Dynamic Object Tracking by Partial Shape Matching for Video Surveillance Applications

Mustafa Husain
DYNAMIC OBJECT TRACKING BY PARTIAL SHAPE
MATCHING FOR VIDEO SURVEILLANCE APPLICATIONS

By

Mustafa Husain

A Thesis submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTERS OF SCIENCE

In
Electrical Engineering

Approved by:

Professor

Eli Saber
(Prof. Eli Saber – Advisor)

Sohail A. Dianat
(Prof. Sohail A. Dianat – Committee Member)

Daniel B. Phillips
(Prof. Daniel B. Phillips – Committee Member)

Vincent Amuso
(Prof. Vincent Amuso – Department Head)

DEPARTMENT OF ELECTRICAL ENGINEERING

COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

JULY 2006
THESIS RELEASE PERMISSION

DEPARTMENT OF ELECTRICAL ENGINEERING

COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

Title of Thesis:

DYNAMIC OBJECT TRACKING BY PARTIAL SHAPE MATCHING FOR VIDEO SURVEILLANCE APPLICATIONS

I, Mustafa Husain, hereby grant permission to Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature __ Mustafa Husain ____________________________

Date
ACKNOWLEDGEMENTS

First I must thank Dr. Eli Saber. Not only because he taught me everything that I know in digital video processing but more importantly because he taught me how to research. I would like to thank Dr. Daniel B. Phillips, Dr. Sohail A. Dianat and Dr. Vladmir Misic for taking time out of their busy schedule and going through my thesis, providing their valuable insights and constructive criticism.

I would also like to thank the Center for Electronic Imaging Systems - a NYSTAR designated Center for Advanced Technology, PLE Communications, LLC and the Electrical Engineering Department of the Rochester Institute of Technology for supporting this work.
ABSTRACT

In this thesis, an algorithm for object tracking through frames of video using a fast partial shape matching technique is proposed. The tracking is divided into two modules: 1) moving object extraction followed by color/edge segmentation, and 2) tracking through frames using partial shape matching. The major challenges of object tracking, such as occlusions, splitting of one object and appearance and disappearance of objects, are effectively resolved. The proposed algorithm is tested on several synthetic and real life video sequences and is shown to be very effective in identifying and tracking moving objects independent of translations, rotations, scale variations and occlusions.

The novelty of the proposed algorithm lies in its ability to independently track full and partial objects undergoing split, merge and occlusion scenarios independent of their location and scale in the scene. The technique assumes that: 1) the video frames are captured at 30 frames per second in order for the object(s) motion (translation, rotation, isometric scale variations) to be well modeled by an affine transformation, 2) the object(s) being tracked are larger than a certain number of pixels to allow for comprehensive shape modeling, and 3) the video camera is kept stationary.
# Table of Contents

Acknowledgements ........................................................................................................ 1  
Abstract ....................................................................................................................... 2  
Table of Contents ......................................................................................................... 3  
List of Figures .............................................................................................................. 4  
List of Tables .............................................................................................................. 5  

**Chapter 1**............................................................................................................... 6  
1. Introduction ........................................................................................................... 6  

**Chapter 2**............................................................................................................... 10  
2. Background ........................................................................................................... 10  

**Chapter 3**............................................................................................................... 16  
3. Object Tracking Algorithm .................................................................................. 16  
   3.1 Low level Frame Analysis ............................................................................ 17  
   3.2 Tracking Algorithm ...................................................................................... 20  
      3.2.1 Case I (object continuity) ................................................................. 24  
      3.2.2 Case II (Object Merge/Exit) ............................................................ 24  
      3.2.3 Case III (Object Split/ Entrance) .................................................... 25  
      3.2.4 Computational Aspects of the Tracking Algorithm ....................... 25  

**Chapter 4**............................................................................................................... 26  
4. Experimental Results and Discussions ................................................................. 26  

**Chapter 5**............................................................................................................... 39  
5. Conclusions and Future Work ............................................................................. 39  

Appendix I – RGB to HSV Transformation ............................................................. 42  
Appendix II – Hausdorff Distance for Comparing Images ..................................... 43  
References ................................................................................................................ 44
List of Figures

Figure 1: Block diagram of the utilized partial shape matching procedure. ......................... 11
Figure 2: Definition of high curvature points. ................................................................. 12
Figure 3: Block diagram of the proposed hybrid tracking algorithm ......................... 17
Figure 4: Flow chart for low level frame analysis. ....................................................... 18
Figure 5: Flowchart of the Tracking Algorithm................................................................. 22
Figure 6: Visual illustration for matching algorithm: (a) Typical moving object, (b)
Segmentation map, (c) Adjacency matrix, (d)-(g) Combination of successfully
matched elementary regions and (h) the largest match History. ......................... 23
Figure 7: Tracking results for “Lab simulated” sequence (a)-(f) represent frames 18, 59,
79, 92, 109, and 152 (Red outline around the object indicates result of tracking
algorithm). ......................................................................................................................... 27
Figure 8: Object trajectory for Lab simulated Scene. The horizontal and vertical axes of
the graphs represent the width and the height of the frame respectively. ........ 28
Figure 9: Tracking results for “RIT Parking lot” sequence (a)-(f) represent frames 2, 11,
25, 37, 44, and 56 (Pink outline around the object indicates result of tracking
algorithm). ......................................................................................................................... 29
Figure 10: Object trajectory for RIT parking Lot Sequence............................................. 30
Figure 11: Tracking results for “toy cars” sequence (a)-(f) represent frames 8, 20, 26, 30,
33, and 38 (Yellow and pink outline around the objects indicates result of
tracking algorithm). ......................................................................................................... 31
Figure 12: Object trajectory for “Toy Car” Sequence. ..................................................... 32
Figure 13: Tracking results for “RIT junction” sequence (a)-(b) frames 25, 26 (c)-(d)
frames 29, 30 (e)frame 32 (f)-(h) frames 34, 35,37 and (i) frame 38(Yellow and
pink outline around the objects indicates result of tracking algorithm). ........ 33
Figure 14: Object trajectories (frames 20 to 38) for RIT Junction Sequence .......... 34
Figure 15: Tracking results for “Parking Lot” sequence (a)-(i) represent frames 4, 24, 36,
48, 62, 70, 74, 80, and 92 ......................................................................................... 35
Figure 16: Trajectories of the objects for parking Lot Sequence................................. 36
Figure 17: (a)-(d) Object regions discovered in the frames as labeled above .......... 37
Figure 18: Object Tracking and Recognition using Multiple Multimodal Sensors........ 40
Figure 19: Template & image superimposed for measuring similarity metric............. 43

List of Tables

Table 1: Similarity Metric for shape: Objects of frame 68 (Templates) compared to

Objects of frame 72, 74 & 80 undergoing merging........................................... 37

Table 2: Computation cost for the proposed tracking scheme................................ 38
Chapter 1

1. Introduction

Object recognition and tracking is of vital importance to image understanding and computer vision. It often constitutes the backbone required to build effective military, security, and commercial surveillance systems. Surveillance systems based on imaging technology capture a tremendous volume of raw data that must be quickly evaluated in tactical applications. Analysis by human observers is time consuming, prone to errors/oversights, and expensive, if not infeasible, given the continuously rising volume
of data. Applications stemming from effective recognition and tracking techniques include homeland security (airports, border control, sports stadiums, city monitoring), military operations (tracking of enemy tanks, vehicles, digital battlefields), vehicle/robot navigation, and patient/elderly care to name just a few.

Although identification and tracking of objects comes naturally to human observers due to our sophisticated shape matching capabilities under a variety of transformations, computer-based object tracking has proven to be a very difficult task in practical situations. Current shape matching techniques cannot effectively match 2-D images of real life objects with 3-D object templates invariant to translations, rotations, scale variations, reflections, perspective projections, partial occlusions, and articulations.

Several methods have been proposed in the object tracking literature. A layered motion based video representation was proposed in [1]. Motion segmentation methods can be classified as dominant [2] and multiple [3]. The dominant motion in a scene is that motion component that can be ascribed to most of the picture material in an image due to camera behavior and apparent object behavior. Thus dominant motion estimation captures the motion of the dominant background (scene) whereas multiple motion estimation captures the scene and object motions. However, object extraction and tracking by motion uniformity also fails to yield semantic objects because many objects undergo deformable or articulated motion. Methods for integration of color segmentation, edge detection, and motion segmentation have been proposed to partially alleviate these problems [4-7]. Attempts were made to integrate color and edge segmentation [4-6] and color and motion segmentation [7]. These algorithms generally offer reasonable improvements, but fall short of providing meaningful segmentations that are capable of
clearly illustrating semantic objects instead of low-level uniform regions. Automatic initialization for semantic object tracking is possible using change detection under restrictive assumptions (static camera, no illumination changes), or global camera motion that can be well represented by parametric models [8]. When these assumptions are not met, semi-automatic (interactive) methods [9-12] have been developed to segment semantic objects from video, where the contour of the object of interest is interactively marked in some key frames (known as manual initialization). Appearance-based template matching was recently proposed for tracking objects in video frames [13]. The Kernel-based tracking technique is a new method proposed for tracking articulated objects [14]. In this approach the histogram-based target feature is regularized using a special kernel and a Bhattacharyya–based similarity metric [15]. This new approach can handle partial occlusions and target scale variations. Kalman-based prediction and tracking methods have also been investigated in the automation, control and communication literature. Simultaneous object recognition and tracking has only recently started to be considered. Existing work is either specific to face recognition [16], or uses only color features [11]. Multi-camera human motion tracking system, which uses a 3-D stick model of a human subject to track its motion by integrating information from several 2-D views have been proposed in [17, 18] with performance measures analyzed in [19]. In general, the above methods are not fully automatic and cannot track real world objects independent of viewpoint, orientation, and occlusions, rendering them somewhat impractical for large sets of imagery.

In this thesis, a partial shape matching based algorithm for tracking moving objects through frames of video is proposed. The algorithm is divided into two major steps: 1)
low level frame analysis and 2) object tracking. The low level frame analysis is designed to provide an effective segmentation map of the regions that exhibit motion changes through frames of video as identified by background subtraction. This helps to significantly reduce the computational burden of the segmentation process. The object tracking step utilizes the segmentation map regions that make up the object(s) in frame \( t \) as template(s) for matching to frame \( t+1 \) and so on using a partial shape matching algorithm [26]. The partial shape matching algorithm is capable of handling full and partial shape matches invariant to translations, rotations, scale variations, reflections, and occlusions. Thus, the tracker is capable of matching moving objects undergoing various occlusions. The novelty of the proposed algorithm lies in its ability to independently track full and partial objects undergoing split, merge and occlusion scenarios independent of their location and scale in the scene. The technique assumes that: 1) the video frames are captured at 30 frames per second in order for the object(s) motion (translation, rotation, isometric scale variations) to be well modeled by an affine transformation, 2) the object(s) being tracked are larger than a certain number of pixels to allow for comprehensive shape modeling, and 3) the video camera is kept stationary.

The remainder of the thesis is organized as follows: Chapter 2 reviews the low level image analysis and the sub-matrix method for shape matching between two 2-D views using distance matrices. Chapter 3 introduces the partial shape matching based tracking algorithm. Experimental results and discussions are presented in Chapter 4. Conclusions are drawn in Chapter 5.
Chapter 2

2. Background

2.1. Low level Image Analysis

In general, the first step for object tracking in a video sequence is to extract Moving Video Objects (MVOs). MVOs are typically the areas that are changing or have changed in the scene and are generally detected by observing the difference between the current and previous frame using background subtraction methods [24]. This approach suppresses shadows and hence provides a robust framework for the identification of
object shape. Once the MVOs are extracted, they are further divided into “homogeneous” color regions by utilizing a Gibbs Random Field (GRF) based segmentation algorithm [25]. Segmentation based on color and edge information provides “meaningful” elementary regions (ER), which are defined as regions with distinct colors whose boundaries coincide with connected spatial edges. Each elementary region and all valid combinations of neighboring regions in each frame are utilized to establish inter-frame partial or full matches to moving object(s).

2.2. Sub-matrix partial shape matching

The partial shape matching algorithm [26] has been utilized effectively to identify objects of similar shape in still images independent of translations, rotations, scale variations, reflections and occlusions (see Fig. 1 for a block diagram). It consists of two major steps:

![Diagram](image.png)

Figure 1: Block diagram of the utilized partial shape matching procedure.
1) **Feature extraction:** In this step, the information required to perform partial shape matching is provided by the high curvature boundary points of the example template and each potential object region. To this effect, the boundary of object regions and template contours are fitted by B-Splines yielding smooth representations with noise suppression. This provides better accuracy in the extraction of feature point’s i.e. high curvature points. The high curvature points are found by utilizing a two phase procedure [27]. In phase 1, candidate boundary points \( C \) are selected by inscribing a variable triangle \((C^-, C, C^+)\) constrained by a set of rules, as shown in Fig. 2 and Equation 1:

\[
\begin{align*}
|C - C^+|^2 & \leq d_{\text{max}}^2 \\
|C - C^-|^2 & \leq d_{\text{max}}^2 \\
\alpha & \leq \alpha_{\text{max}},
\end{align*}
\]

where \( d_{\text{min}} \) sets the resolution of the detector, \( d_{\text{max}} \) avoids false sharp triangles formed by distant points and \( \alpha_{\text{max}} \) is the angle limit that determines the minimum sharpness accepted as high curvature. The triangle with the smallest angle \( \alpha(C) \) is selected. The value \( \pi - \alpha(C) \) is assigned to the candidate point \( C \) as its *sharpness* value. This is followed in phase 2 by testing the candidate point \( C \) for sharpness non-maxima suppression. The
point is discarded if it has a sharper valid neighbor \( C_v \) (valid if the distance between them is less than \( d_{max} \)).

2) **Similarity matching:** Let \((X_i,Y_j), i = 1, 2, \ldots, n\) be contour feature points of a potential object region or object template contour. The distance matrix for the contour is defined as:

\[
D = \begin{bmatrix}
\begin{array}{cccc}
d_{11} & d_{12} & \cdots & d_{1n} \\
d_{21} & d_{22} & \cdots & d_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
d_{m1} & d_{m2} & \cdots & d_{mn}
\end{array}
\end{bmatrix}
\]

(2)

where \(d_{ji} = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \) \( i, j = 1, 2, \ldots, n \) denotes the distance between feature points \( k \) and \( l \) along the contour. The distance matrix is symmetric and has the following properties: (1) it is invariant to translation and rotation, since it only depends on distances between feature points; (2) it is invariant to isometric scale variation of a contour, i.e. zoom or contraction, since that corresponds to scaling all distances by a constant factor; (3) reflection of a contour is equivalent to inverting the initial ordering of feature points, and (4) if two contours partially match, their distance matrices have *matching* sub-matrices. Thus the problem of determining correspondences between feature points of a region contour and object template is reduced to matching sub-matrices extracted from their corresponding distance matrices. In effect, let \((X_j^R, Y_j^R), j = 1, 2, \ldots, m\) and \((X_i^T, Y_i^T), i = 1, 2, \ldots, n\) denote the feature points for the candidate image region \((R)\) and example template \((T)\) respectively, where the corresponding distance matrices for both regions \(D_{(m \times m)}^R\) and \(D_{(n \times n)}^T\) are computed as shown in (2). To establish point correspondences, compare the \(k \times k\) sub-matrices [26] (typically \(k \geq 4\) for an affine match) found on the
diagonal in each of the distance matrices. A partial match is selected by minimizing the variance for each pair of start points \((p,q)\) on the contours of the image region and the example template as shown in Equation 3.

\[
S_{pq} = \text{Variance} \left\{ r_{i,j}, \quad i, j = 1,2,\ldots, k \right\}
\]

(3)

Where \(r_{i,j}\) denote the ratios of the corresponding elements of two candidate \(k \times k\) submatrices given by:

\[
R = \begin{bmatrix}
\frac{d'_{11}}{d^{t+1}_{11}} & \frac{d'_{12}}{d^{t+1}_{12}} & \cdots & \frac{d'_{1k}}{d^{t+1}_{1k}} \\
\frac{d'_{21}}{d^{t+1}_{21}} & \frac{d'_{22}}{d^{t+1}_{22}} & \cdots & \frac{d'_{2k}}{d^{t+1}_{2k}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{d'_{k1}}{d^{t+1}_{k1}} & \frac{d'_{k2}}{d^{t+1}_{k2}} & \cdots & \frac{d'_{kk}}{d^{t+1}_{kk}}
\end{bmatrix}
\]

(4)

Hence, correspondences between region contour points \((p,\ldots,p+k)\) and \((q,\ldots,q+k)\) of template contour is established if \(S_{pq}\) is the smallest one out of \(mn\) variances. These correspondences are validated in two steps: first, computes an affine transformation as shown in (5), in order to map the region contour onto the coordinate system of the example template:

\[
x' = a_{11}x + a_{12}y + a_{13}, \\
y' = a_{21}x + a_{22}y + a_{23},
\]

(5)

where \(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}\) represent the mapping parameters. After registering the two regions using the affine transformation, compute the Hausdorff distance [20] for each point \((X^{RT}_j, Y^{RT}_j)\) on the mapped object region contour:
\[ L_j = \min \left( \sqrt{\left( X_j^R - X_j^T \right)^2 + \left( Y_j^R - Y_j^T \right)^2} \right) \quad i = 1, 2, ..., t \quad (6) \]

where \((X_j^T, Y_j^T)\) denotes a point on the contour of the example template. The distances are then used to classify all the points into two sets by thresholding: Sets 1 and 2 represent those points with \(L_j < \text{threshold}\) and \(L_j > \text{threshold}\) respectively. In cases where the region represents a part of the template object (Set 1 case), all the representative points should form a continuous curve. The continuity of the curve and the number of the points are adopted as a final criterion to determine the existence of a partial match.
Chapter 3

3. Object Tracking Algorithm

The proposed algorithm is capable of tracking partially occluded objects and correctly labeling disjointed parts of the same object using the partial shape matching procedure described above. It consists of two major parts: low-level frame analysis and object tracking (see Fig. 3). These are described in the following sub-sections.
3.1 Low level Frame Analysis

The low-level frame analysis consists of three principal steps: background subtraction, shadow suppression, and region segmentation as shown in Fig. 4. It is utilized to identify the moving objects of interest in two consecutive frames, namely \( t \) and \( t+1 \), and provide their respective segmentation maps. Objects of interests exhibit motion changes through subsequent frames and are initially identified by background subtraction. These objects consist of elementary and/or combinations of elementary regions as labeled by the segmentation module [25].
Reference frames acquired during the first 3-4 seconds of the video surveillance sequence are used to model the background of the scene for the purposes of initialization and training. In this model, each image pixel is described by its mean and standard deviation, derived from all of the acquired frames. Let $\mu(l)$ and $\sigma(l)$ define the mean and standard deviation in lexicographic ordering at pixel location $l$ respectively. These are given by:

$$\mu^C (l) = \frac{1}{N} \sum_{i=1}^{N} I^C_i (l) \quad \quad C = R, G, B.$$  \hspace{1cm} (7)
\[
\sigma^C(l) = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (I^C_i(l) - \mu^C_i(l))^2}
\]

\[C = R, G, B.\]  

(8)

where \(N\) is the number of training frames, and \(I_i(l)\) is a three-dimensional lexicographically ordered color vector describing the pixel at location \(l\) in the \(i\)-th frame (i.e. pixel’s color intensities in RGB color space). The threshold utilized for background/foreground classification at pixel \(l\) is determined by:

\[
\lambda^C(l) = \alpha_s \cdot \sigma^C(l)
\]

(9)

where \(\alpha_s\) is an empirically determined object segmentation constant. At a given frame \(t\), \(\lambda(l)\) is utilized to separate the moving objects of interest from the stationary background.

This is done by utilizing the algorithm described in pseudo-code below:

\begin{verbatim}
For each pixel \(l\) in frame \(t\):
    If abs[I^C_c(l) - \mu^C(l)] > \lambda^C(l)
        pixel \(l\) is classified as foreground
    else
        pixel \(l\) is classified as background
\end{verbatim}

Morphological operators (dilation and erosion) [22] are subsequently employed to remove small outliers (noise artifacts) in the foreground region(s).

It is desirable to prevent shadows from being misclassified as moving objects as demonstrated in [24]. In order to achieve this, all pixel information (acquired image frames and background) is converted from RGB to HSV color space (conversion detailed in Appendix I) to closely correspond to the human perception of color [23]. Let \(B = [B^h, B^s, B^v]\) represent the HSV components of the background model. The “shadow” pixels are identified by utilizing the set of conditions [24] shown in Equation 10 below:
where $I_t$ represents the input image (frame $t$) in HSV space. Hence, the shadow information is comprehended by the ratio of $I_t$ to $B$ in the Value channel and the difference between $I_t$ to $B$ in the Hue and Saturation channels. The ratio of $I_t$ to $B$ is bounded by the upper threshold $\beta$ to provide robustness to noise and the lower threshold $\alpha$ to account for the strength of the light source responsible for the generation of the shadow [24]. The thresholds $\tau_S$ and $\tau_H$ are determined empirically depending on lighting conditions.

Once the shadows have been properly discounted, the remaining foreground object(s) are segmented into elementary regions using a GRF-based segmentation algorithm that utilizes both color and edge information [25]. The procedure merges two neighboring color regions when their means are “similar” and there are no edge elements between them. The regions in the integrated segmentation map are represented as elementary nodes of a segmentation tree. The spatial relationships between them are stored in an adjacency matrix. The details of this algorithm are described in [25].

### 3.2 Tracking Algorithm

The object tracking step (see Figure 5 for a flowchart) matches and tracks potential objects of interest found in the segmentation maps of frames $t$ and $t+1$, respectively. The identified objects of frame $t$ are used as templates for partial matching to potential objects
found in frame \( t+1 \). The proposed tracking algorithm, illustrated in Figure 5, is divided into several steps as shown below:

1) Generate the adjacency matrices of the elementary regions in frames \( t \) and \( t+1 \) based on the segmentation maps provided by the low level frame analysis section.

2) Select objects \( MVO'_i \) and \( MVO'_{i+1} \) as candidates for matching.

3) Select the largest elementary region in \( MVO'_i \) and \( MVO'_{i+1} \) respectively.

4) Perform sub-matrix partial shape matching on the selected regions. Every shape match is validated by utilizing the procedure described in Section 2 followed by a simple color distance measurement, calculated as the Euclidian distance between the mean colors of the matching regions from frames \( t \) and \( t+1 \). The match is valid if the color distance is less than a user specified threshold.

5) If a match is found, combine matched regions with their respective adjacent regions and return to Step 4. The aim of this algorithm step is to identify the largest matching combination of elementary regions.

6) If a match is not found, the algorithm is redirected to another valid combination of elementary regions. The aim of this step is to find any suitable partial match between parts of the observed MVOs.

If all candidate combinations are exhausted and a partial match is not found, the next object pair \( MVO'_i \) and \( MVO'_{i+1} \) is selected and the algorithm is redirected to Step 3 until all suitable combinations of moving objects in the scene have been examined.
Figure 5: Flowchart of the Tracking Algorithm

An example of the above described algorithm is visually illustrated in Figure 6. To this effect, a typical MVO (a vehicle extracted from one of the test video sequences) and its
segmentation map is used to demonstrate the progression of the algorithm through Steps 3, 4, and 5. The largest elementary region of the MVO is combined with adjacent matched regions (see Figs. 6d-h), resulting in full object tracking.

![Diagram](image)

Figure 6: Visual illustration for matching algorithm: (a) Typical moving object, (b) Segmentation map, (c) Adjacency matrix, (d)-(g) Combination of successfully matched elementary regions and (h) the largest match History.

There are three distinctive tracking scenarios based on the number of objects in consecutive frames. There could be: a) the same number of objects in both observed frames (object continuity), b) less objects in the current frame than in the previous frame (object merging, or object exiting the scene), and c) more objects in the current frame than in the previous frame (object splitting and/or occlusion, or object entering the scene). These scenarios are explained in details in the following subsections.
3.2.1 Case I (object continuity)

Let $MVO_i^t$ and $MVO_j^{t+1}$ $(i, j = 1, 2, ..., n)$ denote the number of MVOs in frames $t$ and $t+1$, respectively. In this case, the number of moving objects in both frames is the same (i.e. object continuity), so object tracking is achieved by establishing correspondences for every object in frame $t$ and $t+1$, respectively. The partial matching procedure results in generating full matches for each of the elementary regions under the assumption that all objects are visible in both frames.

3.2.2 Case II (Object Merge/Exit)

The number of moving objects is reduced between consecutive frames $t$ and $t+1$. The possible scenarios are: 1) Two or more objects “interact” thereby creating partial or full occlusion from one frame to another (e.g., merging of two moving objects is a special case of occlusion wherein one object is passing in front of the other), 2) Disappearance of the object from the scene; and 3) Parts of a previously split object merge to form a single object once again (Fig. 11e-11f). In order to track the objects undergoing a “merge”, the algorithm uses shape templates of the MVOs undergoing occlusion (represented by a combination of the elementary regions discovered in previous frames - Fig. 6h) for partial shape matching. Thus, the tracker identifies and separately tracks each object throughout the frames independent of their translations, rotations, scale variations, and/or occlusions. In the event of an object exiting or disappearing from the scene, no correspondence is found for the particular object of the previous frame in the current frame.
3.2.3 Case III (Object Split/ Entrance)

The number of moving objects increases from frame $t$ to $t+1$. The possible scenarios are: 1) Entrance of new object(s) into the scene; 2) Objects separating after mutual occlusion; and 3) Splitting of an object into pieces (occlusion by an immovable entity or entities).

New tracks are initiated for those $MVO^{t+1}$’s that have not received any correspondences to the previous frame. In the event of split, the tracker assigns similar object labels for $MVOs$ in the current frame from the $MVO$ of the previous frame undergoing a split occlusion.

3.2.4 Computational Aspects of the Tracking Algorithm

The computational aspects of the algorithm grow exponentially with the number of tracked objects found on the scene; especially in the case of over-segmentation of object regions. In order to alleviate this, shape matching is performed only for MVOs that are within a maximum allowable spatial displacement. The value of that displacement is set according to the application at hand and is increased proportionally to the speed of the moving objects and/or the rate of the frame acquisition. To further improve the computational efficiency of the algorithm, the largest region is selected to start the partial shape matching procedure as discussed in Step 2 above. This leads to a smaller number of combinations necessary to establish a complete match.
Chapter 4

4. Experimental Results and Discussions

In this chapter, the performance of the proposed algorithm is demonstrated on several simulated and real life image sequences. The sequences chosen for testing consist of different complex scenarios and difficulties in order to demonstrate the effectiveness of the proposed tracker. For each sequence (Figs. 7, 9, 11, 13, 15), the results are displayed on several non consecutive frames where the contour of the tracked objects are highlighted followed by the corresponding trajectory paths (Figs. 8, 10, 12, 14, 16). All sequences are captured at 30 frames per second.
Figure 7: Tracking results for “Lab simulated” sequence (a)–(f) represent frames 18, 59, 79, 92, 109, and 152 (Red outline around the object indicates result of tracking algorithm).
Figure 8: Object trajectory for Lab simulated Scene. The horizontal and vertical axes of the graphs represent the width and the height of the frame respectively.

The first sequence consists of a 20 seconds Lab simulated video shot with a typical surveillance camera mounted on the ceiling. The frames shown in Fig. 7 display a toy truck undergoing a circular motion against a simple background. Since the algorithm uses shape templates (represented by the combination of elementary regions generated by the segmentation map from the previous frame) to establish correspondence in the current frame, and with the underlying assumption that the frame rate is 30 frames/sec, the motion of the object is well modeled between frames $t$, $t+1$, and frames $t+1$, $t+2$ by an affine transformation with possible occlusions if any. Tracking is accomplished (indicated by pink line around object contour in Fig. 7) invariant to translations, rotations and minor scale variations. The object trajectory is shown in Figure 8.
Figure 9: Tracking results for “RIT Parking lot” sequence (a)–(f) represent frames 2, 11, 25, 37, 44, and 56 (Pink outline around the object indicates result of tracking algorithm).
The Second sequence consists of 50 frames of an outdoor real life complex background video typical to the ones captured on surveillance cameras in sports stadiums or secure type facilities. The frames shown in Fig. 9 display a truck that is moving from right to left in the scene where its projected 2D shape is being changed due to 3D translations and rotations. In addition, minor occlusions are created by the presence of the pole in the scene. The proposed algorithm was capable of segmenting, isolating and tracking (see Fig. 10) the moving object successfully throughout the video (see pink line around object contour in Fig. 9) invariant to translations, rotations and minor occlusions.
Figure 11: Tracking results for “toy cars” sequence (a)–(f) represent frames 8, 20, 26, 30, 33, and 38 (Yellow and pink outline around the objects indicates result of tracking algorithm).
The third sequence consists of 40 frames of a lab video sequence in which a toy car and truck are moving in opposite directions. Note that the car undergoes various scenarios of partial occlusion due to the presence of the box in the scene where in some frames the front and rear end of the car are visible while the center section is totally occluded. Fig. 11b-11f shows the blue truck and green car being tracked by the proposed algorithm, where each object is given a unique label. The front end of the green car is being progressively occluded (Fig. 11b, 11c), displays a “split” occlusion scenario – center section occluded - (Fig. 11d, 11e) resulting in 2 objects, and then progressive back end de-occlusion (Fig. 11f). The algorithm still manages to assign the same object label to the car and truck objects by using the partial shape matching module. The corresponding trajectories are shown in Figure 12.
Figure 13: Tracking results for “RIT junction” sequence (a)-(b) frames 25, 26 (c)-(d) frames 29, 30 (e)frame 32 (f)-(h) frames 34, 35, 37 and (i) frame 38 (Yellow and pink outline around the objects indicates result of tracking algorithm).
Figure 14: Object trajectories (frames 20 to 38) for RIT Junction Sequence.

The fourth sequence consists of a 10 second outdoor, real life, video sequence acquired, with a hand held video camera, at a road junction at the Rochester Institute of Technology. The sequence was selected to test the robustness of the algorithm where occlusions are being created by foreground and background moving objects. The frames shown in Fig. 13 display a truck and a van approaching each other (Fig. 13a) resulting in an early partial occlusion which lasts for approximately 10-12 frames, developing progressive occlusion (Fig. 13b, 13c), undergoing severe occlusion (Fig. 13d), and then progressive de-occlusion (Fig. 13f, 13h) and finally a split (Fig. 13i). To indicate this, the matched object boundaries are highlighted in “yellow” and “purple” for display purposes. Figure 13e displays the truck is heavily occluded by the van in frame 32, the tracker fails to establish a correspondence for the truck since the area encompassed by it is insufficient for effective shape matching (size constraint). However it is able to re-track the truck
after it has reappeared with sufficient enough frame size. The proposed algorithm effectively tracks (see Figure 14) each object independent of the occlusion scenario throughout the sequence.

Figure 15: Tracking results for “Parking Lot” sequence (a)–(i) represent frames 4, 24, 36, 48, 62, 70, 74, 80, and 92
Figure 16: Trajectories of the objects for parking Lot Sequence.

Figure 15 shows tracking results from sample frames of a typical surveillance sequence shot at a parking lot where the two cars undergo translation, rotation, scale variations and merging. In contrast to the previous scene, the objects of interest possess similar colors and shape in order to stress the tracking algorithm. Fig. 17 (a) shows the regions (segmentation map) of the objects being tracked in frame 68. These are used as templates for shape matching in frame 69 and so on as the objects are undergoing a “merge” operation. Table 1 provides the corresponding Hausdorff distance values (similarity measure detailed in appendix II) for each of the objects throughout several frames. Thus, given the data in Table 1, it is evident that the proposed tracking algorithm was capable of successfully handling the “merge” scenarios shown in Figure 15 (i.e. assigning a unique label to each object) yielding the object trajectories shown in Figure 16.
Figure 17: (a)-(d) Object regions discovered in the frames as labeled above.

<table>
<thead>
<tr>
<th>Frame No.</th>
<th>Object 1 Template</th>
<th>Object 2 Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>23.09</td>
<td>29.61</td>
</tr>
<tr>
<td></td>
<td>37.48</td>
<td>23.54</td>
</tr>
<tr>
<td>74</td>
<td>11.70</td>
<td>46.84</td>
</tr>
<tr>
<td></td>
<td>31.02</td>
<td>28.02</td>
</tr>
<tr>
<td>80</td>
<td>08.60</td>
<td>35.22</td>
</tr>
<tr>
<td></td>
<td>58.83</td>
<td>22.00</td>
</tr>
</tbody>
</table>

Table 1: Similarity Metric for shape: Objects of frame 68 (Templates) compared to Objects of frame 72, 74 & 80 undergoing merging.

Table 2 lists the typical MVO(s) size, frame size, number of frames and the processing time utilized by the proposed tracking algorithm for the above sequences. The algorithm was prototyped in a MATLAB [28] environment. The processing time per frame is directly proportional to the number of elementary regions (generated by the segmentation module) present in a segmentation map of a given MVO and the number of MVO’s present in the scene. Note that the processing speeds for video Sequences 1 and 2 are almost equivalent since they both possess a single MVO and the same number of elementary regions (typically 2-3) as generated by the segmentation map. Whereas video
Sequence 4 with two MVO's takes approximately 5 times more processing time per frame as compared to video Sequences 3 and 5, also with two MVO's. This is due to the fact that the segmentation map for Sequence 4 consists typically of 5-6 regions for a given MVO as opposed to 2-3 regions for a given MVO in Sequences 3 and 5. These algorithms can be made to operate in real or near real time using hardware implementations.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Object(s) size (Pixels)</th>
<th>Frame size (Pixels)</th>
<th>No. of frames processed</th>
<th>Total processing time (secs)</th>
<th>Processing time/frame (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lab simulated</td>
<td>135 x 60</td>
<td>640 x 480</td>
<td>285</td>
<td>2920.11</td>
</tr>
<tr>
<td>2</td>
<td>RIT Parking Lot</td>
<td>110x70</td>
<td>640 x 480</td>
<td>65</td>
<td>756.10</td>
</tr>
<tr>
<td>3</td>
<td>“Toy cars”</td>
<td>80x40</td>
<td>65x35</td>
<td>280 x 180</td>
<td>1280</td>
</tr>
<tr>
<td>4</td>
<td>RIT Junction</td>
<td>124x54</td>
<td>108x50</td>
<td>320 x 240</td>
<td>4800</td>
</tr>
<tr>
<td>5</td>
<td>Parking Lot</td>
<td>145x60</td>
<td>77x34</td>
<td>720 x 480</td>
<td>2200</td>
</tr>
</tbody>
</table>

Table 2: Computation cost for the proposed tracking scheme.
Chapter 5

5. Conclusions and Future Work

This thesis describes a tracking algorithm that utilizes partial shape matching to track objects in video surveillance applications. The algorithm is capable of handling translations, rotations, scale variations and is effective in instances of occlusions. The approach was successfully tested on several video surveillance sequences. The performance of the tracker requires that the region(s) that constitute the object(s) be clearly separated by the segmentation algorithm from the “clutter”. It should be noted that
any effective region segmentation technique could be utilized to perform this particular step. In addition to generating meaningful regions the segmentation should generate less number of elementary regions so as to reduce the computational cost of the matching procedure. In instances where the object has undergone full occlusions and then re-appeared in the scene (grow in size), it may take more than 1 frame to re-establish tracking (as shown in RIT junction Sequence Fig. 13e-f). Non stationary cameras can be handled using the proposed algorithm by performing frequent background updates.

Figure 18: Object Tracking and Recognition using Multiple Multimodal Sensors.

In future studies, a plan to develop an information fusion framework as shown in Fig. 18 is being considered, that takes advantage of the data generated by multiple sensors - due to their collective ability to collect data from multiple angles/views - to significantly
enhance the object tracking and recognition and compensate for potential occlusions as viewed by a given sensor. In the general case, all sensors would be utilized, in an “inter sensor processing” framework, to improve tracking and recognition.
Appendix I – RGB to HSV Transformation

RGB is an m-by-n-by-3 image array whose three planes contain the red, green, and blue components for the image, while HSV is an m-by-n-by-3 image array whose three planes contain the hue, saturation, and value components for the image. Where hue is an angle from 0 to 360 degrees, typically 0 is red, 60 degrees yellow, 120 degrees green, 180 degrees cyan, 240 degrees blue, and 300 degrees magenta. Saturation typically ranges from 0 to 1 (sometimes 0 to 100%) and defines how grey the color is. Value is similar to luminance except it also varies the color saturation. The transformation between RGB and HSV is given by:

\[ H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \]  

\[ \theta = \cos^{-1} \left\{ \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right\} \]  

\[ S = 1 - \frac{3}{R+B+G} [\min(R,G,B)] \]  

\[ I = \frac{1}{3} (R + G + B) \]
Hausdorff distance is used to measure the resemblance between two objects that are superimposed on one another (as shown in Fig. 18). This similarity metric for a given pair of objects is computed as follows.

Given the image region and the example template’s point set $A = \{a_1, \ldots\}$ and $B = \{b_1, b_2, \ldots\}$ respectively, the hausdroff distance is defined as,

$$H(A, B) = \max(h(A, B), h(B, A))$$  
(1)

Where,

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$  
(2)

and $\|\|$ is the Euclidean norm. The Hausdorff distance, $H(A, B)$, is the maximum of $h(A, B)$ and $h(B, A)$. Thus measures the mismatch between two sets, by measuring the distance of the point $A$ that is farthest from any point $B$ (the most mismatched point of $A$) given by $h(A, B)$ and vice versa.

Figure 19: Template & image superimposed for measuring similarity metric
References


