupted by noise within the given noise bound and then did not significantly affect the identified dynamic.

4.4 Discussion

In the past few years, dynamic vision techniques have proved to be a viable option for a large number of applications, ranging from surveillance and manufacturing to assisting individuals with disabilities. Arguably, at this point, one of the critical factors limiting widespread use of these techniques is the potential fragility of the resulting systems. In this chapter we show that in the case of multi-frame tracking this fragility can be addressed by using interpolation and LMI tools recently developed in the control community to recast these problems into a tractable optimization form. The advantages of this approach and in particular its potential to result in robust tracking algorithms when combined with existing filtering techniques was illustrated with several experimental results.

Also worth mentioning is that there are important cases in which this approach does not provide a complete solution, since the resulting problem is not convex on all the variables involved, leading to an \( NP \) hard optimization problem. An example of
Fig. 4.6. Triangulation method with CF approach to occlusion handling. CF-identified dynamics accurately predicts the position of the target in the right camera view when the target is occlusion by the wall.
this situation is the case in which both the dynamics of the plant and its input must be identified, and the experimental data is corrupted by measurement noise.

As mentioned earlier, the CF approach can be extended to various dynamic modeling where dynamics of particular target features must be identified. A sequence of position (trajectory) of the target is not the only dynamic feature that can be observed from its visual information. The trajectory of the tracked target can be trustworthy when the target is correctly identified. The experiments conducted in this chapter assume that the target of the color histogram does not change while tracking; thus, the initial estimation is used until the object region is found in the predicted region.

The color of the object, however, can be changed by illumination or its viewpoint. This problem requires dynamic color modeling of the target color to identify the target correctly. Chapter 6 will introduce a solution to this problem using the same approach used in this chapter. Before moving onto dynamic color modeling, the next chapter will introduce dynamic homography modeling in the context of camera motion modeling. In this modeling, the dynamics of the camera motion is identified and used for predicting its future motion, similar to the dynamic motion modeling described in this chapter.
Chapter 5

Dynamic Homography Modeling

5.1 Introduction

Recently, multiple camera tracking has become a common requirement for human motion recognition, activity analysis, and visual surveillance due to the limitations of single camera tracking. The benefits of using multiple cameras include: providing a larger coverage range, estimating the three dimensional position of moving targets, and aligning static surroundings, all of which are not easily implemented in single camera tracking without certain constraints.

Many researchers have studied how to estimate 3D position of the moving object and alignment of static surroundings, using multiple cues from each camera views and several assumptions. A common approach to finding 3D position of moving objects is the triangulation method, which estimates 3D ground position under the assumption that the location and orientation of the cameras are known [49][18][15]. In this method, the ground position is calculated by finding the intersection of multiple rays from optical centers of the cameras to the objects. However, the triangulation method can estimate only target position at once.

Another method is to use homography estimation image-to-image under assumptions of planar constraints that the objects move on the planar ground [36][71]. The homography estimation is calculated using the ground marks or the trajectories of the target position, and the obtained homography matrix can be used for an alignment of the camera views. This method is useful to find the moving object as well as corresponding other static objects on the planar ground. In addition, it also provides an efficient occlusion handling when the object is not shown in the other camera due to occlusion [71].

In most scenarios attempted in the previous experiments using the homography matrix, the critical assumption was that all the cameras should be stationary in order to use the homography parameters constantly over time. However, it is necessary to move cameras to cover a larger area and track the objects efficiently with its surrounding. Then the homography matrix at each time needs to be updated over time while the cameras move.
In this chapter, we tackle this problem through exploiting the dynamics of the homography matrix instead of finding corresponding points at each time. The dynamics of the homography parameters are captured by the Caratheodory-Fejer approach [11][39] and produce predictions based on the dynamics. The main goal of this chapter is to analyze the dynamics of the homography matrix and to predict the future homography matrix based on a set of previous observations, instead of using the corresponding points directly to calculate the homography matrix.

5.2 Homography Estimation

If it is assumed that the objects move on the planar ground, we can derive the homography transformation using a set of corresponding points [36][71][26] as shown in Figure 5.1. In this figure $x_i$ and $x'_i$ represent a projected point of $x_{\pi}^i$ on the real world plane $\pi$ onto the camera image plane. $H$ represents the homography transformation matrix that maps $x_i$ to $x'_i$. $C$ and $C'$ represent the optical center of each camera. In general, corresponding points can be obtained by the feature matching method between two images. When there exists a moving object, then the tracked positions in each camera can also be used as corresponding points. In this section the tracked positions by the mean shift algorithm were used to obtained a set of corresponding points.

![Fig. 5.1. Homography method for multiple cameras.](image)

Given a set of 2D to 2D corresponding points, $x_i \leftrightarrow x'_i$, $x_i$ from Camera 1 and $x'_i$ from Camera 2, the transformation is expressed by $x'_i = Hx_i$ where $H$ is a 3 by 3 matrix.
The equation can be expressed in terms of the vector cross product as \( \mathbf{x}' \times H \mathbf{x} = 0 \). This form enables a simple linear solution for \( H \) to be derived.

If \( h^T_j \) denotes the \( j \)th row of the matrix \( H \). Then \( H \mathbf{x}_i \) can be written as follows:

\[
H \mathbf{x}_i = \begin{bmatrix}
h^T_1 \mathbf{x}_i \\
h^T_2 \mathbf{x}_i \\
h^T_3 \mathbf{x}_i 
\end{bmatrix}
\]  

(5.1)

Letting \( \mathbf{x}' = (x'_i, y'_i, w'_i)^T \), the cross product is then given as:

\[
\mathbf{x}'_i \times H \mathbf{x}_i = \begin{bmatrix}
y'_i h^T_3 \mathbf{x}_i - w'_i h^T_2 \mathbf{x}_i \\
w'_i h^T_1 \mathbf{x}_i - x'_i h^T_2 \mathbf{x}_i \\
x'_i h^T_2 \mathbf{x}_i - y'_i h^T_1 \mathbf{x}_i 
\end{bmatrix}
\]  

(5.2)

Because \( h^T_j \mathbf{x}_i = \mathbf{x}_i h_j \) for \( j = 1, \ldots, 3 \), the above equation can be written by a set of three equations in the form:

\[
\begin{bmatrix}
0^T \\
-w'_i \mathbf{x}_i^T \\
-w'_i \mathbf{x}_i^T \\
\end{bmatrix}
\begin{bmatrix}
y'_i h^T_3 \mathbf{x}_i - w'_i h^T_2 \mathbf{x}_i \\
w'_i h^T_1 \mathbf{x}_i - x'_i h^T_2 \mathbf{x}_i \\
x'_i h^T_2 \mathbf{x}_i - y'_i h^T_1 \mathbf{x}_i 
\end{bmatrix}
= \begin{bmatrix}
h_1 \\
h_2 \\
h_3 
\end{bmatrix} = 0
\]  

(5.3)

These equations have the form \( A_i h = 0 \), where \( A_i \) is a \( 3 \times 9 \) matrix and \( h \) is a \( 9 \times 1 \) vector made of the entries of the matrix \( H \):

\[
h = \begin{bmatrix}
h_1 \\
h_2 \\
h_3 
\end{bmatrix}, \quad H = \begin{bmatrix}
h^1 & h^2 & h^3 \\
h^4 & h^5 & h^6 \\
h^7 & h^8 & h^9 
\end{bmatrix}
\]  

(5.4)

where \( h^i \) is the \( i \)th element of \( h \). Although there are three equations in (5.3), only two of them are linearly independent. since the third row is obtained by the sum of \( x'_i \) times the first row and \( y'_i \) times the second row. Thus, each corresponding point gives two equations in the entries of \( H \). Then the set of equations becomes:

\[
\begin{bmatrix}
0^T \\
-w'_i \mathbf{x}_i^T \\
-w'_i \mathbf{x}_i^T \\
\end{bmatrix}
\begin{bmatrix}
y'_i h^T_3 \mathbf{x}_i - w'_i h^T_2 \mathbf{x}_i \\
w'_i h^T_1 \mathbf{x}_i - x'_i h^T_2 \mathbf{x}_i \\
x'_i h^T_2 \mathbf{x}_i - y'_i h^T_1 \mathbf{x}_i 
\end{bmatrix}
= \begin{bmatrix}
h_1 \\
h_2 \\
h_3 
\end{bmatrix} = 0
\]  

(5.5)
The above equation is then written as $A_i h = 0$ where $A_i$ is now the $2 \times 9$ matrix. This equation holds for any homogeneous coordinate representation $(x'_i, y'_i, w'_i)^T$ of the point $x'_i$. By setting $w'_i = 1$, $(x'_i, y'_i)$, the equation represents the coordinate measured in the image.

Then Equation (5.5) can be expressed as follows:

$$
\begin{bmatrix}
0 & 0 & 0 & -x_i & -y_i & -1 & y_i x'_i & y_i y'_i & y'_i \\
x_i & y_i & 1 & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i & -x'_i
\end{bmatrix}
\begin{bmatrix}
h_1 \\
h_2 \\
h_3
\end{bmatrix} = 0
$$

(5.6)

Each point correspondence provides two independent equations in the entries of $H$. Therefore, given a set of four such point correspondences, we can obtain a set of equations $A h = 0$, where $A$ is the matrix of equation coefficients built from the matrix rows $A_i$ contributed from each correspondence, and $h$ is the vector of unknown entries of $H$. In either Equation (5.3) and (5.5) $A$ has rank 8, and thus has a 1 dimensional null space that provides a solution for $h$. This method is known as the direct linear transform (DLT) algorithm [26]. In summary, the DLT algorithm is given as follows:

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x'_i\}$, determine the 2D homography matrix $H$ such that $x'_i = H x_i$.

1. For each correspondences $\{x_i \leftrightarrow x'_i\}$, form the matrix $A_i$ from Equation (5.2)

2. Obtain the SVD of $A$. The unit singular vector corresponding to the smallest singular value is the solution $h$.

3. The matrix $H$ is determined from $h$ as in Equation (5.2)

Once the homography $H$ is calculated, any point $x$ in Camera 1 can be transformed to the point $x'$ in Camera 2 using the homography matrix. This property is very useful to find the correspondence when the target point in Camera 1 is not shown in Camera 2 due to occlusion or scene clutter.

Figure 5.2 shows how the homography matrix can be utilized for handling occlusion occurred in other camera. The first rows of (a) and (b) were taken by Camera 1 and the second rows were taken by Camera 2.

In this experiment, the toy car was segmented by the backprojection method using the color histogram modeling introduced in the previous chapter. Then the mean
shift algorithm was applied to find the position of the toy car at each frame and the Kalman filter was used to estimate the posterior position. The occlusion was detected by assuming the constant target size. The homography matrices $H_{12}$ and $H_{21}$, Camera 1 to Camera 2 and vice versa, were obtained from the correspondences of the tracked positions of the toy car, until Frame 37 in which the toy car was not occluded in both camera views. The difference between the trackers in Figure 5.2 (a) and (b) is that the tracker in (a) used the only constant velocity dynamics for occlusion handling but the tracker in (b) used the homography matrix to estimate the target position from the other camera view when the target object was occluded.

As shown in Frame 46 of (a), the Kalman tracker with the constant velocity dynamics in Camera 1 (first row) started loosing the toy car when it was occluded, depending only on the right previous velocity before occlusion. However, in Frame 46 of (b) the position of the toy car in Camera 1 was estimated using the homography matrix $H_{21}$ even though the toy car was not shown in Camera 1. In Frame 59 of (a), the Kalman tracker in Camera 1 completely lost the toy car and the Kalman tracker in Camera 2 started loosing the toy car. However, in Frame 59 of (b) the Kalman tracker in Camera 1 recaptured when the toy car reappeared in the camera view and the Kalman tracker in Camera 2 estimated the position of the toy car using the homography matrix $H_{12}$ even though the toy car was not shown in Camera 2. Frame 64 shows that the Kalman trackers with the constant velocity dynamics completely lost the toy car while the Kalman trackers supported by the homography matrices successfully recaptured the toy car, when the toy car was shown again in the camera views.

### 5.3 The Proposed Approach

If one of the cameras moves, the homography matrix also needs to be recomputed at every frame because the previous correspondences between two camera views are invalid due to moving the camera. Sometimes it is very difficult to compute the homography matrix at every frame when not enough numbers of corresponding points are available due to occlusion or scene clutter. To overcome this difficulty, we utilize the dynamics of each homography parameter to predict the homography matrix instead of directly using the corresponding points. In this chapter, we assume that one camera is stationary and the other camera is moving. First, the homography matrix at each time instant is computed using a set of ground points for a given time period. Then the dynamics of each homography parameter is identified by the CF approach. The identified dynamics are then used for predicting the future homography matrix.
Frame 37
Frame 46
Frame 59
Frame 64

(a) Kalman filter with constant velocity.

(b) Position estimation using homography matrix.

Fig. 5.2. Comparison between the Kalman filter with constant velocity model and homography matrix for occlusion handling. The first rows of (a) and (b) were taken by Camera 1 and the second rows were taken by Camera 2. Even though the target object is occluded, the position can be estimated from the tracked position in the other camera view using the homography matrix.
5.4 Illustrative Examples

In this experiment, a set of each parameter for a given time period is used as the output of the system that must be identified. As illustrated with an example, the predicted homography parameters by the CF-identified dynamics were virtually as same as the ground truth homography parameters at the same time instant, not actually using the corresponding points.

5.4.1 Experimental Result

Figure 5.5 shows image sequences used in this experiment, in which Camera 1 in (a) was stationary while Camera 2 moved from the left to the right side. In both camera views, grid ground marks were drawn to provide point correspondences between two camera views. Even though finding correspondences from two camera views is of great interest in the computer vision area, in this experiment it was assumed that the correspondences of the ground marks were known at every frame and the numbers of corresponding points were at least more than 4 to calculate the homography matrix.

In this experiment, the dynamic homography transformation from the static camera view to the moving camera view was modeled using the CF approach. Let \( h_t = [h_t^1 \ldots h_t^9]^T \) be the homography parameters calculated at time \( t = \{1, \ldots, N\} \) by using the given set of corresponding points. \( h_t \) is then scaled by multiplying \( 1/h_t^9 \) to each entry of \( h_t \) such that new \( h_t \) becomes \( \{h_t^1, \ldots, h_t^8, 1\} \). This normalization is necessary because the homography matrix is in general determined to a non-zero scale factor. The set of each parameter \( h_t^j \) for time \( t = 1, \ldots, N \) is then used as the output of the system that will be identified by the CF-approach. After the identification process, the predicted homography parameters at time \( t + 1 \) are combined into the homography matrix \( H_{t+1} \). Then the positions of the corresponding points at time \( t + 1 \) are estimated from the positions of the points in the other camera view, multiplying the predicted homography matrix \( H_{t+1} \). All these processes were implemented using Matlab and the LMI toolbox in Matlab.

A priori information for this experiment is given in Table 5.1. The homography matrices for time \( t = 1, \ldots, 15 \) were calculated by known corresponding points and then 8 sets of 15 past homography parameters were used for dynamics identification. In this experiment, the Kalman filter was not used because the position of each point was measured by hand in each camera view. The noise bounds for each parameter were
Fig. 5.3. The images in the first row come from the static camera for reference and the images in the second row come from the moving camera.
chosen based on their ranges. For example, the noise bound of $h_3$ was set to 5 because it ranged from 65 to 106 while that of $h_7$ was set to 0.01 because it ranged from -0.0013 to -0.00105. The all parametric components in Table 5.1 were given in the first test and were reduced until the system was feasible. Other parameters $a$ and $\omega$ were also chosen based on the system feasibility. The sinusoidal transfer functions with very low $\omega$s were used to model parabolic motions of the parameters.

Figure 5.4 shows the ground truth homography parameters for $t = 1, \ldots, 15$ and the estimated homography parameters for $t = 1, \ldots, 20$ by the CF approach. As shown in the result, the CF approach interpolated the ground truth measurements for 15 frames and predicted the future homography parameters for the last 5 frames. Especially, Figure (g) and (h) show that the estimated homography parameters were interpolated without incorporating fluctuations of the ground truth homography parameters. This result indicates that the fluctuations were considered as noises such that those were not modeled during the identification process. These fluctuations were caused by their very small values compared to other parameters. Thus, very small noises affected the measurements significantly. These noises can come from the erroneous measurements by hand and camera distortion.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
</tr>
</thead>
</table>
| Noise Bound         | $\mathcal{N} = \{\eta \in \ell_\infty, \|\eta\|_{\ell_\infty} \leq 0.1(h_1, h_3, h_5, \text{ and } h_4),\nonumber$
|                     | $0.01(h_7 \text{ and } h_8), 1(h_6), 5(h_3)\}.\nonumber$            |
| Input Signal        | $u = \delta(t)$.                                                           |
| Output of System    | $y = 15$ set of past coefficients of homography matrices $h_1 \sim h_8$. |
| Parametric Part     | $F_p \in \text{span} \left\{ \frac{1}{z-1}, \frac{z}{z-1}, \frac{z}{(z-a)^2}, \frac{z^2}{(z-1)^2}, \frac{z^2-\cos \omega z}{z^2-2\cos \omega z+1}, \frac{\sin \omega z^2}{z^2-2\cos \omega z+1} \right\}$ where $a \in \{1, 1.05\}$ and $\omega \in \{0.0001, 0.1, 0.2\}$. |
| Nonparametric Part  | $F_{np} \in \mathcal{BH}_{\infty, \rho}(K)$, with $\rho = 0.99$.          |

Table 5.1. *A priori* information for the moving camera sequence shown in Figure 5.4.
Fig. 5.4. Dynamic homography modeling using the CF-identified dynamics. The blue lines are the ground truth measurements for 15 frames and the red lines are the estimations by the CF approach for 20 frames where the last 5 frames are predicted by the identified dynamics of each parameter.
Figure 5.5 shows how the estimated homography matrices by the CF approach, shown in 5.4, are accurate, compared with the homography matrices computed using the corresponding points. As shown in Frame 18, the predicted positions of the corresponding points by the CF approach are virtually the same as those by the ground truth homography matrices at time 18. This result shows that the variable homography matrix can be estimated by the dynamics of each parameter without finding corresponding points at every frame.

![Frame 1 Frame 6 Frame 12 Frame 18](image)

(a) Moving right camera.

Fig. 5.5. The images in the first row come from the static camera for reference and the images in the second row come from the moving camera. The blue rectangles are the ground truth measurements of the corresponding points of Figure (a) and the red crosses are the estimated positions by the estimated from the dynamic homography matrix by the CF approach.

### 5.5 Discussions

Two of most important problems arising in multiple camera tracking is 1) to find correspondence of feature points from camera to camera and 2) to identify not only moving objects but also static objects. In this chapter, we have presented the dynamic homography modeling that predicts the future homography matrix based on the identified dynamics of the past homography parameters. As illustrated with an example, the homography matrix was accurately predicted without recalculating the parameters using corresponding points at every frame. This approach is very useful when corresponding points are invisible in a moving camera due to occlusion. One restriction assumed in this
chapter was that one of two cameras should be stationary. In further research, dynamic homography modeling for the case that both cameras are moving will be conducted using the trajectories of the target position and the ground points (or feature points) shown in both cameras.
Chapter 6

Dynamic Color Modeling

6.1 Introduction

One of the most important challenges to tracking a target successfully is to overcome changes of its appearance that might occur over time. These changes, including size, shape, and color can be due to several factors such as target motion, self-occlusion, target articulations, and changes in illumination.

Several researchers have proposed methods to design flexible tracking algorithms. Many of these approaches focus on tracking the position of the target and only address changes in appearance indirectly by using probabilistic methods to compare the target to some nominal template using, for example, pixel statistics [53][16][73]. Other techniques focus on building adaptive appearance models. For example, Jepson et al. [30] use an online EM algorithm to adapt the appearance of a target template over time. Hager and Belhumeur [24], Black and Jepson [6] and Ho et al. [27], among others, learn target appearance models using linear subspaces.

While successful in many scenarios, these approaches suffer from the fact that the obtained models tend to be too rigid and fail to capture the dynamics of the color changes. As a result, tracking remains susceptible to incorrect measurements due for example to partial or self occlusion.

In this chapter, we show that all of the above issues can be addressed by identifying dynamics of the color changes from a small number of initial frames. This is accomplished by reducing the problem of learning the color models to that of establishing the existence of an $\ell_2$ to $\ell_2$ operator that satisfies certain interpolation conditions. The next sections illustrates the proposed technique with several examples.

6.2 The Proposed Approach

Figure 6.1 illustrates the proposed approach used in this chapter. The overall procedure is similar to dynamic motion modeling discussed in Chapter 4. The dynamic color histogram modeling, however, is appended to the existing dynamic motion modeling.

The color histograms are updated using the previous target region captured by the search window. Then the mean shift algorithm is applied to search the peak of the
Fig. 6.1. Past observations of the target locations and color are used for dynamic motion and color modeling using CF-based identification. The prediction of future locations and color histogram are produced based on the identified dynamics.
probability distribution obtained by the backprojection method of the color histogram. N number of both histogram data and position data are always kept in the memory for later use. When occlusion is detected using the threshold of the size change between the previous and the current search window, the dynamics of color histogram as well as position are identified using the CF approach. The identified dynamics is then used for a prediction of color and position during occlusion. The following section will show the robustness of the proposed approach.

6.3 Illustrative Examples

In this section we present several examples to illustrate how the proposed approach can be used to improve the robustness of trackers in the presence of appearance changes. In the proposed approach, the tracking algorithm combined the Kalman filter with appearance and motion dynamics learned, using the CF-based identification algorithm. Then the proposed approach was compared with the existing color modeling techniques.

6.3.1 Experimental Setups

The following experiments conducted here compare the robustness of three different types of color modeling: constant color modeling, adaptive color modeling, and finally dynamics color modeling. Each model uses the hue histogram modeling with 10 bins as an appearance descriptor. In all the color modelings, only several significant histogram bins that show variations are manually selected for color modeling because the target objects used in the experiments have only a few strong colors.

In the constant color modeling, the hue histogram of the target object is obtained from the initial search window in the first frame. Then the obtained hue histogram is used constantly for the backprojection method for all the remaining frames. Therefore, only one hue histogram modeling is performed in the first frame.

In the adaptive color modeling, the hue histogram of the target object is updated at every frame based on the captured target region by the mean shift algorithm. The initial search window is placed manually in the first frame. After that, the target region to calculate the hue histogram is determined automatically by the tracking result of the mean shift algorithm. The obtained color model at time \( t \) is then used for the backprojection method at time \( t+1 \). Therefore, the hue histogram modeling is performed at every frame. The procedure of the adaptive color modeling is as follows:
1. Apply the backprojection method using the previous histogram.

2. Apply the mean shift algorithm to capture the target region.

3. Recalculate the hue histogram using the captured target region.

4. Perform Step 1-3 repeatedly at every frame.

The dynamic color modeling utilizes the same method of the adaptive color modeling to obtain the hue histogram at time $t$ before occlusion is detected. However, unlike the adaptive color modeling that preserves only the previous hue histogram at time $t - 1$, the previous hue histograms for the past $N$ frames are preserved for dynamic identification using the CF approach. When occlusion is detected, the identified dynamics of each hue histogram bin is used to predict the current and future value of the corresponding hue histogram bin. The predicted hue histogram is then used for the backprojection method. In all the three color modelings, the occlusion detection rule for the variable target size in Equation (4.3.1) was applied to detect occlusion using the target size given by the mean shift algorithm. In the following experiments, the backprojection method, the mean shift algorithm, and the Kalman filter were implemented using C and C++ languages. The CF approach was implemented using the LMI toolbox in Matlab. In addition, the implemented tracking system is real time except the identification process.

The difference between the adaptive color modeling and the dynamic color modeling is that when an occlusion is detected, the adaptive color model keeps incorporating wrong color information for its adaptive update process, while the dynamic color model predicts accurate color information against observed wrong color information due to occlusion.

The constant motion modeling (constant velocity dynamics) and the dynamics motion modeling introduced in Chapter 4 are also combined with the above three color modelings to predict the target position during occlusion. Then the differences of several combinations with three color modelings and two motion modelings will be provided in the next section.

6.3.2 Experimental Results

In the first example shown Figures 6.2, the proposed technique was used to track a multicolored ball that was rolled in front of the camera. Three histogram bins, 1(orange), 6(blue), and 9(red), for 18 frames were selected for color modeling because the remaining histogram bins were nearly zeros due to the actual colors of the target ball. The variation of the three bins is illustrated in Figure 6.3 in which the blue lines are the measured
values by the adaptive color modeling and the red lines are the predicted value by the dynamic color modeling. \textit{A priori} information for the dynamic color modeling is shown in Table 6.1 in which sinusoidal functions were used for the parametric component by assuming that the ball rolls.

Figure 6.2 (a) shows Frame 1 and Frame 10 of the sequence of tracking a ball, in which the visible part of the ball varied from mostly blue (Frame 1) to a mixture of orange, blue, and red (Frame 10). Figure 6.2 (b), (c), and (d) show the probability distribution calculated by the backprojection method using the proposed histograms by the constant color modeling, the adaptive color modeling, and the dynamic color modeling, respectively. The black color represents high probability that the ball exists and the white color represents low probability. Note that the gray color (from black to white) of the target ball is difficult to model using the hue color histogram because they contain evenly distributed RGB values. Therefore, the hue value along the gray color is very sensitive to small variation of RGB values. To avoid this problem, if the pixel color is close to a gray value, the pixel is excluded in the hue histogram modeling. This result of this condition is shown in both the first and second row of (b), (c), and (d) in which the white or black region of the ball has very low probability (white color in the probability image).

The second row of Figure 6.2 shows that how three color modelings handle color variation. In the second row of (b), the constant color model failed to produce a correct probability distribution, depending on the only initial hue histogram of the ball when the color of the ball changed from mostly blue to a mixture of orange, blue, and red. In the second row (c), the adaptive color model produced a correct probability distribution unless the region of the ball was very bright or dart. As mentioned earlier, this result is a limitation of the hue histogram modeling. The dynamic color model shown in the second row (d) performed similarly to the adaptive color model. This result is because the dynamic color model interpolated the measurements by the adaptive color model. However, the advantage of the dynamic color modeling is to provide a predictive mechanism, which will be presented in the next example.

This lack of robustness of the constant color modeling and the adaptive color modeling is illustrated in Figure 6.4, showing the effects of appearance change. This figure shows a few frames from a sequence in which a blue and red football is tossed in the air. The first row shows the result for tracking the ball using a combination of the constant color model and the Kalman filter with the constant velocity dynamics. The second row shows the result of a combination of the constant color model and the
Fig. 6.2. (a) Frame 1 (first row) and Frame 10 (second row) of a multicolored ball. (b) to (d) probability of a pixel belonging to the ball, according to the histogram from the (b) first frame (constant color model), (c) previous frame (adaptive color model), and (d) predicted by the CF-based dynamic color model.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Bound</td>
<td>( \mathcal{N} = { \eta \in \ell_\infty, |\eta|<em>\ell</em>\infty \leq 10 }, ) histogram unit.</td>
</tr>
<tr>
<td>Input of System</td>
<td>( u = \delta(t) ).</td>
</tr>
<tr>
<td>Output of System</td>
<td>( y = 18 ) number of past values of histogram bin.</td>
</tr>
<tr>
<td>Parametric Part</td>
<td>( F_p \in \text{span} \left{ \frac{z}{z-1}, \frac{z^2 - \cos \omega z}{z^2 - 2 \cos \omega z + 1}, \frac{\sin \omega z^2}{z^2 - 2 \cos \omega z + 1} \right} ) where ( \omega = 0.35 ).</td>
</tr>
<tr>
<td>Nonparametric Part</td>
<td>( F_{np} \in \mathcal{B}\mathcal{H}_{\infty,\rho}(K), ) with ( \rho = 0.99 ).</td>
</tr>
</tbody>
</table>

Table 6.1. *A priori* information for the rolling ball sequence shown in Figure 6.2
Kalman filter with the CF-identified dynamics. In both cases, since only the blue side of the ball is initially visible, these approaches fail to track the ball when it becomes red, even using the identified motion dynamics.

The third row shows the result of a combination of the adaptive color modeling and the Kalman filter with the constant velocity dynamics. While this approach can cope with color change as long as the ball remains visible, it fails to recover after the ball is occluded. This is due to a combination of two factors: i) the visible color of the ball at the time when the ball comes out from below the table (red) is different from the visible colors just before the occlusion (blue), and ii) the binder on the table is similar in color to the blue on the ball from the viewpoint of the HSV color model. Thus, the adaptive color model incorporated incorrect color information into its the current color histogram. Then the tracker with the adaptive color model remained in the green color region of the binder on the table. This result shows the critical problem of the adaptive color modeling.

The example shown in Figure 6.7 shows the tracking results when the dynamic color model with CF-identified color dynamic is used in the sequence shown before in Figure 6.4. In this example, a combination of a priori information shown in Table 6.2 and the a posteriori measurements of the hue histogram bins and the positions from $N = 13$ frames, where the target was not occluded, were used to estimate their dynamics. Similar to the previous experiment, a sinusoidal transfer function was selected to model oscillatory values of the hue histogram bins.
(a) Constant color modeling using the first frame.

(b) Constant color modeling and dynamic motion modeling.

(c) Adaptive color modeling using the previous frame and dynamic motion modeling.

Fig. 6.4. Tracking results of constant and adaptive color modeling. All the above methods fail to track a red and blue ball due to color change or occlusion.
### Table 6.2. *A priori* information for the flying ball sequence shown in Figure 6.7

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Bound</td>
<td>$\mathcal{N} = { \eta \in \ell_{\infty}, | \eta |<em>{\ell</em>{\infty}} \leq 10 \text{(histogram)}, 3 \text{(position)} }$.</td>
</tr>
<tr>
<td>Input of System</td>
<td>$u = \delta(t)$.</td>
</tr>
<tr>
<td>Output of System</td>
<td>$y = 13 \text{ number of past values of histogram bin and position.}$</td>
</tr>
<tr>
<td>Parametric Part</td>
<td>$F_p \in \text{span} \left{ \frac{z}{z-1}, \frac{z}{(z-1)^2}, \frac{z^2 - \cos \omega z}{z^2 - 2 \cos \omega z + 1} \right}$ \text{ where } \omega \in {0.1, 0.12, 0.55}.$</td>
</tr>
<tr>
<td>Nonparametric Part</td>
<td>$F_{np} \in BH_{\infty, \rho}(K)$, with $\rho = 0.99$.</td>
</tr>
</tbody>
</table>

These dynamics were then used in conjunction with Kalman filters, leading to the predictions shown in Figures 6.5 and 6.6 in which the CF-based identified dynamics accurately predicts the color histogram and position when the object appears again after occlusion. Note that the large disparity between the measurements and the estimations by the CF approach is observed in 6.5 (b). Because of the sharp change of the histogram bin value, it was difficult to find a feasible solution that interpolates such a large variation. Thus, the sequences of each histogram bin were scaled down by 10 and a large value of the noise bound was used to interpolate the measurements. Thus, scaling up both the measurements and estimations by the CF approach introduced such a large disparity. However, worth mentioning is that the identified dynamics by the CF approach successfully captured the motion characteristic of the hue histogram values.

As shown in Figure 6.7, the proposed approach is able to track the ball beyond the occlusion. This is possible due to the accurate appearance predicted by the learned model as illustrated in Figure 6.5. Notice that there are no measurements while the ball was occluded behind the table, but once they become available again, there is a close match between measurements and predicted values. The difference between the first row and the second row in Figure 6.7 is the use of the dynamic motion model with the CF-identified dynamics. As shown in Frame 19, the combination of the dynamic color model and the dynamic motion produced a better tracking result, predicting the target positions accurately during occlusion.

### 6.4 Discussion

This chapter demonstrates that the dynamic color modeling as well as the dynamic motion modeling plays an important role in identifying and tracking the target
Fig. 6.5. Measurements and CF estimated values of the (a) blue and (b) red bin counts of the hue histogram of the ball shown in Figure 6.7 (a) and (b) as it rolls in front of the camera.

Fig. 6.6. Measurements and CF estimated values of the (a) $x$ and (b) $y$ positions of the ball shown in Figure 6.7 (b) as it rolls in front of the camera.
Fig. 6.7. CF-based dynamic color model. Tracking using dynamic color model alone (top), and both dynamic color model and dynamic motion model (bottom) of a red and blue ball is successful, even in the presence of occlusion. Conjunction with dynamic motion model provides accurate position predictions during occlusion.
object robustly, especially in the presence of occlusion. The identified color and motion dynamics using the CF approach produces accurate predictions when observation is not available such that it prevents the tracking from drifting or tracking failure. Even though adaptive color modeling provides model update strategy incorporating the past measurement, such modeling does not have a predictive mechanism when the measurement is not available. Therefore, dynamic color modeling can be an alternative solution to increasing the robustness to target color variation.

It should be noted that motion and color are not the only dynamic features of moving targets, even though both are important to characterize the target movement and appearance. Color histogram used for color modeling is global information that does not represent local color variations. Thus, it is often difficult to describe an object using color histogram, because the same color histogram can be appeared in different objects. The next chapter will introduce dynamic appearance modeling that incorporates all visual information of moving targets.
Chapter 7

Dynamic Appearance Modeling

7.1 Introduction

Dynamic appearance is one of the most important cues for tracking and identifying moving people. However, direct modeling spatio-temporal variations of such appearance is often a difficult problem due to their high dimensionality and nonlinearities. In this chapter we present a human tracking system that uses a dynamic appearance and motion modeling framework based on the use of robust system dynamics identification and nonlinear dimensionality reduction techniques. The proposed system learns dynamic appearance and motion models from a small set of initial frames and does not require prior knowledge such as gender or type of activity.

Tracking and identifying moving humans in video sequences is a challenging task. This is because the visual appearance of the targets changes dynamically due to their articulated body structure, and due to viewpoint and illumination changes. The performance of tracking algorithms often depends on how well they can estimate the visual information and also how well they can efficiently handle its temporal variation. Several approaches have been introduced in the literature to address these issues.

Color information is widely used for tracking target objects. Perez et al. [53] proposed probabilistic tracking based on color histograms over time. Lim et al. [39] modeled dynamic variations of color histograms and predicted future color information using robust system identification techniques. Different objects with similar colors, however, may introduce ambiguities because local color and shape information is not taken into consideration by these approaches.


More recently, adaptive models incorporating both spatial and temporal variations of the appearance have been proposed. While these methods perform better than the previous approaches, in general, they lack accurate generative models. Jepson et al. [30]
developed the WSL tracker, in which the appearance variation is divided into stable, lost, and wandering components. Ho et al. [27] used linear subspaces obtained by the Gramm-Schmidt process and updated the orthogonal basis over time to handle temporal variations of local appearance information. Han and Davis [25] utilized a mean shift based algorithm for sequential density estimation of each pixel. Black et al. [5] modeled appearance changes using motion information and Li et al. [37] modeled appearance using geometric transform. Zhou et al. [72] incorporated an adaptive appearance model in a particle filter. Lim et al. [42] and Ross et al. [56] updated appearance using an incremental subspace update. All of these approaches search and validate the target appearance based on previous observations. However, they may drift by incorporating the background into the target template and can fail to track in the presence of long term occlusion.

Fig. 7.1. Generated walking sequence after the learning step, $t = 38, 39, 40, \ldots, 48$.

The work presented here is closely related to approaches that attempt to model visual appearance in low dimensional spaces. In the context of human appearance modeling, Lim and Kriegman [41] and Dimitrijevic et al. [17] used a set of prior learned templates which do not characterize individual differences. Elgammal and Lee [20] learned human appearance using LLE [58] and estimated the 3D pose of the target from the appearance using a nonlinear mapping. However, they did not provide a predictive mechanism. Rahimi et al. [55] proposed the use of second order dynamics on appearance manifolds. While successful in many scenarios, this approach suffers from the fact
that a tracker must rely on the assumed dynamics to produce estimates of the future appearances, introducing a potential source of fragility. A mismatch between this model and the actual dynamics will lead to incorrect predictions [11].

This problem is illustrated in Figure 7.1 where appearance templates that are generated using constant acceleration dynamics on the appearance manifold will deteriorate much more rapidly than templates generated using a dynamic model identified by a robust identification technique on the same manifold. Furthermore, it should also be noted that neither of the above approaches identifies the dynamics of the target motion and therefore does not produce accurate predictions of where the target should be sought for in the next frame. This makes trackers particularly fragile to clutter and occlusion as illustrated in Figure 7.2. Here, a Condensation tracker [29], using dynamic appearance templates in conjunction with constant velocity motion dynamics, fails to find the target after the person is occluded by the tree.

In this chapter, we present a robust human tracking system which learns dynamical appearance and motion models with accurate predictive power from a small set of initial frames. The learning module employs a robust system dynamic identification technique based on Caratheodory-Fejer (CF) interpolation [50]. Because CF interpolation finds a model to fit the data instead of forcing an a priori model onto the data, the identified model can accurately predict future appearances and target positions, even in the presence of long term occlusion. As a result, the predicted appearances are very effective as templates for robust tracking.

---

1 This is the well known divergence phenomenon, see for instance [1], page 133.
7.2 The Proposed Approach

The overall tracking system is based on the following ideas:

1. At each time $t$, the target is located in the image. Its current high dimensional appearance of the target in image space is represented by a point on a low dimensional manifold found using a nonlinear mapping. This preserves the spatial and temporal neighborhoods, where the time evolution is governed by piecewise linear dynamics. This idea leads to a separation type principle that allows for separating and independently identifying the appearance and target motion dynamics.

2. This low dimensional point is the output at time $t$, of a linear time invariant (LTI) dynamical system which is identified from a small set of frames using a robust identification procedure.

3. Future outputs of the LTI system are accurately predicted by the dynamic evolution of the system on the manifold.

4. Future high dimensional appearances of the target in the image space are predicted by a nonlinear inverse mapping applied to these predicted outputs.

5. The location of the predicted appearance in the image space can be predicted by the output of a linear time invariant (LTI) system which is identified from measurements of the target positions at the training images using a robust identification procedure.

6. Finally, the predicted appearances at the predicted locations are used as dynamic templates for tracking, target identification, and occlusion detection in conjunction with a Kalman or a Condensation filter.

Each of these ideas is described in detail in the sequel.

7.3 Dynamic Appearance Modeling

The process of learning dynamic appearance models consists of three steps: dimensionality reduction, dynamics identification, and inverse mapping as illustrated in Figure 7.3. The proposed approach learns the dynamics in the manifold space and predicts future appearances based on the identified dynamics.
7.3.1 Nonlinear Dimensionality Reduction for Dynamic Appearance Modeling

Appearance changes due to human motions such as walking or running can often be represented by a small number of latent variables, which can be found by dimensionality reduction. Principal Component Analysis (PCA) is a simple and useful tool for dimensionality reduction. However, it can project faraway high dimensional data points to nearby points in the lower dimensional space if the data do not lie in a linear subspace.

Nonlinear dimensionality reduction methods are designed to preserve local neighborhoods of the points by analyzing the underlying structure of the data. This is a highly desirable property when modeling temporal visual appearance, since we would like to preserve temporal continuity in the low level representation of the data. In this chapter, we utilize the Locally Linear Embedding algorithm for nonlinear dimensionality reduction proposed by [58] to obtain a low dimensional appearance representation from a small video sequence. Other possibilities include Isomap [64], Laplacian Eigenmaps [3], and Hessian LLE [19], and Semidefinite Embedding [69, 68].

7.3.1.1 Locally Linear Embedding(LLE) for Dimension Reduction

Recently, the Locally Linear Embedding(LLE) algorithm has been developed for the problem of nonlinear dimensionality reduction[60]. Dimension reduction by LLE preserves the neighborhood of the data points both in the high and low dimensional space by analyzing the underlying structure of the manifold.

The procedure of LLE is given as follows:
Fig. 7.4. Steps of Locally Linear Embedding [58]. (1) Find the neighbors of each data point, $X_i$. (2) Compute the weights $W_{ij}$ that best reconstruct $X_i$ from its neighbor. (3) Compute the vectors $Y_i$ best reconstructed by $W_{ij}$. 
Fig. 7.5. Example of Locally Linear Embedding algorithm [58]. A. three dimensional data. B. sampled data from a two dimensional manifold of A. C. two dimensional representation by LLE.

1. Compute nearest neighbors \( z_j \) of each data point \( z_i \) based on the Euclidean distance \( \|z_i - z_j\|^2 \).

2. Compute the weights \( w_{ij} \) that best reconstruct each data point \( z_i \) from its neighbors, minimizing the cost function in Eq. (7.1).

\[
\xi(W) = \sum_i \|z_i - \sum_j w_{ij}z_j\|^2
\] \hspace{1cm} (7.1)

subject to \( \sum_j w_{ij} = 1 \)

3. Compute the vector \( y_i \) best reconstructed by the weights \( w_{ij} \), minimizing the quadratic form in Eq. (7.2)

\[
\Phi(Y) = \sum_i \|y_i - \sum_j w_{ij}y_j\|^2
\] \hspace{1cm} (7.2)

Note that the minimization in Eq. (7.2) can be efficiently achieved by calculating the eigenvectors of \( M = (I - W)(I - W)^T \) with the smallest eigenvalues.

Figure 7.4 shows the graphical procedure of how the three dimensional data points are embedded on in two-dimensional space. Figure 7.5 shows that three-dimensional data
Fig. 7.6. Dynamic appearances of a walking motion for 37 frames, $t = 0, 6, 12, \ldots, 36$.

Fig. 7.7. Low dimensional representation of the walking sequence by LLE.
Fig. 7.8. Three dimensional representation of the walking sequence by LLE. Periodicity of walking cycle is observed in the manifold space.
are mapped on the two dimensional space, and at the same time, preserving both local and global geometry.

Figure 7.8 illustrates how a set of collected appearances of a walking motion as the one shown in Figure 7.6 can be represented in a three dimensional space where the temporal ordering of the original data is preserved. Figure 7.7 shows the temporal evolution of the coordinates of the points on the LLE manifold for the first 37 frames of the walking sequence shown in Figure 7.6. In the next step, we will find a dynamical model to predict future values of these coordinates, based on their past measurements.

7.3.2 Robust Dynamics Identification on the LLE manifold

The Caratheodory-Fejer approach is utilized to identify accurately the dynamics on the LLE manifold, producing future estimates.

Assume that the low dimensional representation of the target appearance at time \( t \), \( y_t \) is related to the previous appearance representations by an ARMAX model of the form:

\[
y_t = \sum_{i=1}^{m} g_i y_{t-i} + \sum_{i=0}^{m} h_i u_{t-i} \tag{7.3}
\]

where \( g_i, h_i \) are fixed coefficients and \( u_t \) denotes a stochastic input. This can always be assumed without loss of generality, since given \( m \) measurements of \( y \) and \( u \), there always exist a linear time invariant system such that (7.3) is satisfied ([50], Chapter 10). This system, in turn, can be represented using a state space description of the form:

\[
x_{t+1} = Ax_t + Bu_t \\
y_t = Cx_t + Du_t \tag{7.4}
\]

where

\[
A = \begin{bmatrix} 0 & 0 & \cdots & g_m \\ 1 & 0 & \cdots & g_{m-1} \\ 0 & 1 & \cdots & g_{m-2} \\ \vdots & \vdots & \cdots & g_1 \end{bmatrix}, \quad B = \begin{bmatrix} h_m \\ h_{m-1} \\ h_{m-2} \\ \vdots \\ h_1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad D = h_0 \tag{7.5}
\]

Note that this (minimal) realization is unique as far as a coordinate transformation. In the sequel we will use the short-hand \( \mathcal{F} \) to denote the operator \( (A, B, C, D) \) that maps the exogenous input \( u \) to the measured feature \( y \), and use robust identification techniques to extract these matrices from the experimental data.

Figure 7.9 illustrates the interpolation and prediction of the three coefficients using the obtained the operator \( A, B, C, D \) by CF approach for the LLE manifold shown in Figure 7.7.
7.3.3 Recovering Dynamic Appearance from the Manifold Space

An inverse mapping from the manifold to the image space - estimating each pixel value at time $t$ from the low dimensional appearance representation - can be learned from the training data by employing using Radial Basis Function (RBF) [54]. Once the RBF network learns the nonlinear mapping between pairs of manifold points and appearance pixels, intermediate and future image appearances can be generated from points on the manifold corresponding to the current/future state.

7.3.3.1 Radial Basis Function

Radial basis function (RBF) networks provides nonlinear mapping of two continuous functions. The temporal sequence of high dimensional appearance images and its low dimensional LLE data are continuous over time. Thus, RBF can be used for this purpose.

Given $N$ different points $\mathbf{y}_i \in \mathbb{R}^n$, $i = 0, \cdots, N - 1$ in the manifold and $N$ real number $z_i \in \mathbb{R}$, $i = 1, \cdots, N$ in the image (image pixel), find a function $f$ from $\mathbb{R}^n$ to $\mathbb{R}$ satisfying the interpolation conditions:

$$ f(\mathbf{y}_i) = z_i \quad (7.6) $$

The radial basis function (RBF) has the following form:

$$ f(\mathbf{y}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{y} - \mathbf{y}_i\|) + p(\mathbf{y}) \quad (7.7) $$
where $\phi$ is the radial basis function, $w_i$ are real coefficients, $\| \cdot \|$ is the Euclidean norm on $R^n$, and $p(y)$ is the linear polynomial. In this context the points $y_i$ are referred to as the center of the RBF.

Popular choices for the basis function include the thin-plate spline $\phi(r) = r^2 \log(r)$, the Gaussian $\phi(r) = e^{-cr^2}$, the multiquadric $\phi(r) = \sqrt{r^2 + c^2}$, the triharmonic $\phi(r) = r^3$, biharmonic $\phi(r) = r^4$.

RBF are popular for interpolating scattered data as as the associated system of linear equations is guaranteed to be invertible. If the polynomial in Equation (7.7) is of degree $m$ then the side conditions imposed on the coefficients are

$$\sum_{i=1}^{N} w_i q(y_i) = 0, \text{for all polynomials of degree at most } m \quad (7.8)$$

These conditions, in addition to the interpolation conditions of Equation (7.6), lead to a linear system to solve for the coefficients that specify the RBF.

Let $\{p_1, p_2, \ldots, p_l\}$ be a basis for polynomials of degree at most $m$, $C = [c_1 \ c_2 \ \ldots \ c_l]^T$ be the coefficients that give $p$ in terms of this basis, $W = [w_1 \ w_2 \ \cdots \ w_N]^T$, and $Z = [z_1 \ z_2 \ \cdots \ z_N]^T$. Then Equations (7.6) and (7.7) can be written in matrix form as

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} W \\ C \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (7.9)$$

where

$$A_{ij} = \phi(\|y_i - y_j\|), \quad i, j = 1, \ldots, N,$$

$$P_{ik} = p_k(y_i), \quad i = 1, \ldots, N, \quad k = 1, \ldots, l \quad (7.10)$$

If it is assumed $p(y) = c_0 + c_1 y_1 + \ldots + c_n y_n$ where $y_n$ denotes $n$th row of the vector $y$, then $P_i$ is the matrix with $i$th row $[1 \ y_{i1}^1 \ y_{i2}^1 \ \cdots \ y_{in}^1]$, and $C = [c_0 \ c_1 \ \ldots \ c_n]^T$.

Solving the linear system in Equation (7.3.3.1) determines $W$ and $C$, and hence $f(y)$. However, the same input data may be able to be approximated to the desired accuracy using significantly fewer centers as shown in Figure 7.10.

Reducing the number of RBF centers results in smaller memory requirements and faster evaluation times, without a loss in accuracy. Such centers can be obtained using $k$-means clustering or EM algorithm.

Figure 7.11 shows appearances for the walking sequence of Figure 7.6 recovered by the interpolated and predicted states on the manifold space. It can be seen there, that the learned RBF network successfully interpolates the unseen appearances.
Fig. 7.10. Center reduction in RBF interpolation.

Fig. 7.11. Generated walking sequence for 76 frames by RBF, $t = 0, 3, 6, \ldots, 75$. 
7.4 Dynamic Motion Modeling

Given a set of measurements of the position of the target in image space, one can use the same techniques described in Chapter 4 to identify the motion dynamics of the target. This dynamics, in turn, can be used to predict the location of the target in future frames accurately, even in the presence of occlusion.

7.5 Human Tracking using Dynamic Appearance and Motion Modeling

Unlike other approaches, our approach utilizes the motion dynamics of the target in the image space to predict the location of the target and the dynamics of the appearance nonlinear projections on the manifold space to generate the dynamic appearance templates that can be sought at the predicted locations. The advantage of our approach is that the dynamics, which are identified on line from a small number of training frames, provide accurate location and visual information of the target over time.

Once the target location and its appearance are predicted, template-based tracking techniques such as Condensation [29] or affine region tracking [24] can be used to follow the target. In this chapter we used a Condensation filter, in which the likelihood of the target location and appearance was measured by evaluating the pixel-wise color similarity between the tracked region and the predicted template.

7.5.1 Occlusion Handling

Two cases of occlusion are taken into account in this chapter; occlusion by background objects and occlusion by other moving people. In the first case, occlusion is detected by comparing the likelihood against a threshold. The appearance and position of the template are predicted based on the identified dynamics. The predictions of the position and appearance are conducted until the target person is detected again using color similarity. As mentioned before, the success of the proposed approach hinges on the close fit of the system dynamics to the data provided by robust identification [11].

In the second case, the occlusion is detected by the collision of multiple templates of moving people. If the occlusion is detected, the probability that the template is likely to be the actual appearance within the region overlapping the two templates decides either the front or rear layer. The likelihood is measured by color similarity within the overlapped region. If the template is assigned to the front layer, the tracker for that person continues to track. On the other hand, if the template is assigned to the rear layer, the tracker handles the occlusion as in the first case, e.g., the person in the front layer is treated as a background object with respect to the person in the rear layer.
7.6 Illustrative Examples

The proposed approach was tested on several outdoor scenes captured with stationary and moving color cameras. In every case, the dynamic appearance of the target was learned from frames from the beginning of the sequences, where it was assumed that the target was not occluded and that the camera was stationary while learning.

7.6.1 Implementation Details

A sequence of $T$ frames of the target is collected by tracking the target while the target is initially unoccluded and the camera is stationary. The foreground regions were extracted by background subtraction and morphological operations were applied to the binary foreground to remove noise and fill in regions. The target appearance was then obtained using this binary mask. The mean shift algorithm was then applied to capture the appearance into a rectangular window, which was normalized to a given size. To reduce the dimensionality of the appearance vector, YCbCr 4:2:0 color compression was applied to reduce the image data by a half while preserving almost all color information. After applying color compression, the appearance vector $z_t$ at time $t$ was obtained by rasterizing the preprocessed window.

The corresponding three dimensional representation $y_t$ was then obtained by applying the LLE algorithm using a set of the appearances $z_t$ for $T$ frames. Next, the dynamic evolution of the low dimensional representation was identified using CF interpolation with a priori information shown in Table 7.1.

Finally, a RBF network was used to reconstruct the high dimensional representation. Each pixel $z_k^t$ was reconstructed by nonlinear mapping from the three dimensional manifold space. Once dynamics appearance is learned using the above methods, the predicted appearance can be generated at every frame using the learned model. The predicted appearance templates were combined with the particle filter, in which the template was used for the observation model to calculate the observation likelihood of the samples in Equation 2.3.3.

In the following experiments, the backprojection method, the mean shift algorithm, and the particle filter were implemented using C and C++ languages. The LLE algorithm, the CF approach, and the RBF network were implemented using Matlab. Even though two different platforms were used for implementation, all the processes were conducted online by communicating each other. In addition, the implemented tracking system is real time except the learning process of dynamic appearance.
Parameters & Conditions

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Bound</td>
<td>$\mathcal{N} = { \eta \in \ell_\infty, | \eta |<em>{\ell</em>\infty} \leq 2 \text{(LLE data)}, 10 \text{(position)} }$.</td>
</tr>
<tr>
<td>Input of System</td>
<td>$u = \delta(t)$.</td>
</tr>
<tr>
<td>Output of System</td>
<td>$y = 38$ number of past LLE data.</td>
</tr>
<tr>
<td>Parametric Part</td>
<td>$F_p \in \text{span} \left{ \frac{z}{z-1}, \frac{z}{(z-1)^2}, \frac{z^{2} - \cos \omega z}{z^{2} - 2 \cos \omega z + 1}, \frac{\sin \omega z}{z^{2} - 2 \cos \omega z + 1} \right}$ where $\omega \in { \omega_0/2, \omega_0, 2\omega_0, 4\omega_0 }$. $\omega_0$ is found by FFT.</td>
</tr>
<tr>
<td>Nonparametric Part</td>
<td>$F_{np} \in \mathcal{BH}_{\infty, \rho}(K)$, with $\rho = 0.99$.</td>
</tr>
</tbody>
</table>

Table 7.1. *A priori* information for the walking sequences shown in Figures 7.12, 7.14, and 7.15.

Fig. 7.12. Handling occlusion caused by the background object, $t = 60, 75, 90,$ and 105. The white box in the second row is the predicted appearance and the black box is the probability map based on the predicted appearance.
7.6.2 Tracking Results

Three experimental results are reported here: tracking moving people in the presence of occlusion by a scene object, occlusion by other moving people, and tracking using a moving camera.

Figure 7.12 shows again the video sequence introduced in Figure 7.2, using the proposed approach. What can be seen in this case is that the appearances and their location are accurately predicted even during long term occlusion by the tree. These predictions allow the tracker to “hallucinate” the target while it is occluded and to recover it after the occlusion is over. It also should be noted that the template remains uncorrupted by the occlusion, unlike all adaptive appearance techniques, preventing tracking drift.

![Layer selection by the likelihood to the observed appearance in the overlapped region by two templates. The higher likelihood represents the front layer and vice versa.](image)

Figure 7.13. Layer selection by the likelihood to the observed appearance in the overlapped region by two templates. The higher likelihood represents the front layer and vice versa.

Figure 7.14 shows the use of the predicted appearances for layer selection among people. As illustrated in the Figure 7.13, the likelihood of person 0 is lower than the ones for persons 1 and 2, while they walk across each other, respectively. Thus, person 0 is assigned to the rear layer and persons 1 and 2 are assigned to the front layer while occluded. Since the dynamic appearance is individually learned for each person, the trackers can track and identify each target person even during occlusion and in the presence of clutter.

Finally, Figure 7.15 shows that once the dynamic appearance is learned, the tracker can track the target person even though the camera moves.
Fig. 7.14. Handling occlusion caused by other moving people. The layer is automatically selected based on the likelihood of each template to the observed appearance with the overlapped regions.

Fig. 7.15. Moving camera after learning appearance. The tracker robustly tracks the target person even though the camera moves.
7.7 Discussion

In this chapter, we formulated dynamic appearance and motion modeling for human tracking as a three step process: dimensionality reduction, dynamics identification, and inverse mapping. The proposed approach predicts the location of the target and predicts its future appearance based on accurately identified dynamics learned from a small set of initial frames. The predicted appearances can then be used as dynamic templates for tracking and identifying moving people even in the presence of occlusion and clutter. Future work is needed to incorporate the worst identification error bounds provided by CF to perform model (in)validation and decide when the current appearance model is no longer valid (e.g. if the target changes activity from walking to running) and to decide when a new model should be obtained.
In the past few years visual tracking techniques have proved to be a viable option for a large number of applications, ranging from surveillance and human computer interface. A number of techniques have been developed for such purposes. Arguably, at this point one of the critical factors limiting widespread use of these techniques is the potential fragility of the resulting systems, which is a lack of relevant dynamics for modeling dynamic elements representing the target object.

In this thesis we have shown that this fragility can be addressed by relevant dynamics identified using the robust system identification. The subjects of dynamics for modeling the target includes motion, color, and appearance, which can be extended to any dynamic elements that reside internally or externally in the visual information of the target.

As illustrated with many examples, the robustness of visual tracking has been accomplished by accurately predicting the future estimate of the target, based on the identified dynamics. Specifically for appearance modeling, the dynamics identification is combined with nonlinear dimensionality reduction techniques to handle high dimensionality and nonlinearity of the target appearance efficiently, thus accurately producing the future appearances contained all visual information.

Future work includes incorporating the worst identification error bounds provided by the CF approach to perform model (in)validation and deciding when the current appearance model is no longer valid (e.g. if the target changes activity from walking to running) and to decide when a new model should be obtained. A faster method to solve the LMI equation in the CF approach will be sought for real time applications in further work.

8.1 Related Publications

- A Caratheodory-Fejer Approach to Robust Multiframe Tracking, O. Camps, H. Lim, C. Mazzaro, and M. Sznaier, ICCV’03
- A Caratheodory-Fejer Approach to Dynamic Appearance Modeling, H. Lim, O. I. Camps, and M. Sznaier, CVPR’05
- Dynamic Appearance Modeling for Human Tracking, H. Lim, V. I. Morariu, O. I. Camps, and M. Sznaier, CVPR’06
Appendix

Notation

\( \mathbf{x} \) real–valued (unless otherwise stated) column vector.

\( x_k \) \( k^{th} \) element of a vector \( \mathbf{x} \).

\( \| \mathbf{x} \|_p \) \( p \)-norm of a vector: \( \| \mathbf{x} \|_p = \left( \sum_{k=1}^{m} |x_k|^p \right)^{\frac{1}{p}} \), \( p \in [1, \infty) \), \( \| \mathbf{x} \|_\infty = \max_{k=1,...,m} |x_k| \).

\( \mathbf{A}^T \) conjugate transpose of matrix \( \mathbf{A} \).

\( A_{i,j} \) \((i,j)\) element of \( \mathbf{A} \).

\( \mathbf{x}_i \) \( i^{th} \) row of \( \mathbf{A} \).

\( \sigma(\mathbf{A}) \) maximum singular value of the matrix \( \mathbf{A} \).

\( \mathbf{A} > 0 \) \( \mathbf{A} = \mathbf{A}^T \) is positive definite, i.e. \( \mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \ \forall \mathbf{x} \in \mathbb{C}^n, \mathbf{x} \neq 0 \).

\( \mathcal{B}\mathcal{X}(\gamma) \) open \( \gamma \)-ball in a normed space \( \mathcal{X}: \mathcal{B}\mathcal{X}(\gamma) = \{ x \in \mathcal{X}: \|x\|_{\mathcal{X}} < \gamma \} \).

\( \overline{\mathcal{B}\mathcal{X}(\gamma)} \) closure of \( \mathcal{B}\mathcal{X}(\gamma) \).

\( \mathcal{B}\mathcal{X} \) (\( \overline{\mathcal{B}\mathcal{X}} \)) open (closed) unit ball in \( \mathcal{X} \).

\( (\mathcal{X}, m) \) metric space of elements in \( \mathcal{X} \) equipped with the metric \( m(x_1, x_2) \).

\( d(\mathcal{A}) \) diameter of \( \mathcal{A} \subseteq \mathcal{X} \): \( d(\mathcal{A}) = \sup_{x,a \in \mathcal{A}} m(x,a) \).

\( \ell^m \) extended Banach space of vector valued real sequences equipped with the norm:

\[
\| x \|_p = \left( \sum_{i=0}^{\infty} \| x_i \|_p^p \right)^{\frac{1}{p}} \quad p \in [1, \infty),
\]

\( \| x \|_\infty = \sup_{x} \| x_i \|_\infty \).

\( \mathcal{L}_\infty \) Lebesgue space of complex–valued matrix functions essentially bounded on the unit circle, equipped with the norm:

\[
\| G \|_\infty = \text{ess sup}_{|z|=1} \sigma(G(z)).
\]

\( \mathcal{H}_\infty \) subspace of functions in \( \mathcal{L}_\infty \) with bounded analytic continuation inside the unit disk, equipped with the norm:

\[
\| G \|_\infty = \text{ess sup}_{|z|<1} \sigma(G(z)).
\]

\( \mathcal{H}_\infty,\rho \) space of transfer matrices analytic in \( |z| \leq \rho \), equipped with the norm \( \| G \|_{\infty,\rho} = \sup_{|z|<\rho} \sigma(G(z)) \).

\( X(z) \) \( Z \)-transform of a right–sided real sequence \( \{x\} : X(z) = \sum_{i=0}^{\infty} x_i z^i \).
References


Vita

Hwasup Lim was born in Korea on June 3, 1974. In 2000 he received the B.S. degree in Electronics, Communication, and Radio Engineering, from the Hanyang University in Seoul. In 2001 he enrolled in the M.S. degree in Electrical Engineering at the Pennsylvania State University. Since 2003 Fall semester he has been worked in the Robust Vision Systems Lab as a research assistant under the direction of Dr. Octavia I. Camps. His main research interest is dynamic motion and appearance modeling using the identified dynamics for robust object tracking in the single and multiple camera environments. Other areas of interest includes real time distributed visual tracking system, 3D pose estimation using silhouettes, and multiple target tracking with multiple cameras.