VISUAL TRACKING WITH
GROUP MOTION APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ ERKİN ARSLAN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2003

Approval of Graduate School of Natural and Applied Science.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Mübeccel DEMİREKLER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Mübeccel DEMİREKLER
Supervisor

Examining Committee Members:

Prof. Dr. Kemal LEBLEBİCİOĞLU (Chairman)

Prof. Dr. Mübeccel DEMİREKLER

Assoc. Prof. Dr. Yasemin YARDIMCI ÇETİN

Assist. Prof. Dr. Aydın ALATAN

MSc. Umur AKINCI

ABSTRACT

VISUAL TRACKING WITH GROUP MOTION APPROACH

Arslan, Ali Erkin
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mübeccel Demirekler

September 2003, 72 pages

An algorithm for tracking single visual targets is developed in this study. Feature detection is the necessary and appropriate image processing technique for this algorithm. The main point of this approach is to use the data supplied by the feature detection as the observation from a group of targets having similar motion dynamics. Therefore a single visual target is regarded as a group of multiple targets. Accurate data association and state estimation under clutter are desired for this application similar to other multi-target tracking applications. The group tracking approach is used with the well-known probabilistic data association technique to cope with data association and estimation problems. The applicability of this method particularly for visual tracking and for other cases is also discussed.

Keywords: visual tracking, video tracking, feature tracking, multi-target tracking, group tracking, probabilistic data association, Kalman filtering.

ÖZ

GRUP HAREKETİ YAKLAŞIMI İLE GÖRSEL İZLEME

Arslan, Ali Erkin
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mübeccel Demirekler

Eylül 2003, 72 sayfa

Bu çalışmada görsel hedefler için bir izleme algoritması geliştirilmiştir. Öznitelik bulma işlemi bu algoritmada gereken piksel işleme tekniğidir. Bu uygulamadaki temel yaklaşım öznitelik bulma işlemi ile elde edilen verileri, yakın dinamiklere sahip hedef grubuna ait ölçümler olarak değerlendirmektir. Bu nedenle görsel hedef pek çok hedeften oluşan bir grup olarak değerlendirilmektedir. Gürültü altında hassas veri eşleştirmesi ve durum kestirmesi diğer çoklu hedef izleme uygulamalarında olduğu gibi burada da önemlidir. Grup izleme yaklaşımı, olasılıksal veri ilişkilendirme tekniği ile beraber kestirme ve ilişkilendirme sorununa karşı kullanılmıştır. Metodun görsel ve diğer uygulamalar için uygunluğu da tartışılmıştır.

Anahtar Kelimeler: görsel izleme, video izleme, belirti izleme, çoklu hedef izleme, grup izleme, olasılıksal veri ilişkilendirme, Kalman filtreleme.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

CHAPTER 1

INTRODUCTION

Applications of tracking are so wide that the term differs in meaning depending on in which area it is used. A search for the word "tracking" on the web usually results in many radar tracking applications. This area is where tracking techniques found applications and flourished. Video tracking is mostly and historically used for tracking IR video and it has also applications with day video. Video tracking systems are used in military, for guidance, navigation, passive range estimation and target discrimination. In security area, it is used for fire control, person tracking and traffic surveillance. In industry, robotics and automation are the areas for video tracking. Tracking applications mostly include methods from signal processing, estimation and control disciplines. This chapter gives the definitions of tracking and video tracking. An introduction to the basic problems and to the topics where the research is focused is also made in this chapter.

The goal of the thesis is to develop a single target tracking method for visual targets. Using visual features of the target can result in accurate tracking where the formation of the target will be known during the tracking. Therefore these visual features are used in this study. However, using multiple features required to develop a multiple target tracking approach. Considering the dependency of features to the tracked target, a group tracking approach is developed. The novel part of this work is using PDA based approach to track visual features having group motion. The group model is developed considering the dynamics of visual features and tracking with PDA based association is preferred to increase accuracy for the individual features in the group.

1.1. Introduction to Video Tracking

Tracking is estimation of current and future states of sources by classifying the current sensor data considering the source(s) it originates from. This definition is general and different sensors and sources result in different tracking applications.

In automatic video tracking, examples for sources can be air and ground vehicles. The sensor may be IR or day light sensitive imager. Definition of video tracking can be made as the estimation of location of visual objects within the sensor field of view using image and data processing. Video tracking is generally used to maintain a stable sensor-to-target line of sight automatically in the presence of target and platform motion [1]. Therefore a closed-loop system is present in most video tracking systems as seen in Figure 1.

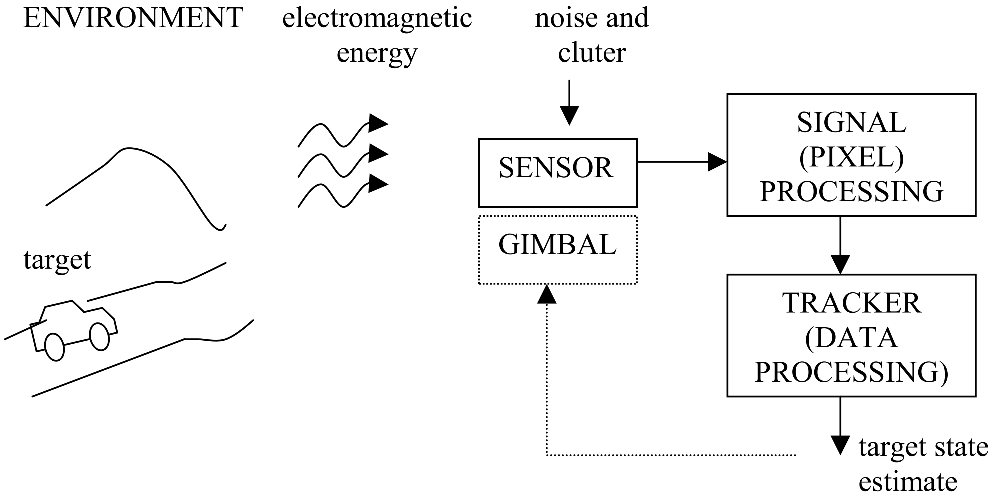


Figure 1- Components of a generic video tracker including the feedback loop.

Pixel processing for target detection requires high computation power. The output of this step, the measurement information, is input to the tracker. The structure of this information determines the tracker's technique. In other words, information structure directs how data processing for track to measurement association and state estimation is made in the tracker.

In 1970s and 1980s innovative hardware designs were required to cope with the limited signal processing power of the available hardware. After mid-1980s better A/D converters, powerful digital signal processors (DSP) and application specific integrated circuits (ASIC) turned the design focus toward software [1]. Today, inevitable use of high performance re-configurable hardware (FPGAs) for signal processing together with DSPs determines the new course of video trackers [2,3]. Detection and estimation theory plays more role in the design process parallel to the increase in computation power.

Detection phase of a video tracker is based on well-known pixel processing techniques. Thresholding using target statistics, 2-D correlation, edge and corner feature detection are some fundamental tools for measurement gathering from raw pixel data [1,3,4,5,6,13,14].

When measurements are available, suitable information processing technique should be used to deal with them. This is the tracking and data association side of the tracker. "*It was recognized as far back as 1964 [7], that in target tracking there can be an uncertainty associated with the measurements in addition to their inaccuracy, which is usually modeled by additive noise*" [8]. This uncertainty is related to the origin of the measurements. They may not originate from targets of interest or wrong associations of targets to measurements are possible.

Although the approaches for tracking will be explained in the following chapters, it may be appropriate to mention the names of mostly used and known ones here. When we are dealing with a single target tracking problem, measurements may be target originated or false alarms caused by clutter. If the target has a constant dynamic model the problem can be solved using optimal Bayesian approaches or with sub-optimal probabilistic data association filter (PDA) [9]. When the target model switches in time multi-model PDA can be used in tracking [9]. If tracking of multiple-targets is required, joint probabilistic data association filter (JPDA) and multiple hypothesis tracking (MHT) are the fundamental Bayesian techniques. If there are multiple targets having dependent motion such as a convoy of vehicles, it turns into group tracking where this dependency can be used effectively [10,11].

After this brief introduction it can be said that a good tracker is the one that converts the raw data into information efficiently and use this information with the most appropriate information processing technique so that state estimation of the target(s) is found satisfying the accuracy and latency requirements of the system.

1.2. Thesis Overview

This thesis is devoted to tracking a single visual target using multiple features. The method developed uses a group motion model and PDA based data association. It should also be stated that this tracking method is developed for video tracking applications. Although multiple features are being tracked with the developed algorithm the aim is to track single visual targets. The background information and the developed method are explained in the following chapters.

In Chapter 2 fundamental pixel processing techniques are explained with an emphasis on feature tracking. Features are patches from visual targets that can easily be tracked. The selection criteria of features and use of this information in the later steps of tracking is discussed in this chapter.

Chapter 3 is an overview of tracking algorithms and statistical tools used in tracking. The selection of the best approach considering the application requirements will be pointed. In this chapter PDA, JPDA and group tracking approaches, which are related to the algorithm proposed in this thesis, is covered.

In Chapter 4, a visual target tracking approach is developed using visual features of the target. The approach is basically a multi-feature tracker with group motion assumption. The constraints, requirements and application of this approach are also argued.

Chapter 5 covers implementation of the developed algorithm with simulations for some video sequences and artificial data.

Chapter 6 is on the conclusion, summary and possible future work after this study.

CHAPTER 2

PIXEL PROCESSING FOR VISUAL TRACKING

2.1. Introduction

In visual tracking problems, the first phase is processing of pixel data to get valuable information for the position of the targets. Alternative methods for detection of targets in the image plane have different complexity and advantages. The well-known centroid and correlation techniques are most popular ones that satisfy real time performance and used in most video tracking applications [1,5]. Computer vision is used to extract relevant information from an image or sequence of images, and to represent this information in a way so that higher level reasoning processes can interpret it. For computer vision applications corner detection and optical flow are mostly used where corner detection is preferred for structure from motion studies and optical flow, which can also be considered as an alternative method, is used for motion estimation for video compression [21].

The aim of this chapter is to present the corner feature detection method that the tracker developed in this study uses. The image processing techniques mentioned above will briefly be described for comprehensiveness. As stated before, the way the measurements gathered, affects the tracking phase. Therefore the advantages and properties of the measurement information generated by feature detection will also be discussed in this chapter.

2.2. Techniques for Target Detection from Video

In this section centroid, correlation, optical flow and feature correspondence will be discussed. Feature detection and related topics will be explained in detail in the following sections.

We begin with centroid trackers, which can be separated into two types. The binary centroid tracker calculates center of mass for a segmented image. Segmentation is classifying each pixel as target related or not. If a pixel is target related 1 is assigned and 0 is used to denote background pixels. On the other hand, intensity centroid trackers keep the intensity value of the target pixels for centroid calculations. Let the index *i* be used to denote the pixels in a target cluster of *n* pixels, where *i=1,…,N*. If pixel *i* of this cluster has intensity $I_{i,}$ then the n^{th} coordinate of the centroid of the cluster is defined as

$$x_{n_{centroid}} = \frac{\sum_{i=1}^{N} x_{n_i} I_i}{\sum_{i=1}^{N} I_i} \tag{2.1}$$

where x_{n_i} is the n-th coordinate of point i.

Histogram analysis for segmentation of target from background is important for centroid calculation accuracy. Details of this technique will not be discussed but it should be added that centroid trackers can be implemented with a relatively simple hardware successfully. More details of centroid method can be found in references [1,22].

Correlation technique can be used for template matching so that previously defined target template can be found in a search area of current scan using a match criterion. This criterion can be found by summing of squared differences of the corresponding pixels of the template and candidate patch. A better alternative may be using the correlation coefficient as

$$\hat{\rho}_{JI}(k) = \frac{\sum_{i=1}^{N}\left[I_i(k)-\bar{I}(k)\right]\left[J_i(k)-\bar{J}(k)\right]}{\sqrt{\sum_{i=1}^{N}\left[I_i(k)-\bar{I}(k)\right]^2 \sum_{i=1}^{N}\left[J_i(k)-\bar{J}(k)\right]^2}} \tag{2.2}$$

where *I* and *J* are template and candidate image patches with *N* pixels. The index *i* is used to denote the pixels in the patches where *i=1,…,N*. Details on correlation method can be found in [1,22,23]. Correlation technique is computationally more expensive then the centroid and can be efficient if the target signature is stable and target size is not relatively too small so that a reliable target signature is present for template generation.

Optical flow is one of the traditional ways used for the problem of computing motion and then structure. “Optical flow” is used to mean motion of iso-brightness contours in the image. The moving of this equal brightness pattern that represents the object can be described with a constraint equation as follows.

First assuming the image brightness varies smoothly and has no spatial discontinuity, consider a patch of image (brightness pattern) that has displaced a distance $(\delta x, \delta y)$ in a time interval δt. Assuming the brightness level of patch remain constant then,

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \tag{2.3}$$

where $E(x, y, t)$ is the intensity in the image at point (x,y) at time t. Using Taylor series approximation of equation 2.3 we can write

$$E(x, y, t) \cong E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + e \tag{2.4}$$

where e contains higher order terms, which are negligible when $\delta x, \delta y$ and δt are small. Therefore,

$$\frac{\delta x}{\delta t}\frac{\partial E}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} = 0 \tag{2.5}$$

defining image plane velocities (v_x, v_y) as

$$v_x = \frac{\delta x}{\delta t},\ v_y = \frac{\delta y}{\delta t} \tag{2.6}$$

results to the final well-known “Image Brightness Constraint Equation”

$$\frac{\partial E}{\partial x} v_x + \frac{\partial E}{\partial y} v_y + \frac{\partial E}{\partial t} = 0 \tag{2.7}$$

This equation contains two unknown velocity components and is not sufficient uniquely to specify optical flow. Additional constraints below are commonly used:

- Optical Flow is smooth and neighboring points have similar velocities.
- Optical flow is constant over an entire segment of image plane.
- Optical flow is a result of restricted motion such as planar motion.

Even when the above constraints are satisfied optical flow suffers from a drawback known as aperture problem: the motion of image points can only be determined normal to the image intensity contour while tangential components are unobservable. There are some methods to cope with this problem but in general optical flow has been found to be too error prone for use in structure from motion algorithms. It is also a computationally expensive method.

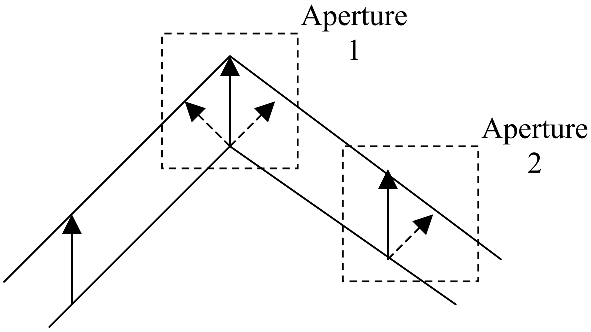


Figure 2- Illustration of the aperture problem: The displacement estimation (the dashed arrow) in second aperture is normal to the intensity contour (line) whereas true displacement is the solid arrow. In aperture 1, corner has gradient in the two perpendicular dimensions and true motion is estimated using this property.

It is also possible to track objects by selecting their particular, reliable features and finding their correspondence in successive frames. These features can be edges, corners or textures, which can reliably be identified in successive frames. Two criteria for selecting good features can be:

- Features should be distinct and clearly defined.
- Features should be temporarily stable, *i.e.* over successive frames the same feature should look similar.

Edges are one dimensional features with only the motion normal to the direction of edge can reliably be recovered. Two-dimensional features are appealing because they are not affected from aperture problem [13,14]. Optical flow and feature correspondence can be compared as [23]:

- Optical flow is noise sensitive due to its dependence on spatio-temporal gradients.

- Optical flow requires that the motion is smooth and small (This is not possible when inter scan period of imager is high).
- Optical flow requires that the motion vary continuously over the image.

For feature correspondence:

- Correspondence between features and frames must be established and maintained between successive frames.
- Rigid motion of the object is usually assumed, *i.e.* static world assumption.
- Stable and accurate algorithms for feature detection are difficult to implement considering the poor localization accuracy of many feature detectors.

Before going in details it is important to understand the distinction between detecting and tracking features. Detection is finding regions satisfying feature definition whereas tracking features requires prior appearance of the feature and an estimate of the current position. Temporal instability of the features is an unwanted situation for efficient tracking therefore only stable features should be used in tracking. In addition to small random displacements of features, occlusions, lightning changes, noise in the camera and digitizer and camera motion can cause feature instability. Next section is on feature detection methods.

2.3. Corner Feature Detection

Corners are distinctive points that are well localized in both directions. They are reliable features and do not have aperture problem. There are some different corner detection approaches where most of these are similar to each other and implicitly or explicitly they search for high curvature on edges [13,14]. Only the corner detection technique, used in this study, will be explained in detail. The method explained here is based on the method Lucas, Kanade and Tomasi developed [24,25,26].

A patch, which has well defined peak in its auto-correlation function, can be classified as a corner. That is, since we are looking for distinctive points we expect that when a corner patch or a distinctive region is compared with its neighbor region using sum of squared differences, we must get a high value. If we compute the change in intensity as the sum of squared differences in the direction $\boldsymbol{h}$ for a patch W centered in $\boldsymbol{x}=(u,v)$:

$$E_h(\mathbf{x}) = \sum_{d \in W} \left[I(\mathbf{x}+\mathbf{d}) - I(\mathbf{x}+\mathbf{d}+\mathbf{h}) \right]^2 \tag{2.8}$$

using the Taylor series expansion truncated to the linear term,

$$E_h(\mathbf{x}) = \sum_{d \in W} \left[I(\mathbf{x}+\mathbf{d}) - I(\mathbf{x}+\mathbf{d}) - \nabla I(\mathbf{x}+\mathbf{d})^T \mathbf{h} \right]^2$$

where

$$\nabla I(\mathbf{x}+\mathbf{d})^T = \begin{bmatrix} \dfrac{\partial I(\mathbf{x}+\mathbf{d})}{\partial u} & \dfrac{\partial I(\mathbf{x}+\mathbf{d})}{\partial v} \end{bmatrix} = \begin{bmatrix} I_u(\mathbf{x}+\mathbf{d}) & I_v(\mathbf{x}+\mathbf{d}) \end{bmatrix}$$

then

$$E_h(\mathbf{x}) = \sum_{d \in W} \left[\nabla I(\mathbf{x}+\mathbf{d})^T \mathbf{h} \right]^2 = \sum_{d \in W} \left[\mathbf{h}^\mathbf{T} \nabla I(\mathbf{x}+\mathbf{d}) \right) \left[\nabla I(\mathbf{x}+\mathbf{d})^T \mathbf{h} \right]$$

$$E_h(\mathbf{x}) = \sum_{d \in W} \mathbf{h}^\mathbf{T} \nabla I(\mathbf{x}+\mathbf{d}) \nabla I(\mathbf{x}+\mathbf{d})^T \mathbf{h} = \mathbf{h}^\mathbf{T} \left(\sum_{d \in W} \begin{bmatrix} I_u^2(\mathbf{x}+\mathbf{d}) & I_u(\mathbf{x}+\mathbf{d}) I_v(\mathbf{x}+\mathbf{d}) \\ I_u(\mathbf{x}+\mathbf{d}) I_v(\mathbf{x}+\mathbf{d}) & I_v^2(\mathbf{x}+\mathbf{d}) \end{bmatrix} \right) \mathbf{h}$$

$$E_h(\mathbf{x}) = \mathbf{h}^\mathbf{T} \mathbf{C} \mathbf{h} \tag{2.9}$$

Equation 2.9 gives the change in intensity around $\boldsymbol{x}$, where

$$\mathbf{C} = \begin{bmatrix} I_u^2(\mathbf{x}+\mathbf{d}) & I_u(\mathbf{x}+\mathbf{d}) I_v(\mathbf{x}+\mathbf{d}) \\ I_u(\mathbf{x}+\mathbf{d}) I_v(\mathbf{x}+\mathbf{d}) & I_v^2(\mathbf{x}+\mathbf{d}) \end{bmatrix} \tag{2.10}$$

If $||\boldsymbol{h}|| = 1$ then

$$\lambda_1 < E_h(\mathbf{x}) < \lambda_2 \tag{2.11}$$

where λ_1 and λ_2 are the eigenvalues of $\boldsymbol{C}$. So for any orientation $\boldsymbol{h}$ the minimum change in intensity will be λ_1. Therefore we evaluate a patch k as a good feature if the smallest eigenvalue of the related matrix $\boldsymbol{C}_\mathbf{k}$ is above a determined threshold.

The corner detection technique described above implies that a minimum of the smallest eigenvalues of the patches should be determined to accept patches as corners. This value may be problem dependent but a robust approach can be generated for general use. Next section will cover the approach developed to detect features in our tracking algorithm.

2.4. The Use of Corner Features

Corner features on the image can be generated from true corner like structures, such as the ones generated from vehicles. Another possibility is that non-physical corners can be present on the image plane because of the projection of different real world edges. These will be unstable and non-informative. They can be considered with the other spurious background corner measurements. Other than spurious features, some real world corners will also flicker during the track because of sensor noise or clutter. The important point is to find reliable target originated features for the quality of track. Reliability is related with temporal stability of the features.

In this study, it is assumed that the tracking is started with selection of a window, that contains the target in the current frame, by the operator. This window will be named as target window and we assume that this window contains mainly target related features. Figure 4 is an example for target window selected by the operator, in the frame given by Figure 3.

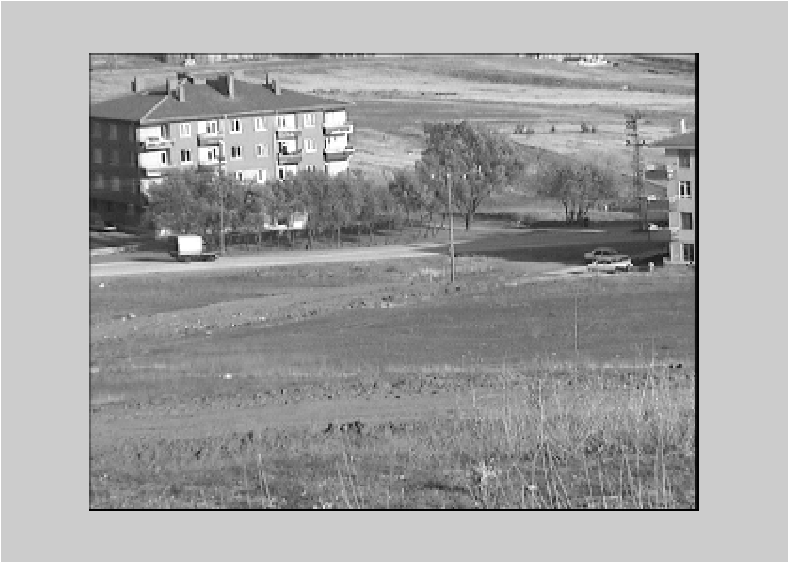


Figure 3- Example for an initial frame.

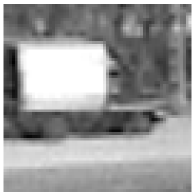


Figure 4- An example for the target window selected by operator to start tracking.

Feature detection is carried on this interest region. Applying corner detection method explained in the previous chapter, we can find the small eigenvalues of all patches in the target window. Denoting each patch with its center pixel *i* and the smallest eigenvalue of the patch with λ_1^i, we can get an eigenvalue map (*eigenmap*) of this patch as seen in Figure 5.

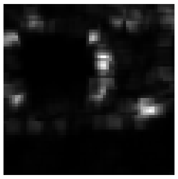


Figure 5- Eigenvalue map, showing the smallest eigenvalues of corresponding C matrices for each patch in the target window.

In this figure, intensity value of any pixel *i* is λ_1^i to visualize the eigenvalues of the patches in a gray scale image. The dark regions represent the locations (centers) of the patches where λ_1 of the patch is small compared with the ones of other patches. Bright regions are for the patches where corresponding λ_1 values and therefore cornerness property of the patch is higher than the others. The terms small and high are fuzzy. If we don't have a certain threshold value at hand to compare with the eigenvalues of features, we can still manage to find target features we require for the rest of tracking. This is what we used for our tracking algorithm and will be explained in detail now.

First we can define some constraints for features to simplify both detection and tracking phases. The feature patch size, *fpatchsize*, is the first parameter to determine. This determines the selected dimension of feature patches in each direction and it should be determined based on the size of the target we are tracking. If the target is relatively small as 50x50, using a patch size of 20x20 will result in detection of too few features for example. For such size of targets, patch size of 5x5 is reasonable.

We can also constraint the maximum number of detected features, *fn*, per search area. This limitation on the number of features is necessary for the feasibility of the tracking algorithm described in Chapter 4.

A distinction constraint, *fspan*, is also useful. This parameter determines the minimum allowable distance between neighboring detected features in each coordinate. It is used for keeping target features unique. It can be understood easily if Figure 5 is investigated. It is seen that patches with high cornerness (brighter pixels) are cumulated on some regions in the map. If a patch has a high cornerness, a second patch centered one pixel next to this patch will share some pixels with the first one and therefore will also have high cornerness. If we use these two features in the tracking, the same corner region will be represented by two features. This does not give any additional information but increases ambiguity for the tracker.

In other words, resolved and unique features are required for the tracker and this can be satisfied by searching for local peaks in the map and discarding their neighbors using the discrimination (separation) value, *fspan*. So *fspan* is the span of the region to be discarded during the rest of feature detection around the detected patch, which has a local peak considering the smaller eigenvalues of patches.

Figure 6 shows the effect of *fspan* value in feature detection for the eigenvalue map in Figure 5. Setting *fspan* as 0 means that there is no constraint on distinction and any neighboring features can be accepted. Similarly setting *fspan* to 1 will avoid patches P(x+1,y), P(x-1,y), P(x,y+1), P(x,y-1), P(x+1,y+1), P(x+1,y-1), P(x-1,y-1), P(x-1,y+1) to be accepted as feature if patch P(x,y) (centered at (x,y)) is accepted once during the feature detection phase of current frame. See how neighboring features are accepted if fspan is 0 and the features become unique when distinction value *fspan* is increased.

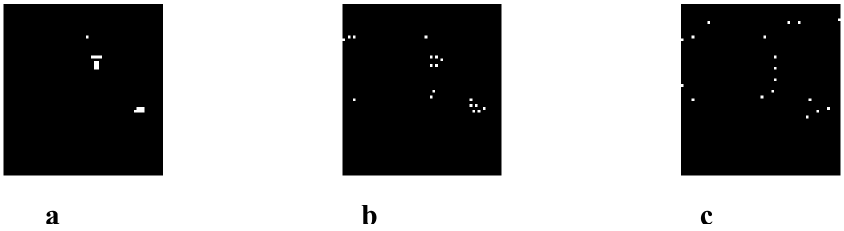


Figure 6- Illustration for the effect of distinction constraint: a-) fspan=0, b-) fspan=1, c-) fspan =3; White points are the centers of the selected feature patches (all other parameters are kept fixed for a, b and c).

Then, we can find the features in the first target window using the constraints *fn, fspan* and *fminthre,* the minimum thereshold value for the smaller eigenvalue of a patch to be accepted as feature. Figure 8 is the flowchart representation of feature detection process.

Since *fminthre* is a predefined threshold, it is set to a low value at the initial frame, because there is no information about the eigenvalues of patches so far. The process starts by finding the patch having maximum smaller eigenvalue, λ_1. If λ_1 is higher then *fminthre* threshold, this patch is saved as the first detected feature. This feature and the patches in *fspan* neighborhood are removed from candidate list. This process is repeated until the number of saved features are equal to *fn* or the λ_1 of the patches left are below *fminthre*. The minimum of λ_1 values of the selected features is saved as *fmineig*, to use in next frames. When eigenmap of the next frame is available *fn* is increased with a factor to allow for more features to be detected so that target features tracked are kept detected even when the target is disturbed by spurious features with high cornerness during the track. If this is not the case, however, increasing maximum allowable number of features constraint, *fn*, may result in accepting new features with low λ_1. Therefore also *fminthre* value is updated with the *fmineig* found from the initial frame. In the first frame *fminthre* is a low, predetermined eigenvalue threshold and updating it with *fmineig* is used for discarding detection of spurious features with low λ_1 after *fn,* maximum number of features, is increased in subsequent frames.

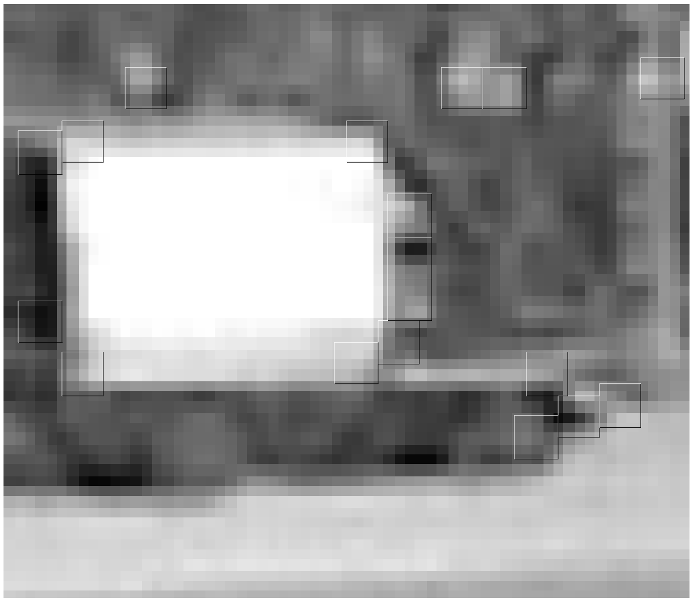


Figure 7- Detected features from the initial target window with parameters *fn*=2/400 (2 features per 400 pixels), *fspan*=3 pixels, and *fpatchsize*=5x5. Note that most detections are target originated.

Figure 7 shows the patches (features) detected in the initial target window in Figure 4 with the parameters used in Figure 6-c. A lorry is given as an example here, which is also used to illustrate the tracking algorithm explained in Chapter 4.

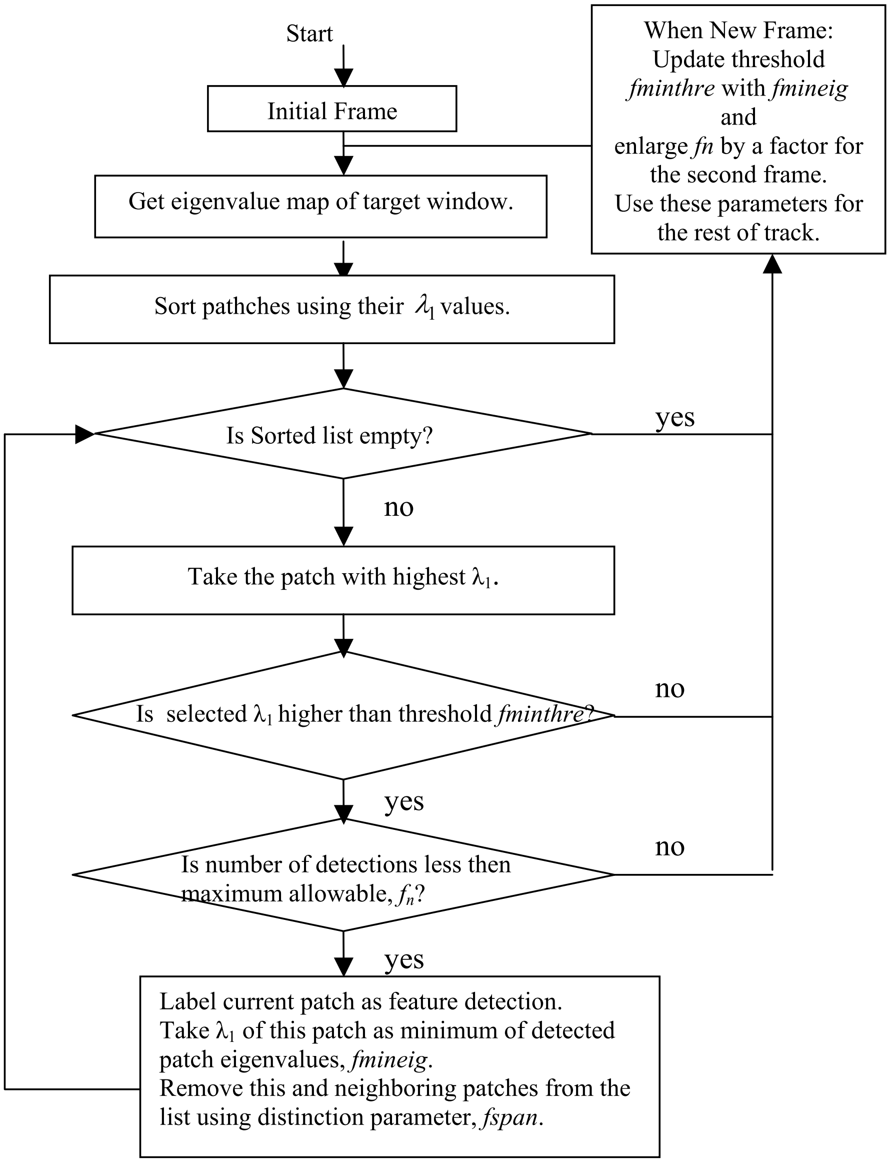

Start
Initial Frame
Get eigenvalue map of target window.
Sort pathches using their λ_1 values.
Is Sorted list empty?
yes
no
Take the patch with highest λ_1.
Is selected λ_1 higher than threshold *fminthre*?
no
yes
Is number of detections less then maximum allowable, f_n?
no
yes
Label current patch as feature detection.
Take λ_1 of this patch as minimum of detected patch eigenvalues, *fmineig*.
Remove this and neighboring patches from the list using distinction parameter, *fspan*.
When New Frame:
Update threshold *fminthre* with *fmineig* and enlarge *fn* by a factor for the second frame.
Use these parameters for the rest of track.

Figure 8- Flowchart illustration of the feature detection method.

2.5. Summary

In this chapter some video and image processing techniques are briefly mentioned. The emphasize of the chapter was on the corner feature detection. Since corners do not have aperture problem and are reliable they are already widely used for “motion from structure” and “computer vision” areas. A corner feature detection strategy is described which will be the pixel processing block of the tracker developed in Chapter 4.

CHAPTER 3

INFORMATION PROCESSING APPROACHES IN TRACKING

3.1. Introduction

This chapter is dedicated to information processing techniques in target tracking. Previous chapter investigated the information gathering from raw image (pixel) data. Other sensor types such as radar and sonar is not discussed in that chapter. But this chapter is more general and the tools and approaches explained here are applicable for different types of tracking problems. After measurement information is supplied to tracking unit, the treatment of it shows parallelism in many applications.

A universal tracking method that can be used in every situation is not possible. If there is only one target in the region of interest, the measurements taken can either be target or clutter originated. If there are multiple targets, measurement origin ambiguity increases because this time there is an additional uncertainty about which target the measurement is related to. Methods mainly differ from each other depending on how they resolve this ambiguity and how much accuracy is required. Gating and validation of measurements is necessary for all applications, which will be discussed here. Similarly, related topics, such as association problem and clutter modeling are also explained in this chapter. Use of standard filter, advanced ways for tracking single and multiple targets in clutter, centroid and formation tracking for groups of targets are the topics that are covered with examples.

3.2. Validation Region and Association Problem

To avoid searching for the signal in the entire measurement space, a multi-dimensional gate should be defined. There is a certain detection probability of a signal originated from the target in this gate. This gate is also named as validation or association region.

If the true measurement conditioned on the past is normally distributed with its probability density function given by

$$p\left[z(k+1)\,|\,Z^k\right] = \mathrm{N}\left[\mathrm{z(k+1)}; \hat{\mathrm{z}}(k+1\,|\,k); S(k+1\,|\,k)\right] \tag{3.1}$$

then the true measurement will be in the following region

$$\mathrm{V}(k+1,\gamma) = \left\{ z : \left[z - \hat{z}(k+1\,|\,k) \right]^T S(k+1)^{-1} \left[z - \hat{z}(k+1\,|\,k) \right] < \gamma \right\} \tag{3.2}$$

with the probability determined by γ. The term in 3.2 is chi square distributed with degree of freedom equal to n_z, the dimension of z. Therefore as a design criteria, the probability of true measurement being in the validation region must be decided first. γ increases if this probability is increased and implies that more region is covered to increase the probability of finding target originated measurement in the validation region.

The volume of validation region is related to S and γ as given below

$$\mathrm{Vol}(k+1) = c_{n_z} \left| \gamma S(k+1) \right|^{1/2} \tag{3.3}$$

where c_{n_z} is the volume of the unit hypersphere of n_z dimension.

Validation region simplifies the association problem resulted from measurement origin uncertainty. This simplification in association logic is a trade off because by determining a validation gate, target originated measurements outside the gate are ignored.

3.3. Clutter Model and Detection Probability

Before discussing the tracking algorithms a brief explanation on clutter modeling is informative. A detection occurs in a cell of the sensor if the output of the cell is above a certain threshold. Detection can sometimes happen when there is no target due to sensor noise, background noise or an unknown energy source. Such detections can be denoted as false alarms or clutter. To model this situation we can make the assumption that events of detection in each cell are independent of each other and the probability of a false alarm, p_{FA}, is constant for each cell. Then the probability mass function of the number of false alarms in the N cells having a volume V is given by the binomial (Bernoulli) distribution

$$P\{n_{FA} = m\} = \mu_{FA}(m) = \binom{N}{m} p_{FA}^{m}(1 - p_{FA})^{N-m} \tag{3.4}$$

The density of the false alarms is

$$\lambda = \frac{E[n_{FA}]}{V} = \frac{Np_{FA}}{V}. \tag{3.5}$$

If

$p_{FA} << 1$

and N is large enough so that Np_{FA} is of the order of 1 or larger, 3.4 is approximated by a Poisson distribution [15]. This approximation is

$$\mu_{FA}(m) = e^{-Np_{FA}} \frac{(Np_{FA})^{m}}{m!} \tag{3.6}$$

Since p_{FA} is a very small quantity in most cases, this approximation is reasonable. The independency assumption of false alarms with a space and time independent constant p_{FA} is accepted for most of the tracking algorithms. Similar to false alarm probability, detection probability in each cell, denoted by p_D, is assumed to be independent, non-unity and equal for each cell. These assumptions are again trade offs but help to derive simple formulations for most of the Bayesian tracking algorithms.

3.4. Assignment Problem

The assignment problem became more relevant to tracking problems recently. Therefore study on this subject increased and new techniques are developed as a result. For comprehensiveness, a brief explanation and overview of assignment problem is given in this section.

Assume that there is an association conflict between tracks and measurements as given in Figure 9. For example, in this figure O_5 may be observation (measurement) of target 1,2 or 3. It can be a false alarm, either.

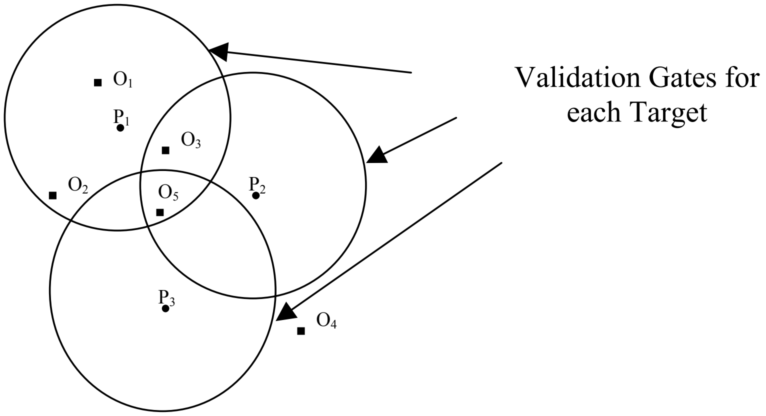


Figure 9- An association conflict situation. O_i is the ith observation, P_i is the initial prediction for the ith target before current scan (frame) observations.

The general assignment problem is modeled as a discrete optimization problem as follows:

Given the matrix elements a_{ij},

Find $X = \{x_{ij}\}$ such that $C = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_{ij}$ is minimized subject to:

$$\sum_{i} x_{ij} = 1 \ \forall j$$

$$\sum_{j} x_{ij} = 1 \ \forall i$$

where $x_{ij} \in \{0,1\}$.

Considering the definition and constraints above, an assignment matrix for Figure 9 can be generated as:

$$\begin{array}{lcccccccc}
 & \textbf{N/F1} & \textbf{N/F2} & \textbf{N/F3} & \textbf{N/F4} & \textbf{N/F5} & \textbf{T1} & \textbf{T2} & \textbf{T3} \\
\textbf{O1} & a_{11} & - & - & - & - & a_{16} & - & - \\
\textbf{O2} & - & a_{22} & - & - & - & a_{26} & - & - \\
\textbf{O3} & - & - & a_{33} & - & - & a_{36} & a_{37} & - \\
\textbf{O4} & - & - & - & a_{44} & - & - & - & - \\
\textbf{O5} & - & - & - & - & a_{55} & a_{56} & a_{57} & a_{53} \\
\textbf{ND1} & 0 & 0 & 0 & 0 & 0 & a_{66} & - & - \\
\textbf{ND2} & 0 & 0 & 0 & 0 & 0 & - & a_{77} & - \\
\textbf{ND3} & 0 & 0 & 0 & 0 & 0 & - & - & a_{88}
\end{array}$$

where rows are for the observations and columns are for targets. Both of them are extended. The extension in columns is to account for false alarms and the extension in rows are to account for undetected targets. *N/Fi* represents new or false alarm for the ith observation and *NDi* is to account for no detection of target i. *Ti* is the ith target, *Oi* is the ith observation and *"-"* represents impossible assignments. For example considering Figure 9, first observation *O1* can be a false alarm or can be originated from target 1. Therefore in any assignment *X,* either x_{11} or x_{16} can be 1 and all others are 0 for the first row. a_{16} is the cost of relating observation 1 to target 1. This value can be found considering the statistical distance of observation 1 to target 1 and the detection probability of target 1. a_{11} is the cost of observation 1 being a false alarm. Similarly a_{66} is the cost for miss of target 1. The inclusion of detection and false alarm costs in the assignment matrix give a way to find the best assignment with an integer programming method. It is also possible to find best m assignments with a proper partitioning of assignment alternatives [19,20]. This can be used in situations where ignoring less likely assignment alternatives is required.

An earlier assignment algorithm, the Hungarian method, is a primal-dual method that uses the dual assignment problem and preferred to simplex method for assignment problems. It is only applicable to square assignment matrices. Munkres algorithm, Jonker-Volgenant relaxation techniques and *auction* methods are some efficient alternatives that are preferred today [17]. Advanced multi-dimensional assignment algorithms are also used for associating observations to targets over multiple-scans [18].

3.5. Single Target in Clutter

This and following section explains some tracking methods for different tracking problems. The aim is to build an insight for the specific tracking problem in this thesis and the tracking method proposed. Particularly, in this section, tracking approaches for one target in clutter case will be investigated.

To handle the tracking problem for single target in clutter, there are some well-known techniques. First, assuming that the state equation of the target is known and a measurement equation relating measurements to state variables with an additive noise is present, a linear Gaussian system equation can be modeled as

$$x(k+1) = F(k)x(k) + \upsilon(k) \tag{3.7}$$

$$z(k) = H(k)x(k) + w(k) \tag{3.8}$$

3.7 is the state model and 3.8 is the measurement model. Assuming mean and covariance of initial state is known, at each new scan (sampling), a validation gate region is set up around the predicted measurement positions. There may be more than one measurement in this region and the usual assumption is that at most one of them is target originated and others are false alarms. Below we present some tracking methods applicable to the model and assumptions described so far.

Non-Bayesian techniques are the neighboring and track-split approaches. The Nearest Neighbor Standard Filter (NNSF) method selects statistically nearest measurement to the predicted measurement. If Strongest Neighbor Standard Filter (SNSF) is used, the measurement with the strongest attribute is selected. As an example this attribute is the signal intensity for radar and sonar applications. Update of the target state for the Kalman Filter is done with this selected measurement as if it were the correct one. These approaches resolve the association ambiguity and lower the probability of using incorrect measurement. But when the incorrect measurement is used, this action is irreversible causing in an “overconfidence” which can lead to loss of target even in a moderate clutter density.

Another approach is to split the track if there are more than one measurement at the validation region and applying standard Kalman Filtering to each generated branch. The number of branches increases exponentially and the unlikely ones should be pruned using a statistical test to keep number of splits manageable.

Denote θ_i^k as the i$^{\text{th}}$ branch which has a sequence of measurements to the current time *k*. A likelihood function defined for this branch is

$$\Lambda(\theta_i^{\,k}) = p\left[z(1),....,z(k) \mid \theta_i^{\,k})\right] = \prod_{j=1}^{k} p\left[z(j) \mid Z^{j-1}, \theta_i^k\right] \tag{3.9}$$

where *z(1)…z(k)* are the measurements selected by the branch, and Z^j is used to refer all the measurements up to time j.

This likelihood value is used to keep or prune decisions depending on how well the measurements fit the target model. Its power is limited against false tracks and computation and memory requirements are high because of the exponentially growing number of branches.

An alternative approach can be a Bayesian one. The probabilistic data association filter (PDAF) is a Bayesian technique in which the past information at current time k is summarized with

$$p\left[x(k) \mid Z^{k-1})\right] = N\left[x(k); \hat{x}(k \mid k-1), P(k \mid k-1)\right] \tag{3.10}$$

Using this information a validation gate is generated and every measurement in validation gate region generates hypothesis. Each hypothesis is a possible association event. Only one of these events is true but the probabilities of each event can be calculated. Denote association (event, hypothesis) for *ith* measurement with $\theta_i(k)$ and its probability with $\beta_i(k)$. Let Z^k be all the measurements up to time k and *Z (k)* be the latest set of data. Then

$$\beta_i(k) = (1/c)p\left[Z(k) \mid \theta_i(k), m(k), Z^{k-1}\right] P\left\{\theta_i(k) \mid m(k), Z^{k-1}\right\} \tag{3.11}$$

where c is the normalization term. Using Poisson clutter and some derivations 3.11 becomes

$$\beta_i(k) = \begin{cases} \dfrac{e_i}{b + \sum_{j=1}^{m(k)} e_j} ....i = 1,..,m(k). \\ \dfrac{b}{b + \sum_{j=1}^{m(k)} e_j} ....i = 0 \end{cases} \tag{3.12}$$

where

$$e_i \overset{\Delta}{=} e^{-(1/2)\nu_i(k)' S(k)^{-1}\nu_i(k)} \tag{3.13}$$

$$b \overset{\Delta}{=} \lambda \left|2\pi S(k)\right|^{1/2} \frac{1-P_D P_G}{P_D} \tag{3.14}$$

and $\nu_i(k)$ is the innovation for i^{th} event, *S(k)* is the innovation covariance, P_G is a normalization factor for validation region, P_D is detection probability and λ is the spatial density for Poisson clutter model and *m(k)* is the number of measurements at scan k.

Define

$$\hat{x}_i(k \mid k) \overset{\Delta}{=} E\left[x(k) \mid \theta_i(k), Z^k\right] \tag{3.15}$$

as the state estimate assuming i^{th} measurement is correct. The state estimate is updated with all the validated measurements weighted by their probabilities of having originated from the target,

$$\hat{x}(k \mid k) = E\left[x(k) \mid Z^k\right] = \sum_{i=1}^{m(k)} E\left[x(k) \mid \theta_i(k), Z^k\right] P\left\{\theta_i(k) \mid Z^k\right\}$$

$$= \sum_{i=1}^{m(k)} \hat{x}_i(k \mid k)\beta_i(k) \tag{3.16}$$

Covariance update is done similarly but includes an extra positive semi definite term accounting for measurement uncertainty. PDA algorithm is powerful in clutter and almost simple as standard Kalman Filter. Therefore it is a popular tracking algorithm for single target in clutter.

A more accurate but complex tracker is possible if decomposition of the state estimate in 3.16 is done using all compositions of measurements from the initial to the current time rather than only the latest set of measurements. This is the optimal approach but computation and memory requirements are too high to implement practically. Sub-optimal methods can be generated by using pruning strategies.

3.6. Multiple Targets in Clutter

When there are multiple targets, measurement uncertainty increases. Even when the measurement in a validation region is target originated, it is uncertain that to which target it belongs if it is in the intersection of validation regions of multiple targets. The associations of measurements to targets should be done simultaneously to avoid associating one measurement to more than one target so that feasible hypothesis can be generated.

Joint PDA, (JPDA) is an extended version of PDA, developed for multiple targets. This approach uses the latest set of measurements for state update of targets as in PDA. Using all the measurements from initial to present time to update targets generates the optimal algorithm, Multiple Hypothesis Tracker (MHT).

In JPDA, each feasible joint association event is considered as hypothesis for the present time. A joint association event is feasible if at most one measurement is associated to a target and at most one target is associated to a measurement. Let $\theta^i(k)$ denote i^{th} hypothesis (joint association event), $\theta_{jt}(k)$ denote the event that measurement j originates from target t, $\beta_{jt}(k)$ denote the probability of $\theta_{jt}(k)$. Note that $\beta_{jt}(k)$ is the sum of probabilities of hypotheses in which, measurement j is associated to track t. That is,

$$\beta_{jt}(k) \overset{\Delta}{=} P\{\theta_{jt} \mid Z^k\} = \sum_{i:\theta_{jt} \in \theta^i} P\{\theta^i \mid Z^k\} \tag{3.17}$$

To make the individual state update of a target t, $\beta_{jt}(k)$ values are used as weightings similar to PDA where j varies from 1 to number of measurements at present time k. A limitation of JPDA is that the number of targets is known and constant during the tracking.

A more complex alternative, MHT, considers the associations of sequences of measurements up to present. By using them, MHT evaluates all the association hypothesis. The exponentially growing complexity requires unlimited computation and memory sources. To cope with this problem clustering, pruning of low probability hypothesis and tracks, and merging of similar tracks is inevitable. There are also some parallel processing approaches developed for MHT as well [16,17]. New tracks can also be initiated during tracking with MHT.

The details of the techniques explained so far can be found in comprehensive references [8,9,11,12]. Next section will be on group tracking. Group tracking problem can be solved by considering group centroid as a single target in clutter. It is also possible to handle the problem with a multiple-target approach where each group member is a target. The association logic can possibly be made by nearest neighborhood or some possible associations can be considered in the tracker by hypothesis generation as in JPDA and MHT. Therefore the cases and approaches explained so far are strongly related to the group tracking explained in next section.

3.7. Group Tracking

3.7.1. Introduction

This section discusses the approaches for group tracking. It is mostly based on the past studies explained in Chapter 11 of [11]. A group can be defined as a set of targets traveling in the same direction with the distances between group members being less then the distances between other groups or targets. The conceptual advantages of group tracking, when compared to standard methods having individual tracks on all targets, can be seen as follows. First, fewer track files are required for group tracking because a single group track replaces the individual target tracks. Air to ground tracking situations, where the tracking of convoys containing many vehicles all traveling together are good examples where this point is important. Another point is the use of group tracking provides stability and additional smoothing where miscorrelation is inevitable for closely spaced targets.

3.7.2. Issues in Group Tracking

Group tracking also requires the same phases of more conventional trackers, initialization, confirmation and deletion. Similarly, filtering, prediction and gating methods are also required. Despite the apparent simplicity of the group tracking concept, it conceals difficulties unique to this approach.

One difficulty is on recognizing the change in the group size due to splitting or merging of groups. A related problem is to estimate the number of targets in the group. Missing observations due to non-unity detection probability can lead to instability in the estimates of number of targets in the group, group position centroid and group velocity. Keeping individual positions is an alternative approach used with estimated group velocity to increase accuracy.

Next subsections outline two basic approaches and their important differences. These are the centroid and formation group tracking approaches.

3.7.3. Centroid Group Tracking

In this method centroid of the target is tracked. Therefore measurement centroid is used for track update. Gating on the predicted centroid may be done considering a dispersion factor. Frazier and Scott [11] define a generalized residual covariance as

$$S_G = \hat{S}_D + R_G + HPH^T \tag{3.1}$$

The covariance defined here includes additional term, $\hat{S}_D$ for the estimate of the dispersion of the group. R_G is the group measurement covariance and HPH^T represents the standard contribution due to target dynamics, and is defined by one-step prediction covariance matrix $P(k+1|k)$ and the measurement matrix H.

Details for choices and calculations of R_G and $\hat{S}_D$ can be found in [11]. Using S_G for gating, the measurements are checked to see if one or more satisfy gating test. A feasible approach may be selecting the best measurement (closest to predicted group centroid) as seed and form the group measurement as:

- Any measurement that is added to the group observation should satisfy distance criteria with respect to its nearest neighbor or group centroid
- The maximum number of observations in a group may be limited.

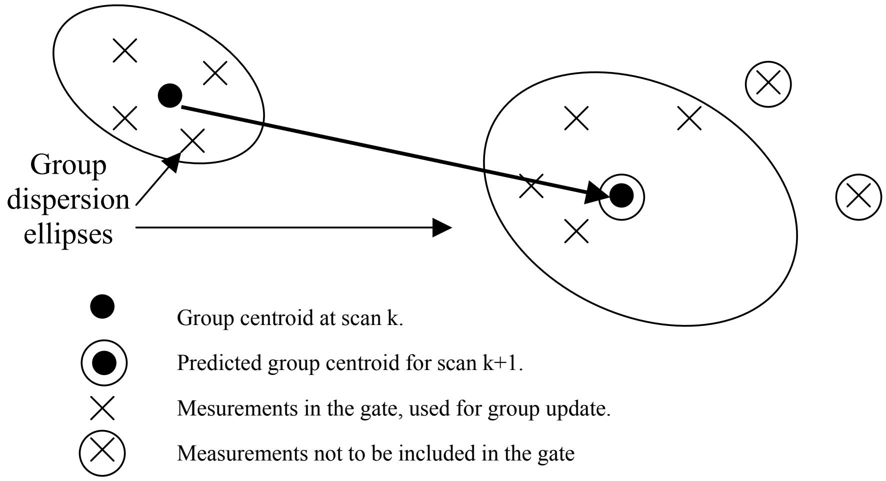

Group
dispersion
ellipses
Group centroid at scan k.
Predicted group centroid for scan k+1.
Mesurements in the gate, used for group update.
Measurements not to be included in the gate

Figure 10- Illustration of centroid group tracking.

Once a group measurement is correlated with a group track, the measured group centroid is combined with the group track predicted position in a standard Kalman filter update step to form the filtered and the new predicted position estimates.

3.7.4. Formation Group Tracking

Formation group tracking is an approach such that accounting for individual targets is maintained within a group tracking structure. Main advantage of this is minimization of the adverse effects of false and missing measurements. This is illustrated by an example given in Figure 11. Assume that observations correspond to true positions for simplicity. T_{ij} refers to the i^{th} target at scan j. C_j is the measured centroid calculated with available measurements, and C_{jtrue} is the true centroid of the group for scan *j*. This figure illustrates the situation where targets 3 and 4 in second scan and targets 1 and 2 in the third scan are not detected by the sensor during the track, because they enter into a masked region. Masked region may be an unknown object, a cloud when tracking planes for example. It is seen that, if group centroid is calculated with available measurements as in centroid group tracking method, it will be very erratic. This results in erratic velocity estimation also. However formation group tracking keeps individual target position estimates and use them in centroid calculation to handle this problem.

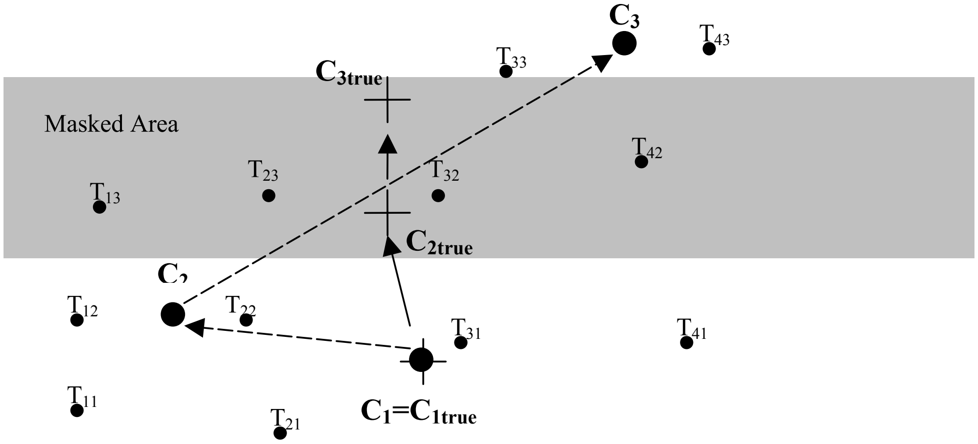


Figure 11- Example for erratic centroid calculation during target mask.

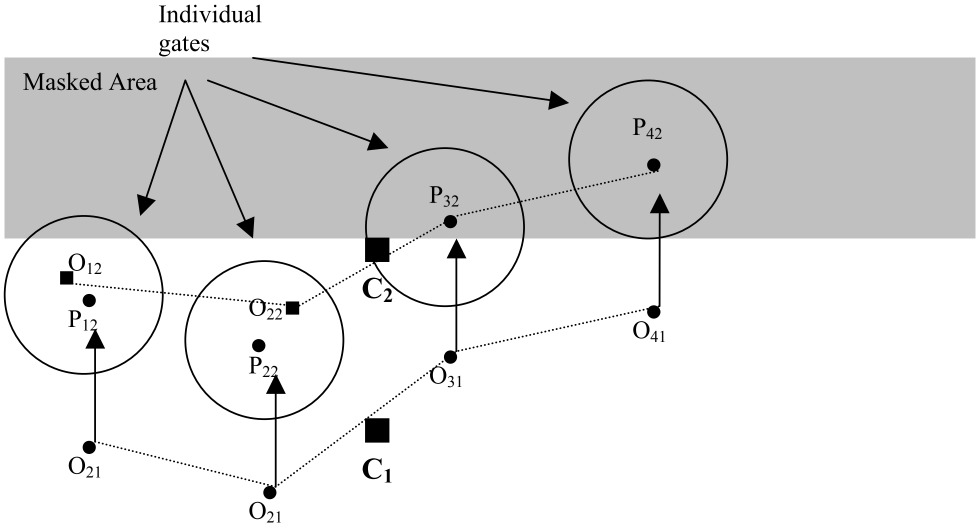


Figure 12- Illustration of formation group track update method

In Figure 12 how predictions are used for undetected targets is shown. O_{ij} refers observation of the i^{th} target at scan j and P_{ij} refers to the prediction of i^{th} target at scan j. Group centroid, C_1, calculation at scan 1 is done using observations O_{11}, O_{21}, O_{31}, and O_{41}, where second scan group centroid, C_2, is calculated using observations O_{12}, O_{22} and predictions P_{32}, $P_{42.}$ It is important to note that no position smoothing is applied to individual targets when their observations are available at the present scan although their predictions are maintained. Observations are assumed to be the best current estimates of the target state and these estimates are used for group centroid and velocity estimation. Therefore formations are represented by groups of target position estimates, which are computed directly by unsmoothed measurements and the group velocity. Since the tracking is based on group centroid and group velocity, accuracy for individual targets is ignored. But centroid is still a smoothed value because all the detected targets and the predicted positions for undetected ones are included in its calculation. Although this method seems relatively simple in principle, a complex logic is still required to handle ambiguities. A nearest neighbor approach is preferably applied to associate new measurements with the current tracks.

3.7.5. Comparison and Possible Extensions

Blackman, [11], notes that "*Although the applications seem apparent, the development or reporting of methods for group tracking has lagged that of other techniques*" (1986). He adds that; "*despite the potential advantages of the group tracking approach, the practical problems such as recognizing groups, incorporating new members to the group and splitting and merging groups have apparently discouraged the widespread use of group tracking*"(1999). The centroid method is simpler and resembles to single target Nearest Neighborhood (NN) tracking approach. Formation tracking also uses centroid velocity and position therefore resembles to single target case, either. However the estimation of centroid is improved by using predicted positions of undetected members. A more advanced way of tracking is possible keeping the global displacement of targets with a complex association and update scheme. [10] describes a group model and shows how an MHT based algorithm can be implemented. Our approach is a PDA/JPDA like method which is better than MHT considering computation and memory resources and more accurate than NN like association with formation tracking. Furthermore, the applicability of group tracking approach for visual objects using their features increased motivation for this method.

3.8. Summary and Comments

This chapter covered basic techniques for evaluation of sensor information for tracking. They are based on statistical tools and linear estimation filters but differ in the depth they use the information. Techniques depending on hypothesis generation are more accurate but exponential growth of hypothesis for optimal tracking is inevitable. Pruning the hypothesis by considering their likelihood or summarizing history to single information are some ways to confront this problem. Real time constraints and sharing processor sources for raw data processing especially in image applications makes the problem more severe. Simple association logic and filtering are therefore practical solutions for most cases.

CHAPTER 4

TRACKING VISUAL TARGETS BY THEIR FEATURES USING GROUP MOTION AND PDA APPROACH

4.1. Introduction

This chapter starts with a discussion on treating visual targets as multi-targets having group motion. Then an appropriate tracking algorithm for multi-targets having group motion will be described. As stated before, the aim of this study is tracking single visual targets. Therefore application of the method on single visual targets with multi-features will be made clear after the end of this chapter. Tracking simulation results of the algorithm with artificial and real image data and comments on results are in the following chapter.

4.2. Visual Targets as Multi-Targets with Group Motion

Tracking visual objects using frame sequences requires using both image processing techniques and tracking algorithms together. The structure of the latter mostly depends on the former. In Chapter 2, some pixel processing choices for visual target tracking are explained and how good features can be used in tracking is discussed.

Some basic assumptions can be made for visual targets. These assumptions effect the selection and application of the tracking algorithm. First of all, a visual target can be assumed quite rigid. That is, its projection on the image plane slightly changes in succeeding frames. In other words its formation is quite stable considering scanning frequency. This assumption is proper for tracking vehicles using imagers at 25 Hz or even lower frame frequency. This constraint can be named as rigidity constraint. [12] describes a feature tracking method with a projective camera model and rigid object assumption for collision prediction. Another proper assumption is on the stability of features in subsequent frames. [13] gives stability results of corner features depending on the detection methods.

Corner feature detection can reveal stable object related features at subsequent frames. Since the visual target object will most likely have multiple corner like features, each having dependency to the main dynamic of the target, application of group tracking techniques becomes reasonable.

4.3. Group Model For The Visual Target

In Chapter 3 group tracking approach is explained. Considering this explanation it can be said that, application of any group tracking algorithm requires that the targets obey to a group motion model. In this section a group motion model will be described based on the features of the visual targets. However, the model developed in this study is not unique to the current problem and can be used as a general group motion model for any other problem.

Considering the visual tracking problem, group motion and group dynamics will be used to refer main motion and main dynamics of the visual object, respectively. Each feature used in the model will be named as member of the group. These members form the tracks in the algorithm. First let the group (object) dynamic be modeled with a white noise acceleration model. Noting that it is possible to use lower or higher order models depending on the target structure, the group motion equation becomes:

$$\mathbf{m}_{k+1} = F\mathbf{m}_k + \Gamma\mathbf{v_k} + \mathbf{w_{k+1}} \tag{4.1}$$

where $\mathbf{m_k}$ is the state vector at scan k, $\mathbf{v_k}$ is the zero mean Gaussian state noise modeling the acceleration of the object and $\mathbf{w_k}$ is the rigidity noise vector which holds the additional noise components of position of each member. Each component of $\mathbf{w_k}$ is independent of the others so that each member has an additional independent displacement in each direction. This models the formation change of the visual target.

Let N_t be the number of members (features) tracked. Then $\mathbf{m_k}$ vector has components

$$\mathbf{m_k} = \begin{bmatrix} m_1(k) & m_1(k) & m_3(k) & m_4(k) & . & . & . & m_{Nt}(k) & g(k) & \overset{*}{g}(k) \end{bmatrix}^T$$

where component

$$\mathrm{m_n}(k) = \begin{bmatrix} m_{n_x}(k) & m_{n_y}(k) \end{bmatrix}^T$$

holds the nth member's position in x and y coordinates of image frame; component

$$\mathrm{g(k)} = \begin{bmatrix} g_x(k) & g_y(k) \end{bmatrix}^T$$

is the position of a reference point of the visual object; component

$$\overset{*}{\mathrm{g}}(k) = \begin{bmatrix} \overset{*}{g}_x(k) & \overset{*}{g}_y(k) \end{bmatrix}^T$$

is the velocity variable of the visual target that all the group members are assumed to have.

$$\mathbf{v_k} = \begin{bmatrix} v_x(k) & v_y(k) \end{bmatrix}^T$$

has zero mean Gaussian random variables v_x and v_y to model the acceleration of the object in x and y directions as random noise. v_x and v_y are independent of each other and of the components of $\mathbf{w_k}$.

$$\mathbf{w_k} = \begin{bmatrix} w_1(k) & w_2(k) & w_3(k) & w_4(k) & . & . & . & w_{Nt}(k) & 0_{2x1} & 0_{2x1} \end{bmatrix}^T$$

has components

$$\mathrm{w_n}(k) = \begin{bmatrix} w_{n_x}(k) & w_{n_y}(k) \end{bmatrix}^T,$$

modeling independent displacements of members in each direction and they are zero mean Gaussian random variables independent of each other and of components of $\mathbf{v_k}$.

Since our model is a white noise acceleration model,

$$F = \left[\begin{array}{c:c:c:c:c} I_2 & 0_2 & .\ . & 0_2 & I_2^T \\ \hdashline 0_2 & I_2 & .\ . & 0_2 & I_2^T \\ \hdashline 0_2 & 0_2 & .\ . & 0_2 & I_2^T \\ . & . & .\ . & . & . \\ . & . & .\ . & . & . \\ 0_2 & 0_2 & .\ . & 0_2 & I_2^T \\ \hdashline 0_2 & 0_2 & .\ . & I_2 & I_2^T \\ \hdashline 0_2 & 0_2 & .\ . & 0_2 & I_2 \end{array}\right]$$

and

$$\Gamma = \begin{bmatrix} I_2^{T^2/2} & I_2^{T^2/2} & I_2^{T^2/2} & I_2^{T^2/2} & . & . & . & I_2^{T^2/2} & I_2^T \end{bmatrix}^T$$

where

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ I_2^T = \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix},\ I_2^{T\ /2} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \end{bmatrix}, 0_{2x1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, 0_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Then equation 4.1 can be written more explicitly as

$$\begin{bmatrix} m_1(k+1) \\ m_2(k+1) \\ m_3(k+1) \\ m_4(k+1) \\ . \\ . \\ . \\ m_{Nt}(k+1) \\ g(k+1) \\ \overset{*}{g}(k+1) \end{bmatrix} = \left[\begin{array}{c|c|cc|c|c} I_2 & 0_2 & . & . & 0_2 & I_2^T \\ \hline 0_2 & I_2 & . & . & 0_2 & I_2^T \\ \hline 0_2 & 0_2 & . & . & 0_2 & I_2^T \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0_2 & 0_2 & . & . & 0_2 & I_2^T \\ \hline 0_2 & 0_2 & . & . & I_2 & I_2^T \\ \hline 0_2 & 0_2 & . & . & 0_2 & I_2 \end{array}\right] \begin{bmatrix} m_1(k) \\ m_2(k) \\ m_3(k) \\ m_4(k) \\ . \\ . \\ . \\ m_{Nt}(k) \\ g(k) \\ \overset{*}{g}(k) \end{bmatrix} + \begin{bmatrix} I_2^{T^2/2} \\ I_2^{T^2/2} \\ I_2^{T^2/2} \\ I_2^{T^2/2} \\ . \\ . \\ . \\ I_2^{T^2/2} \\ I_2^T \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} w_1(k) \\ w_2(k) \\ w_3(k) \\ w_4(k) \\ . \\ . \\ . \\ w_{Nt}(k) \\ 0_{2x1} \\ 0_{2x1} \end{bmatrix}.$$

The measurement can be modeled as:

$$\mathbf{z}_k = H\mathbf{m}_k + \mathbf{n} \tag{4.2}$$

where,

$$H = \left[\begin{array}{c|c|ccc|c|c} I_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hline 0_2 & I_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hline 0_2 & 0_2 & . & . & . & 0_2 & 0_2 \\ 0_2 & 0_2 & . & . & . & 0_2 & 0_2 \\ \hline 0_2 & 0_2 & 0_2 & 0_2 & I_2 & 0_2 & 0_2 \end{array}\right];$$

$$\mathbf{z}_k = \begin{bmatrix} z_1(k) & z_2(k) & z_3(k) & z_4(k) & . & . & . & z_{Nt}(k) \end{bmatrix}^T$$

and component

$$\mathrm{z_n}(k) = \begin{bmatrix} z_{n_x}(k) & z_{n_y}(k) \end{bmatrix}^T$$

holds the measurement for the nth member's position in x and y coordinates of image frame;

$$\mathbf{n_k} = \begin{bmatrix} n_1(k) & n_2(k) & n_3(k) & n_4(k) & . & . & . & n_{Nt}(k) \end{bmatrix}^T$$

has components

$$\mathrm{n_i}(k) = \begin{bmatrix} n_{i_x}(k) & n_{i_y}(k) \end{bmatrix}^T$$

modeling measurement noise of the ith member in each direction with zero mean Gaussian random variables independent of each other and of components of $\mathbf{v_k}$, and $\mathbf{w_k}$.

Then equation 4.2 can be written more explicitly as

$$\begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \\ . \\ . \\ . \\ z_{Nt}(k) \end{bmatrix} = \left[\begin{array}{c|c|ccc|c|c} I_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hline 0_2 & I_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hline 0_2 & 0_2 & . & . & . & 0_2 & 0_2 \\ 0_2 & 0_2 & . & . & . & 0_2 & 0_2 \\ \hline 0_2 & 0_2 & 0_2 & 0_2 & I_2 & 0_2 & 0_2 \end{array}\right] \begin{bmatrix} m_1(k) \\ m_2(k) \\ m_3(k) \\ m_4(k) \\ . \\ . \\ . \\ m_{Nt}(k) \\ g(k) \\ \overset{*}{g}(k) \end{bmatrix} + \begin{bmatrix} n_1(k) \\ n_2(k) \\ n_3(k) \\ n_4(k) \\ . \\ . \\ . \\ n_{Nt}(k) \end{bmatrix}$$

As stated before, 4.1 models the dynamics of the target with a state variable containing positions of each member, the reference point and velocity components of the group. The unique part of the dynamic for each member is the rigidity noise **w** which reflects the very reasonable situation that the formation of the group changes frame by frame since the target or the projection of the target on the image plane is not perfectly rigid.

Equation 4.2 shows the case where all measurements of each member being tracked are observed with a measurement noise vector **n.** In the following sections this point will be discussed in detail by explaining the partial observations of members (components of **z**).

4.4. Group Tracking With PDA using Visual Target’s Features

4.4.1. Approach

In Chapter 3 group tracking and data association approaches are discussed with their advantages. It is apparent that if tracking of targets having common dynamics is required, this property should be used in tracking. In our problem of single visual object tracking, if feature detection is applied to image frame, then the tracking algorithm is fed by observations having group motion. It is also clear that this kind of rich data directs to more accurate and advanced trackers.

In Chapter 3 the major problem of association ambiguity is discussed which is also present in our problem. Considering this problem, our approach will result in more accurate tracking than the centroid and formation group tracking approaches. As stated before, centroid group tracking lacks individual target (member) position estimates and although formation group tracking keeps individual member positions during the tracking, a simple nearest-neighbor type association is applied. The accuracy of tracking can be increased with hypothesis generation and data association. This is the improvement proposed in this study. The use of multiple hypothesis tracking is also possible if the increased computational complexity can be handled [10]. Data association will be used after each scan in our algorithm. Therefore there will be only one state estimate and covariance summarizing all the history, which is the characteristic of PDA and JPDA algorithms. Following sections explain the algorithm in more detail.

4.4.2. Generation of Feasible Hypothesis

Using the general approach and notation used for modeling multiple hypothesis conditions, we can formulate our problem as follows. First, we assume that currently we have a state estimate and state covariance matrix. With the measurement from current scan we can generate a validation matrix Ω as follows:

$$\Omega = \lfloor \omega_{jt} \rfloor \qquad \omega_{jt} \in \{0,1\}, \qquad j=1,\dots,N_O; \quad t=0,1,\dots,N_T$$

Binary elements of the validation matrix indicate whether measurement j is in the validation gate of member t. N_O is the number of current measurements (observations) and N_T is the number of features being tracked. $t=0$ is used to denote false alarms. Here is an example:

$$\Omega = \begin{array}{cccc} {}^{\quad t}_{j} & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 1 & 0 \\ 4 & 1 & 0 & 1 \end{array}$$

This example describes the case given in Figure 13 where O_i stands for i^{th} observation and P_i is used to denote the prediction of the i^{th} target.

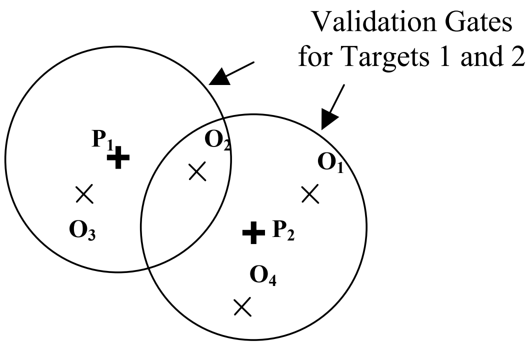


Figure 13- An example case for measurements and member predictions with validation gates.

Note that observation O_2 can originate from target (member) 1, target 2 or it may even be a false detection. Feasible partitioning of measurements to group members can be made with two assumptions. Firstly, measurement can be a false alarm or can only originate from a single group member being tracked. Second assumption states that there cannot be more than one measurement originating from a currently tracked member. These assumptions for feasible events restrict validation matrix to have only one nonzero element in each row and column except for the first column because any number of measurement can be false. Name any feasible association event as hypothesis, θ_i.

Let $\{\theta_i\}$ denote the set of hypothesis where i=1,2…,$N_{\theta;}$ and let N_θ be the number of hypothesis at current scan. Any hypothesis θ_i is represented by its event matrix

$$\hat{\Omega}(\theta_i) = \left[\hat{\omega}_{jt}(\theta_i) \right],$$

which is a binary matrix derived from Ω by considering the choices related with decisions of this hypothesis.

The following is an example of $\hat{\Omega}(\theta_i)$:

$$\Omega(\theta_i) = \begin{array}{c|ccc} j \backslash t & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 4 & 1 & 0 & 0 \end{array}$$

and

$$\hat{\omega}_{jt}(\theta_i) \overset{\Delta}{=} \begin{cases} 1 \text{ if } j^{th} \text{ obsevation is associated with } t^{th} \text{ target according to the hypothesis } \theta_i. \\ 0 \text{ otherwise.} \end{cases}$$

.

where $\hat{\omega}_{jt}(\theta_i)=1$ for (j,t) pairs of (1,0),(2,2),(3,1),(4,0) for this example.

We can also define new indicators for convenience.

These are the measurement association indicator,

$$\tau_j(\theta_i) \overset{\Delta}{=} \sum_{t=1}^{N_T} \hat{\omega}_{jt}(\theta_i),$$

and number of false measurements in event θ_i.

$$\phi(\theta_i) \overset{\Delta}{=} \sum_{j=1}^{N_o} \left[1 - \tau_j(\theta_i)\right]$$

Note that $\tau_j(\theta_i) = 0$ if no target is associated to the measurement j.

4.4.3. Estimation under Partial Observation

Depending on the hypothesis θ_i, some members of measurement set *Z(k)* at scan k are assigned as the components of the measurement vector **z,** which is defined in 4.2. This means that even though equation 4.2 defines the measurement vector $\boldsymbol{z}$ with N_T components from the formulation, it is obvious that N_T is frame and hypothesis dependent. Some targets may not be observed. Therefore for every hypothesis we define a measurement vector $\boldsymbol{z}_i$. *H* matrix should also be modified to H_i matrix accordingly.

For example if we are tracking 4 features of visual target we can have at most 4 measurements for each of these feature tracks in measurement vector as

$$\mathbf{z} = [z_1 \quad z_2 \quad z_3 \quad z_4]$$

Let the current hypothesis θ_i has an event matrix as

$$\Omega(\theta_i)=\begin{array}{c|ccccc} {}_{j}\,{}^{t} & 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 \end{array}$$

In this feasible hypothesis, feature track 2 and 4 are associated with measurements 1 and 2, respectively. Other feature tracks are undetected (in most of tracking problems detection probabilities of tracks are non-unity) at the current scan and measurement 3 is a false alarm. So we define z_i as

$$\mathbf{z}_i \overset{\Delta}{=} [z_2 \quad z_4].$$

This means that according to hypothesis θ_i the measurement vector has components for tracks 2 and 4. So the H matrix relating the measurements to system should also contain the rows for the tracks 2 and 4. Removing rows 1 and 2 for track 1 and rows 5 and 6 for track 3 from H matrix, we get,

$$H=\left[\begin{array}{cccc:c:c} I_2 & 0_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hdashline 0_2 & I_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hdashline 0_2 & 0_2 & I_2 & 0_2 & 0_2 & 0_2 \\ \hdashline 0_2 & 0_2 & 0_2 & I_2 & 0_2 & 0_2 \end{array}\right] \Rightarrow H_i=\left[\begin{array}{cccc:c:c} 0_2 & I_2 & 0_2 & 0_2 & 0_2 & 0_2 \\ \hdashline 0_2 & 0_2 & 0_2 & I_2 & 0_2 & 0_2 \end{array}\right]$$

The modifications applied to z and H with the hypothesis can be explained as follows:

Given the hypothesis $\theta_{i,}$ denote $\delta_{\theta i}$ as the target detection indicator vector for this hypothesis. The size of this binary valued vector is N_T. $\delta_{\theta i}(n)$ is 1 if *n-th* member being tracked is associated with a measurement and 0 otherwise for $n=1,..,N_T$. z_i is formed first by assigning the appropriate measurements to members using $\Omega(\theta_i)$ and marking nth pair of z, $z(2n)$ and $z((2n)-1)$ for removal if $\delta_{\theta i}(n)=0$. Similarly row 2n and (2n –1) of H will be marked for removal if $\delta_{\theta i}(n)=0$. Remove is done in pairs because each track is represented with two components, one for x and one for y coordinates. Removing all the components marked will result in z_i and H_i.

These hypotheses dependent modifications effect hypothesis dependent updates. Because of the Markov property of the process, Kalman filter requires only state estimate and state covariance of the previous step for next step calculations. Thus, state update is made by the modified measurement equation. This is, switching to another measurement model and setting a new Kalman Filter any time measurement vector structure changes. By this switching we always use the appropriate model for the current Markov system and we are still making the best one step estimate in MMSE sense.

4.4.4. Evaluation of Hypothesis Probabilities

Using the validation matrix we are able to generate hypothesis each of which is a candidate for the measurement origin uncertainty. If we can use a metric to evaluate the probability of each hypothesis we will be able to make estimation on the current state of group members, $\hat{m}(k\,|\,k)$.

Let

$$\hat{\mathbf{m}}^{i}(k\,|\,k) \overset{\Delta}{=} E\left[\mathbf{m}(k)\,|\,\theta_i(k), Z^k\right]$$

be the state estimate assuming ith hypothesis is correct. Then,

$$\hat{\mathbf{m}}(k\,|\,k) = E\left[\mathbf{m}(k)\,|\,Z^k\right] = \sum_{i=1}^{N_\theta} E\left[\mathbf{m}(k)\,|\,\theta_i(k), Z^k\right] P\left\{\theta_i(k)\,|\,Z^k\right\} = \sum_{i=1}^{N_\theta} \hat{\mathbf{m}}^{i}(k\,|\,k) P\left\{\theta_i(k)\,|\,Z^k\right\} \quad (4.3)$$

here, $Z^k=\{Z(i)\}$ for $i=0,1,...,k;$ is all the measurements up to current scan k.

The second term in equation 4.3 is the conditional hypothesis probability, and we apply the Bayes' formula for this term:

$$\begin{aligned}
\beta_i(k) &\overset{\Delta}{=} P\left\{\theta_i(k)\,|\,Z^k\right\} = P\left\{\theta_i(k)\,|\,Z(k), N_o(k), Z^{k-1}\right\} \\
&= (1/c)\,p\left[Z(k)\,|\,\theta_i(k), N_o(k), Z^{k-1}\right] P\left\{\theta_i(k)\,|\,Z^{k-1}, N_o(k)\right\} \\
&= (1/c)\,p\left[Z(k)\,|\,\theta_i(k), N_o(k), Z^{k-1}\right] P\left\{\theta_i(k)\,|\,N_o(k)\right\} \qquad (4.4)
\end{aligned}$$

here the first term c is the normalization coefficient and Z^{k-1} is irrelevant (unnecessary) information for the third term. Therefore it is omitted at the last line. The second term of 4.4 is the likelihood function of measurements.

The likelihood function of measurements is:

$$p\left[Z(k)\,|\,\theta_i(k), N_o(k), Z^{k-1}\right] = V^{-\phi(\theta_i)}\,p\left[\mathbf{z}_{\mathrm{i}}(k)\,|\,\theta_i(k), Z^{k-1}\right] \quad (4.5)$$

$$p\left[Z(k)\,|\,\theta_i(k), N_o(k), Z^{k-1}\right] = V^{-\phi(\theta_i)}\,N\left[\mathbf{z}_{\mathrm{i}}(k); \hat{\mathbf{z}}_{\mathrm{i}}(k\,|\,k-1), S_i(k)\right] \quad (4.6)$$

The measurements unassociated with a member (target) are assumed uniformly distributed in the surveillance volume V. This uniform distribution implies a likelihood V^{1}, as a product term in equation 4.5. Since total number of unassociated measurements is $\phi(\theta_i)$ we get the first term in equation 4.5. Likelihood of all the other measurements associated with targets is the second term in 4.5, which should be evaluated in a coupled manner because as stated with 4.1 and 4.2 there is one motion dynamic causing each measurement to be dependent to the others. This is calculated using the joint distribution of the detected tracks. This distribution is Gaussian, having mean and covariance calculated with Kalman Filter equations modified as described in previous subsection. Mean vector in 4.6 is the prediction of the filter and covariance is the innovation covariance found by this filter. Note that in JPDA, each measurement associated target has its own marginal distribution.

The last term in equation 4.4 is the prior probability of a joint association event. A hypothesis θ_i, implicitly hold the target detection indicator vector, $\delta_{\theta i}$, and number of unassociated measurements, $\phi(\theta_i)$. Then,

$$\begin{aligned} P\{\theta_i(k) \mid N_o(k)\} &= P\{\theta_i(k), \delta(\theta_i), \phi(\theta_i) \mid N_o(k)\} \\ &= P\{\theta_i(k) \mid \delta(\theta_i), \phi(\theta_i), N_o(k)\} P\{\delta(\theta_i), \phi(\theta_i) \mid N_o(k)\} \qquad (4.7) \end{aligned}$$

Now we will use some combinatorics. First, the number of permutations of N_O measurements taken as N_O-$\phi(\theta_i)$, gives the number of possible measurement to member track associations for the set of detected tracks. This is the number of different associations for detected tracks, $P^{N_o(k)}_{N_o(k)-\phi(\theta_i)}$. If we assume each of these equally likely, we get

$$P\{\theta_i(k) \mid \delta(\theta_i), \phi(\theta_i), N_o(k)\} = (P^{N_o(k)}_{N_o(k)-\phi(\theta_i)})^{-1} = \frac{\phi(\theta_i)!}{N_o(k)!} \qquad (4.8)$$

The last term in 4.7 is straightforward assuming $\delta_{\theta i}$ and $\phi(\theta_i)$ are independent;

$$P\{\delta(\theta_i), \phi(\theta_i) \mid N_o(k)\} = \prod (P_D^t)(1 - P_D^t)^{1-\delta(\theta_i, t)} \mu_F(\phi(\theta_i)) \qquad (4.9)$$

P_D is feature detection probability and $\mu_F(\phi(\theta_i)$ is the probability mass function for the number of false measurements.

4.4.5. State Estimation

Now we have hypothesis probabilities for each feasible hypothesis and the state estimate is made by weighting estimates of each hypothesis as given below:

$$\hat{m}^{i}(k \mid k) = \hat{m}(k \mid k-1) + W_i(k)\nu_i(k) \qquad \text{i=1,…,N}_{\theta,} \tag{4.10}$$

where the innovation for θ_i is,

$$\nu_i(k) = z_i(k) - \hat{z}_i(k \mid k-1)\,,$$

and $\hat{z}_i(k \mid k-1)$ is the modified version of prediction vector $\hat{z}(k \mid k-1)$ where the unassociated measurements are removed depending on the hypothesis as described in the section 4.4.3. The hypothesis gain W_i is:

$$W_i(k) = P(k \mid k-1)H_i(k)^T S_i(k)^{-1} \tag{4.11}$$

Similar to the case for $\hat{z}_i(k \mid k-1)$, $S_i(k) = H_i(k)P(k \mid k-1)H_i(k)^T + R_i(k+1)$ is the modified (reduced) version of the generic innovation covariance matrix $S(k)$, given by 4.17.

Combining 4.10 and 4.11 in 4.3 we get

$$\hat{m}(k \mid k) = \hat{m}(k \mid k-1) + \sum_{i=1}^{N_\theta} \beta_i(k)W_i(k)\nu_i(k) \tag{4.12}$$

and the covariance update is,

$$P(k \mid k) = E\left\{\left[m(k) - \hat{m}(k \mid k)\right]\left[m(k) - \hat{m}(k \mid k)\right]^T \mid Z^k\right\}$$

$$= \sum_{i=1}^{N_\theta} E\left\{\left[m(k) - \hat{m}(k \mid k)\right]\left[m(k) - \hat{m}(k \mid k)\right]^T \mid \theta_i(k), Z^k\right\}\beta_i(k)$$

$$= \sum_{i=0}^{N_\theta} \beta_i(k)P_i(k \mid k) + (\sum_{i=0}^{N_\theta} \beta_i(k)\hat{m}_i(k \mid k)\hat{m}_i(k \mid k)^T) - \hat{m}(k \mid k)\hat{m}(k \mid k)^T \tag{4.13}$$

After state update is completed hypothesis independent next scan prediction equations can be calculated as in a standard filter:

$$\hat{m}(k+1 \mid k) = F(k)\,\hat{m}(k \mid k) \tag{4.14}$$

$$\hat{z}(k+1 \mid k) = H(k+1)\,\hat{m}(k+1 \mid k) \tag{4.15}$$

$$P(k+1 \mid k) = F(k)P(k \mid k)F(k)^T + Q(k) \tag{4.16}$$

$$S(k+1) = H(k+1)P(k+1 \mid k)H(k+1)^T + R(k+1) \tag{4.17}$$

where

$$Q(k) = E\left[(\Gamma v(k) + w(k))(\Gamma v(k) + w(k))^T\right] = \Gamma U \Gamma^T + M \tag{4.18}$$

and U and M are covariance matrices for v and w respectively.

One cycle of the developed filter is explained above. This derivations describe how the state estimation and Joint Probabilistic Data association can be made for the group of targets.

4.5. Summary

This chapter first defined a group model for features of visual object. Then necessary derivations for state prediction and update are carried to develop a tracker which uses all neigborhod approach as the PDA filter, for measurement to target (member) association. Both this group tracking approach and its application to single visual target tracking can be seen as the unique sides of our method. History is also summarized to reduce computational complexity and memory requirements. Next chapter will be on implementation and simulations of the developed tracker.

CHAPTER 5

IMPLEMENTATION AND SIMULATIONS

5.1. Implementation Issues

A few notes for the implementation of the algorithm may help to comment on the simulation results given in the following sections. The corner detection method explained in Chapter 2 is used for feature detection. In initialization phase it is used to classify features originating from visual target and during the tracking phase it generates the measurement information required by the tracker.

5.1.1. Initialization

Since no model for the object is present in the initialization, a model for the target dynamic should be approximated to start the tracking phase. Group motion assumption can be used to detect the global motion of the target features during initialization. The initial target window selected by the operator is the only available data we have initially. But we already have the maximum velocity and acceleration constraint for the targets. This is given by the requirements defining the target limits that the tracker is expected to track and used for gate determination for the subsequent frame in initialization phase.

After the corners are detected for the initial target window, we get many target related features with some spurious ones. This is an assumption and valid if the target window is properly selected by the operator. Figure 7 of Chapter 2 reveals the target and clutter features for an example initial target window. Target window for second frame depends on the maximum target speed constraint we have, and it is wide because target can be moving in any direction. After we get detections from the second frame, with the use of motion information we can find an estimate for the group motion. Relative displacements of all initial frame to second frame feature associations are calculated. If the displacement value of any association is not within the range determined by maximum target velocity constraint then it is considered as impossible association. Feature matching is applied to only possible associations. Feature matching is made using the correlation equation (2.2) of Chapter 2. After correlation factor (resemblance values) for these associations are calculated, the ones with low correlations are discarded. Therefore the associations left satisfy both displacement and resemblance constraint.

After these filtering steps, we have possible individual feature to feature associations and each association has a displacement attribute with it. Figure 14 shows an example where symbols "+" and "x" are used for first and second frame features respectively. Associations are identified by the lines. If a target is present we expect the displacement attributes (line length in figure) have similar values for most of these possible associations. Then a histogram of displacements should have a peak around the group motion displacement. We use the fact that even if there can be unfiltered spurious features within the possible associations, these should have random displacements so that the peak is only related to group motion. Using this peak displacement of the histogram, we can have an estimate on the group motion velocity. Figure 15 shows the associations left after a group displacement estimate. Note that the association on the bottom left is discarded and the appropriate association is selected for the upper left. The features with associations on the second frame can be taken as members of the tracker. (they are the (x) signs which are connected to (+) signs in Figure 15).

More accurate estimation can be possible using more frames for initialization and using polynomial fitting methods. But if the number of frames before a model is set is high, search region is increased in every direction during the initialization. Early but less accurate initialization can possibly be compensated by the convergence of the tracker to true state variables.

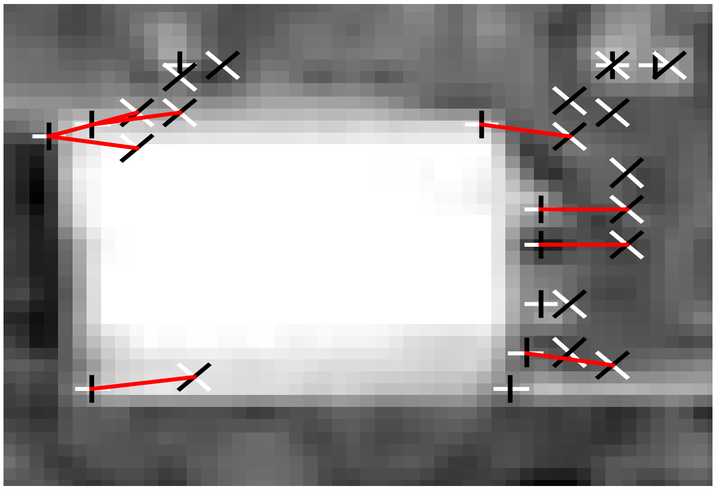


Figure 14- Possible associations using feature matching and initial constraints. (Second frame detections ("x") are superposed on initial frame)

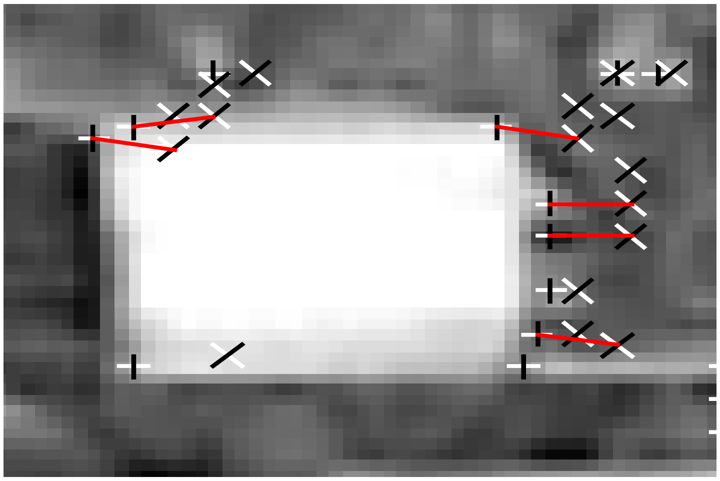


Figure 15- Group motion determination using displacement values of possible associations where some associations are discarded considering the estimated group motion.

5.1.2. Member Deletion And Addition

The derivations for the tracking method developed in Chapter 4 are theoretical and depends on some assumptions such as random clutter and detection with a certain probability. Similar to other tracking problems as radar tracking, random clutter is an assumption where clutter is sometimes persistent. Similarly, detection of a tracked member is not random and members sometimes disappear up to the end of track or a target related feature may suddenly appear at the middle of the tracking. Member deletion and addition is therefore important and necessary for track continuation. Thus, the core of the algorithm depends on the derivations of previous chapter with additional logic described herein.

Track deletion is possible with simple logic or with use of a score function having Bayesian information. Score function is not simple here because there are multiple members and a hard decision is not present to be used in score calculation. Then using a simple logic is favored. If there is no measurement in the individual validation gate of a member for a certain number of consecutive frames (scans), the track for that member is removed from the model. (Group model equations are modified by removing related components of the matrices in equations of previous chapter after member deletion.)

A member is generated from the observed features at the current frame, which are not involved in any individual validation gate of the members and can generate an association with the features of previous two frames without violating the group velocity estimate of the tracker. This is similar to the initialization logic explained in previous section. Figure 16 shows two consecutive frames and reveals how an obvious stable corner (up left) is inserted to tracker within tracking. Addition of a stable corner improves accuracy and life of tracking. Deletion of an unstable corner prevents the tracker from disturbances, similarly. This is the way how the tracking will survive in real world conditions.

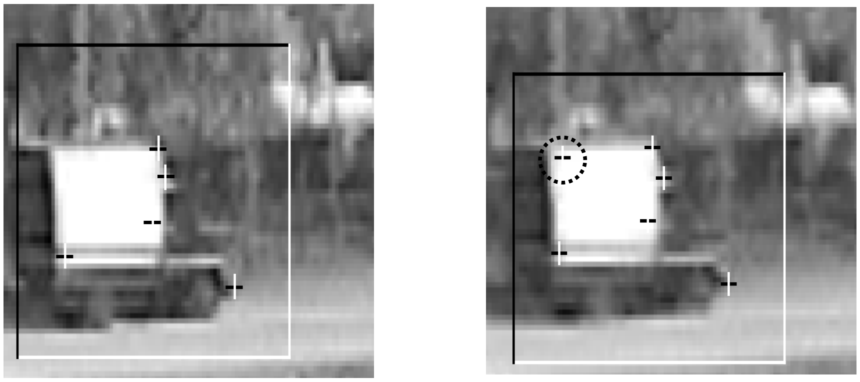


Figure 16- Member addition during track. A stable feature is added to the tracker (circled) during the track.

5.2. Simulation Results for Artificial Data Sequences

In this section, tracking results for artificial sequences are given. Artificially generated measurements are directly sent to the tracker without any need for image processing. The aim is to test the tracker with a totally known set of data. Since the true state vector of the system is known, true error of the tracker state estimate can be calculated easily. Also the artificial data is generated with known detection and false alarm probabilities, system and measurement noise variances. Therefore we have the chance to see the effects of tuning of the tracker parameters.

5.2.1. Artificial Data Simulation for 4 Member Tracking

Sequences for a group of 4 members is generated for different detection and false alarm probabilities to apply the tracking algorithm developed. In Figure 17 (a), the true trajectory of the 4 members of the group is seen for 50 scans. Figure 17 (b) shows the same trajectory but hiding the unobservable member states for *Pd*=0.6. The tracker error for different *Pd* values will be investigated first. Then the effect of false alarm rate will be presented by simulation. During the analysis, estimated and true values of a (any) member will be plotted and some snapshots from the tracking process will be given to visually analyze where the true and estimated target positions are, together with measurements and target window.

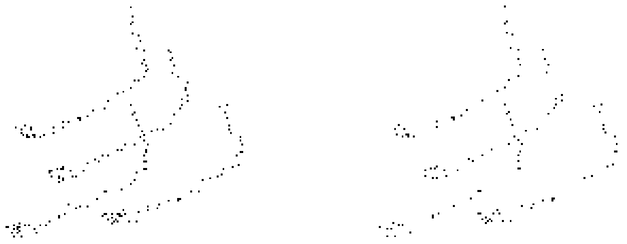


Figure 17- (a) Trajectory of the tracked 4 group members for 50 scans; (b) member detections for Pd=0.6.

Figure 18,19,20, and 21 are snapshots from a tracking process at frames 5, 20,35 and 50. P_d=1 and P_{fa}=0.001 for this sequence. The motion of the track gate, according to trajectory can be recognized easily from these figures. Figure 22,23,24 and 25 shows the track gates for these frames to visualize measurements, estimates and true values of members during track. Figure 26 is the comparison plot for reference point estimate and centroid of the estimated positions of the tracked group members for this case. Figure 27 shows the estimation performance with a plot of true and estimated position of member 1 (only x coordinate).

Figure 28 is the same plot as the previous one but for this case the input sequence has parameters *Pd*=0.6 and P_{fa}=0.001, and the tracker parameters are set to P_d=0.6 and P_{fa}=0.001. Compared with the previous plot the performance decrease can be seen when detection probability is decreased. Another case for performance decrease is the effect of an increase in false alarm probability. This is shown in Figure 29 where P_{fa} is 0.0025 this time.

We can also analyze mismatched tracker performances. A mismatch can be generated by increasing the tracker's rigidity variance value, Figure 30. This will deteriorate the tracking performance because this time, trust of the tracker to the target originated measurement will decrease and false alarms will become to be considered as true detections. The true rigidity deviation of member displacements is 0.5 pixel where we let the tracker suppose it as 3.

Gating can also result in over confidence and therefore track loss as explained in Chapter 3. Figure 31 is an example where individual gatings of targets are set to include target originated measurements with 95 percent probability. Compare this with the result for 99 percent probability in Figure 27.

All these trials reveals that we can first determine the worst case detection and false alarm probabilities first considering feature stability and clutter density of environment where the tracker is supposed to be used. Using artificially generated data with these parameters we can get accuracy and durability results for different gating size possibilities. This approach may help the initial tuning of the tracker for true image sequences.

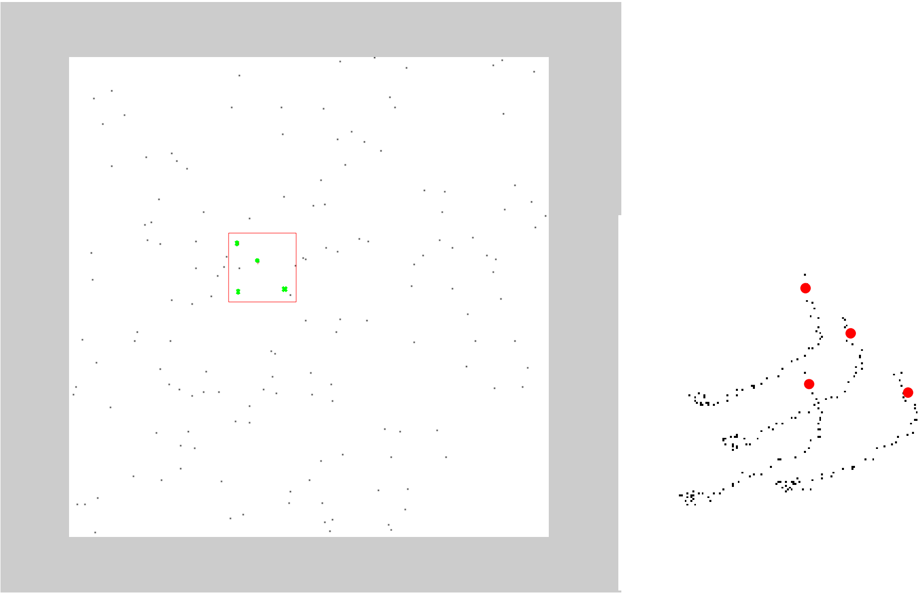


Figure 18- Position of track gate in Frame 5

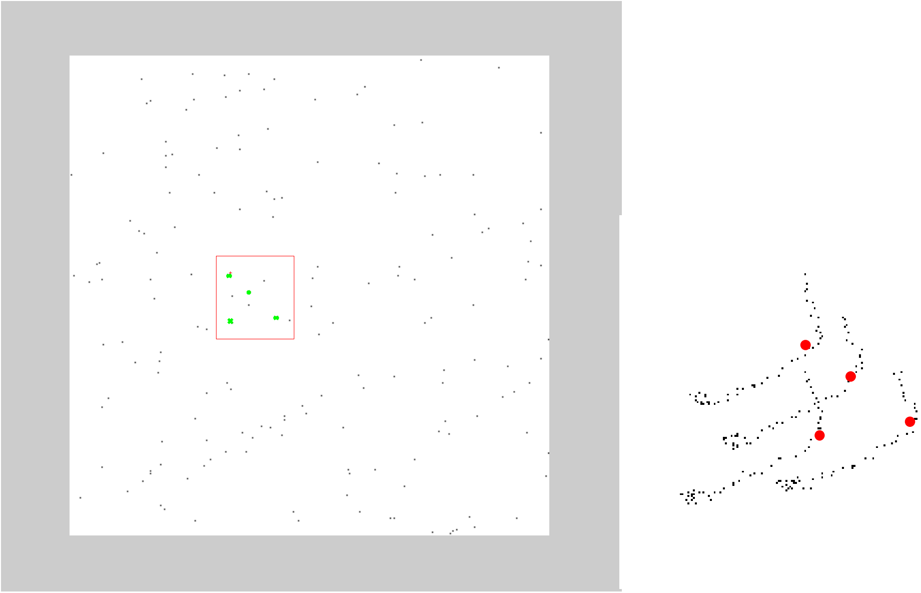


Figure 19- Position of track gate in Frame 20

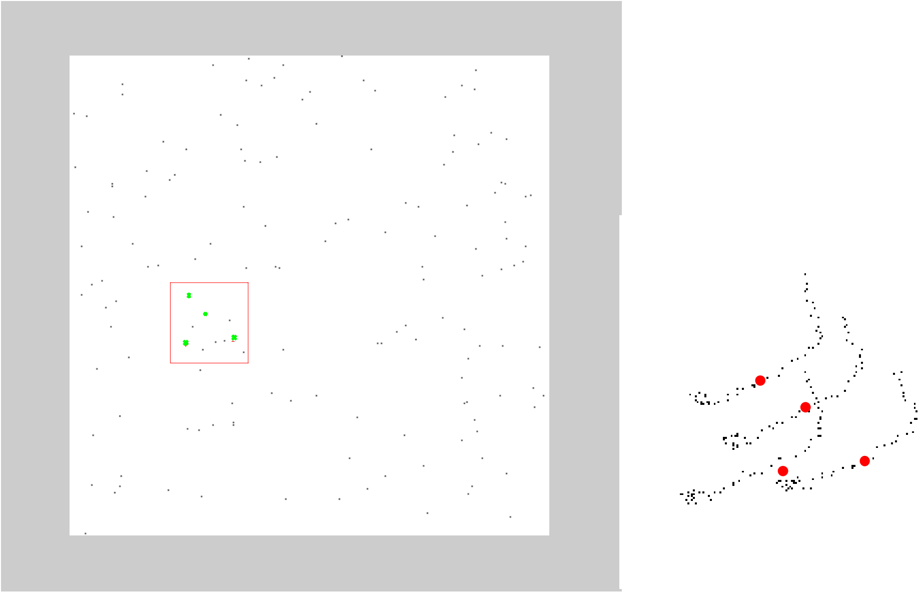


Figure 20- Position of track gate in Frame 35

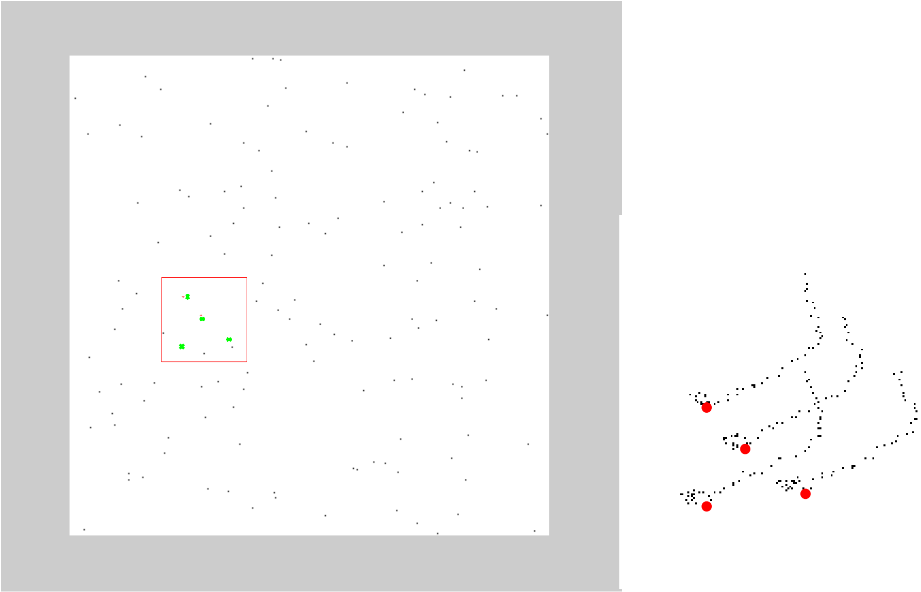


Figure 21- Position of track gate in Frame 50

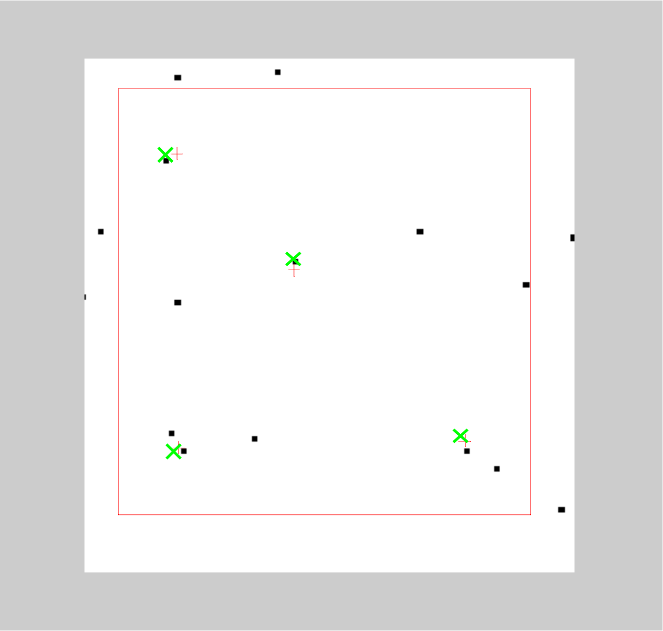


Figure 22-Track gate details for frame 5. (+→ estimated member position, x→true member position) P_d=1, P_{fa}=0 .001 for sequence and tracker parameters are: P_d=0.95, P_{fa}=0.001.

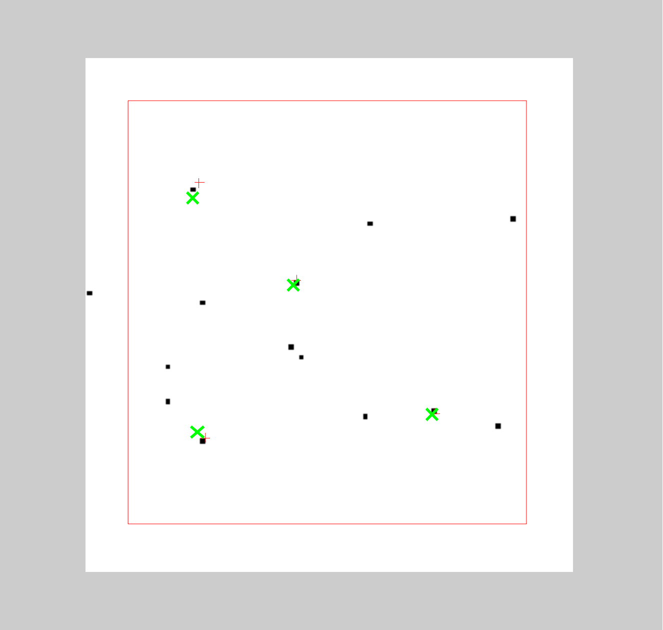


Figure 23- Track gate details for frame 20

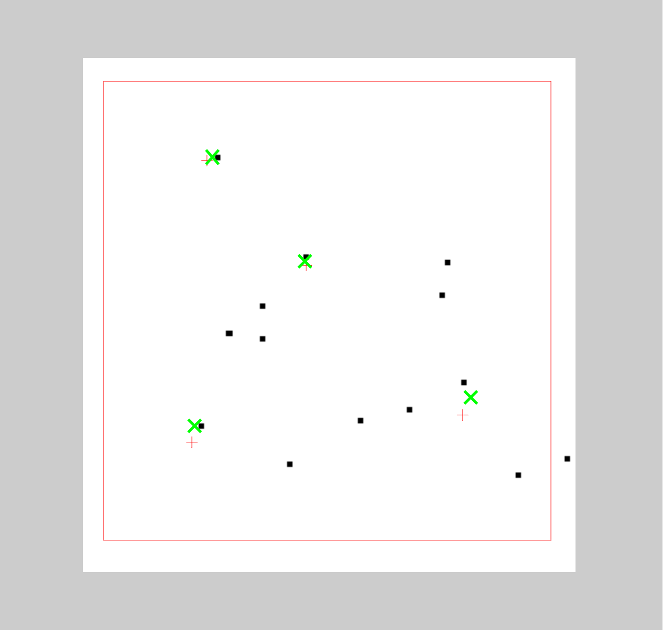


Figure 24- Track gate details for frame 35

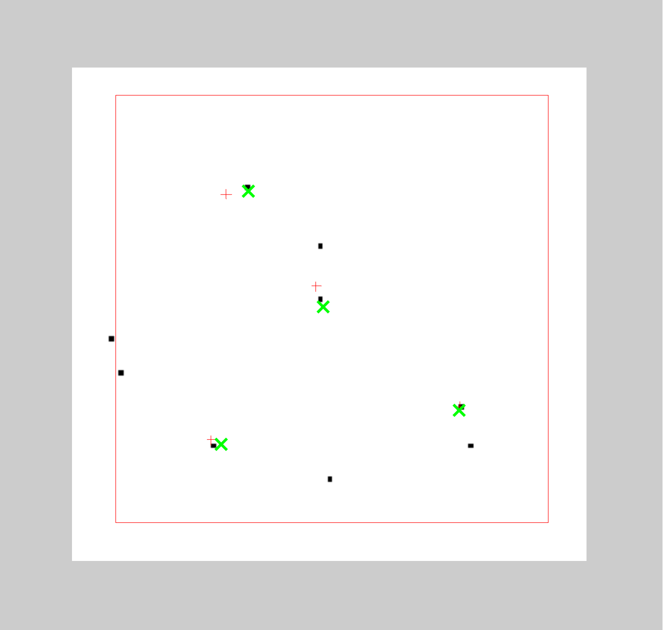


Figure 25- Track gate details for frame 50

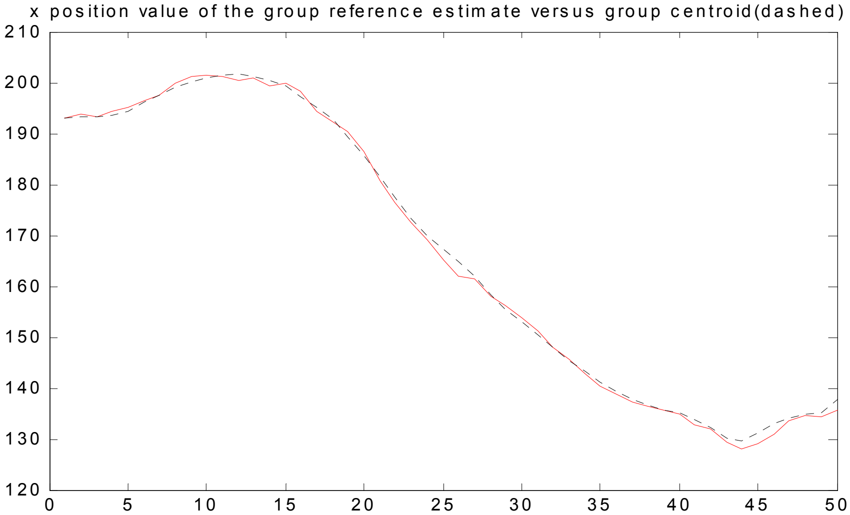


Figure 26- Comparison of the reference point estimate and centroid of the estimated positions from a sequence with P_d=1 and P_{fa}=0.001, tracked by a tracker having parameters P_d=0.95, P_{fa}=0.001.

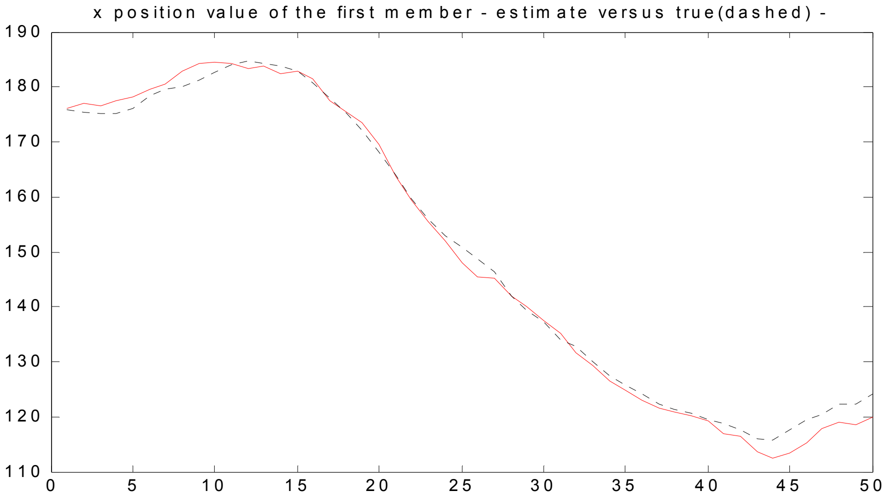


Figure 27- Comparison of true and estimated x coordinate position of member 1 from a sequence with P_d=1 and P_{fa}=0.001, tracked by a tracker having parameters P_d=0.95, P_{fa}=0.001

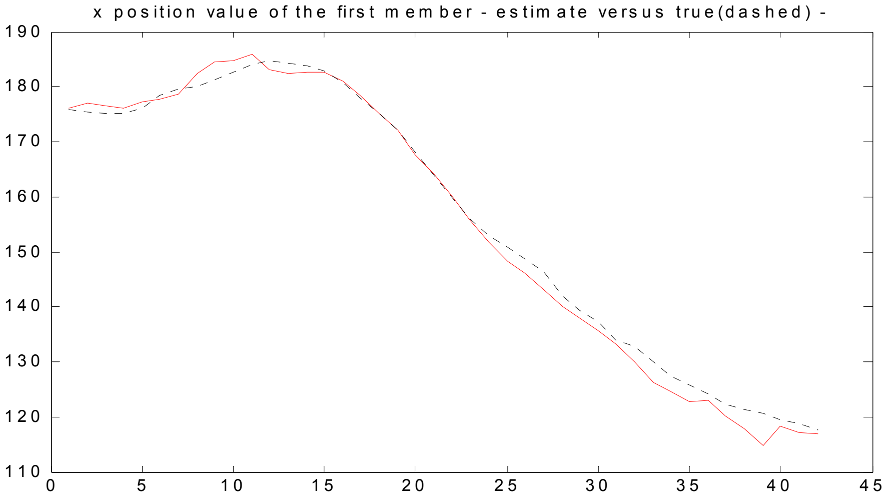


Figure 28- Comparison of true and estimated x coordinate position of member 1 from a sequence with P_d=0.6 and P_{fa}=0.001, tracked by a tracker having parameters P_d=0.60, P_{fa}=0.001.

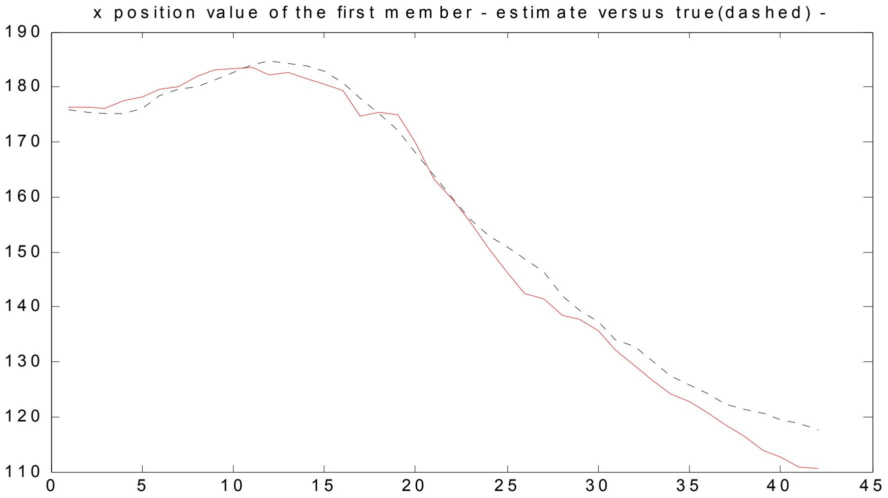


Figure 29- Comparison of true and estimated x coordinate position of member 1 from a sequence with P_d=0.6 and P_{fa}=0.0025, tracked by a tracker having parameters P_d=0.6, P_{fa}=0.0025.

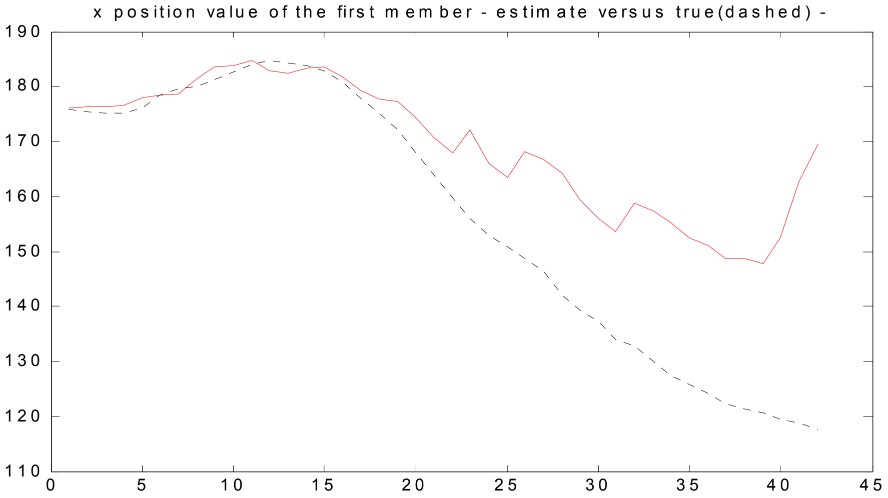


Figure 30-A mismatch and loss of track case where true rigidity noise deviation is 0.5 pixel for each member but the tracker assumes 3, for the case P_d=0.6 and P_{fa}=0.001

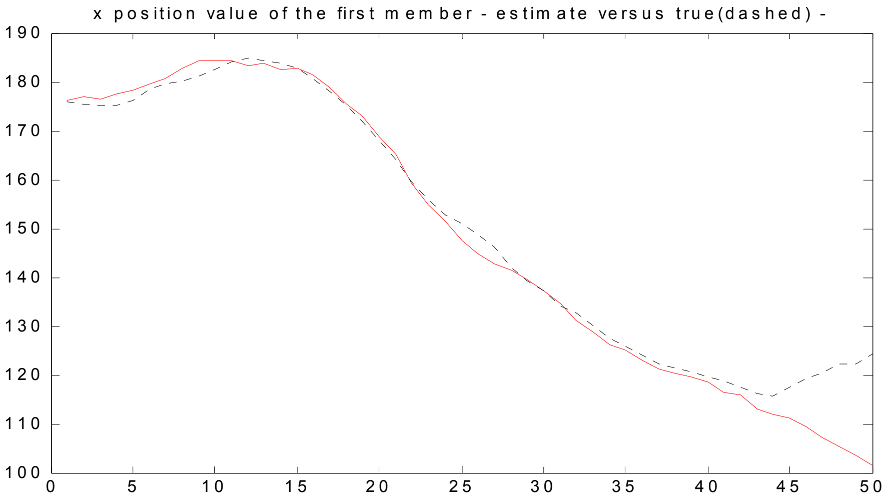


Figure 31- Loss of track due to narrow gating (overconfidence)(all parameters are same with Figure 27 except that the confidence percentage is decreased from 99 to 95)

5.3. Simulation Results for Image Data Sequence

5.3.1. Small Lorry Sequence

A real sequence of a small lorry will be used for the simulation of the developed tracker. Table 1 shows different tracker settings for some trials with this sequence. Since we have no observable state variables, member position estimates during various frames is given for the trials, to visually analyze how the estimated positions of member features are consistent during the tracking.

The eigenvalue maps and the detected features are also shown to see how tracker is fed by information. The quality (stability) of target features and clutter density is understood using these figures. Then, effect of detection and false alarm probability settings is investigated with trials 2 and 3. It is seen that for this sequence of data, changing detection and false alarm probability parameters do not effect the tracking too much. This is possible if there is not too much competition between hypothesis. This implies that successful information on member positions is available from feature detection process and spurious features are low in number.

In trial 4, *fspan* parameter is decreased to allow more neighbor features to be detected and it is seen that this affects the stability of some tracked features when compared with previous trials. The reason is the increased ambiguity because we allowed detections which are close to each other, some of which may be clutter (Figure 46).

Member addition is also observed in all trials (see Figure 37). If we compare frame 14 and 26 of trial 1 (Figure 37 and Figure 38) we see that smoothing effect of group tracking help for the survival of the upper feature, which is disturbed by background clutter. Below are related figures for comparison and analysis.

	Pd	Pfa	fspan
TRIAL 1	0.5	0.005	4
TRIAL 2	0.9	0.005	4
TRIAL 3	0.5	0.03	4
TRIAL 4	0.5	0.03	3

Table 1- Parameter values for different trials.

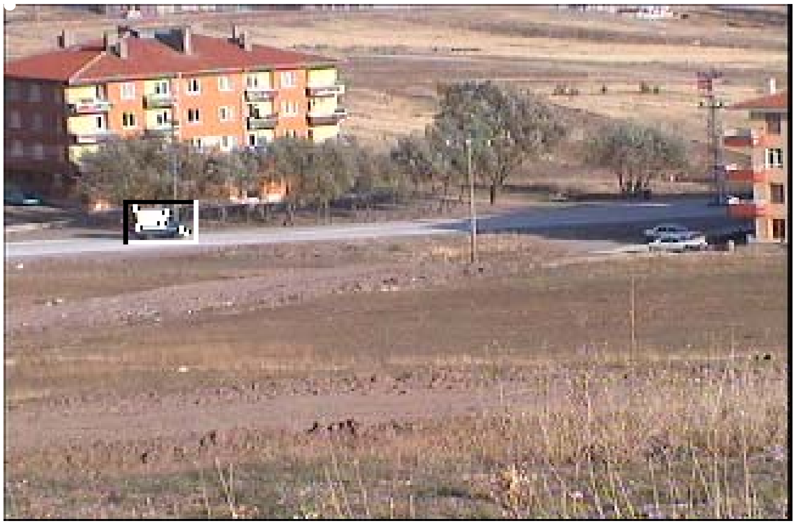


Figure 32- First frame for trial 1. Initial position of target vehicle.

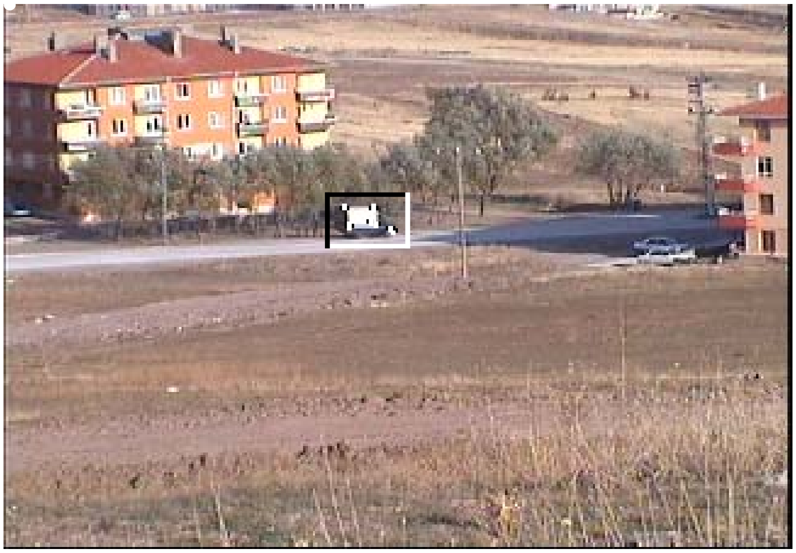


Figure 33- Last **(**38-th) frame for trial 1. (Compare motion of vehicle with its initial position at frame 1 to get a feeling on the speed and background characteristics.)

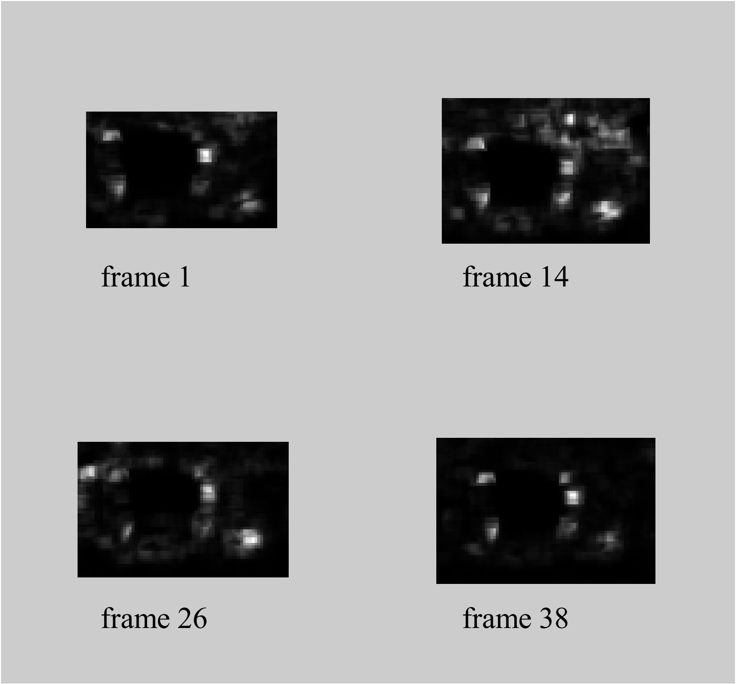


Figure 34- Eigenmaps for frames 1,14,26 and 38 for trial 1. Given to visualize how regions with high cornerness (brighter regions) are distributed in the track window during the track.

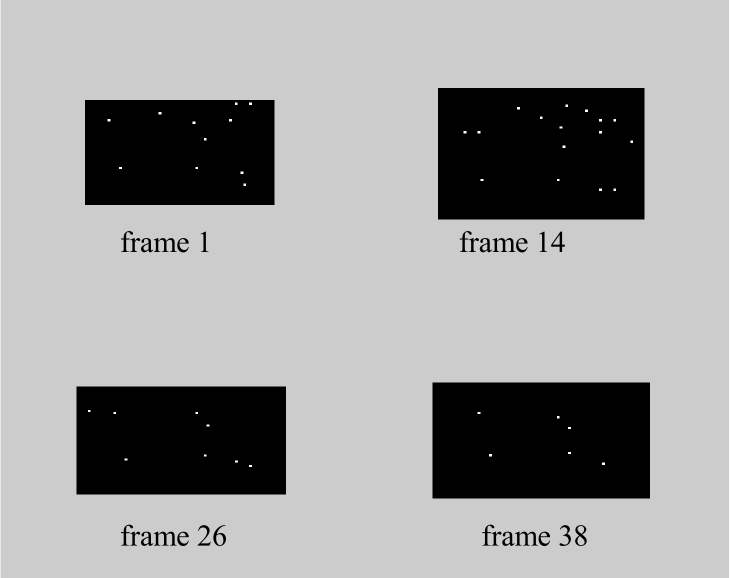
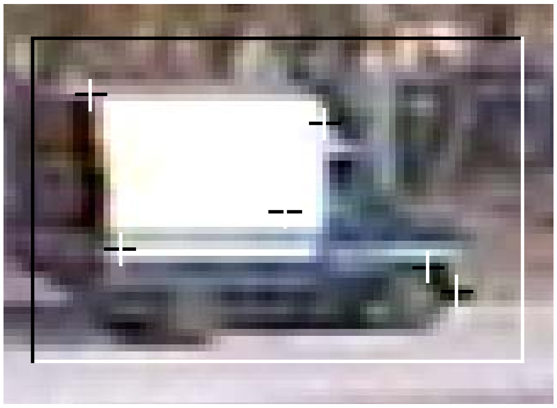


Figure 35- Corresponding detections for trial 1 using eigenmaps in previous figure. Illustrates how stable target corners and how spurious detections (clutter) are present during different scans (frames) of the track.



-

Figure 36- Member feature position estimates ("+") for frame 1 of trial 1.

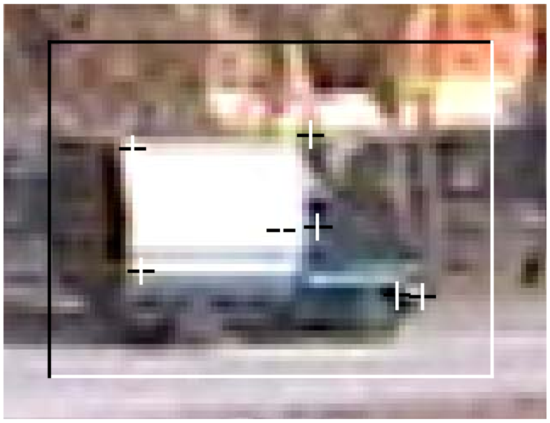


Figure 37- Member feature position estimates ("+") for frame 14 of trial 1. When compared with the feature estimate at frame 1 it is seen that the feature at the upper right of vehicle is disturbed by the corner at the background. It is also seen that a feature addition is occurred at the middle right of the vehicle.

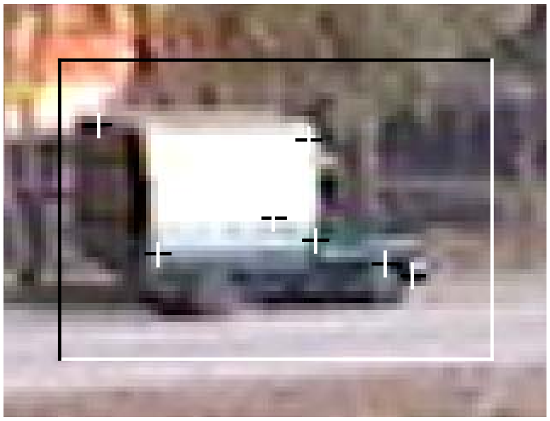


Figure 38- Member feature position estimates ("+") for frame 26 of trial 1. Considering frame 14, the feature disturbed by background clutter (upper right) is survived and it represents target related feature now. This survival is because of the smoothing effect of the group motion. Dependency of each target feature to the other prevents the distortion of features. However this time the feature at the upper left of the vehicle is effected by the high contrast background, if you compare its original position with respect to vehicle at frame 14. But in frame 38 it is observed that this feature recovered its original location.

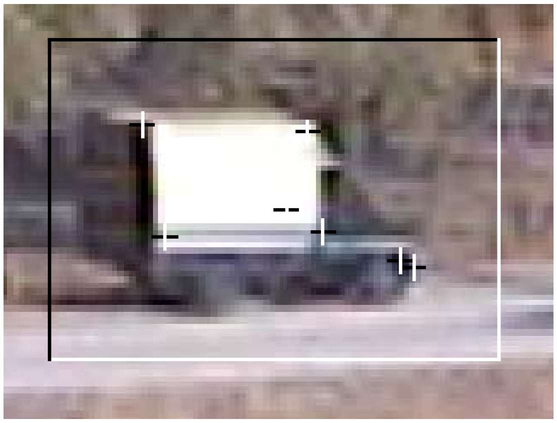


Figure 39- Member feature position estimates ("+") for frame 38 of trial 1. Illustrates final positions of member feature estimates after 38 frames.

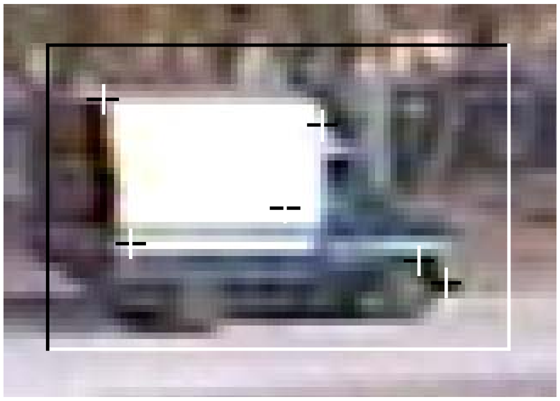


Figure 40- Member feature position estimates ("+") for frame 1 of trial 2. This figure is given for comparison with trial 1.

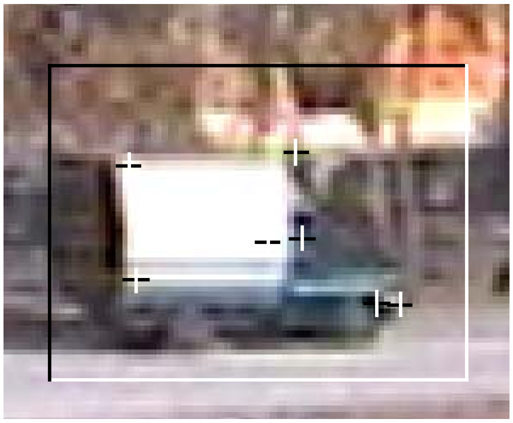


Figure 41- Member feature position estimates ("+") for frame 14 of trial 2. This figure is given for comparison with trial 1.

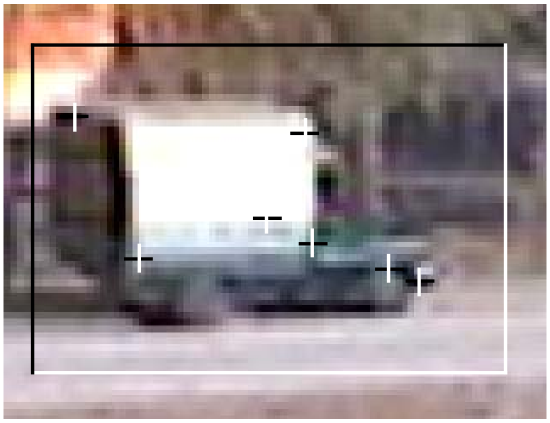


Figure 42- Member feature position estimates ("+") for frame 26 of trial 2. This figure is given for comparison with trial 1.

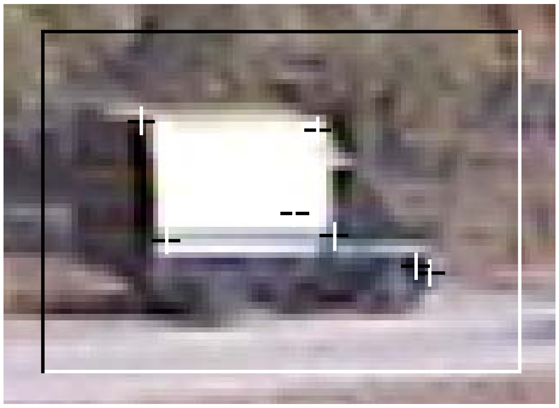


Figure 43- Member feature position estimates ("+") for frame 38 of trial 2. This figure is given for comparison with trial 1.

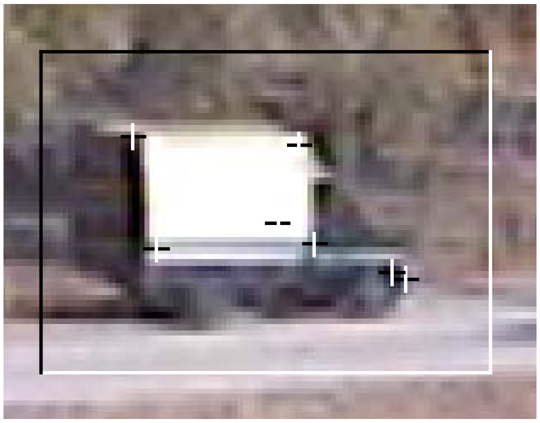


Figure 44- Member feature position estimates ("+") for frame 38 of trial 3. This figure is given for comparison with trial 1 and 2.

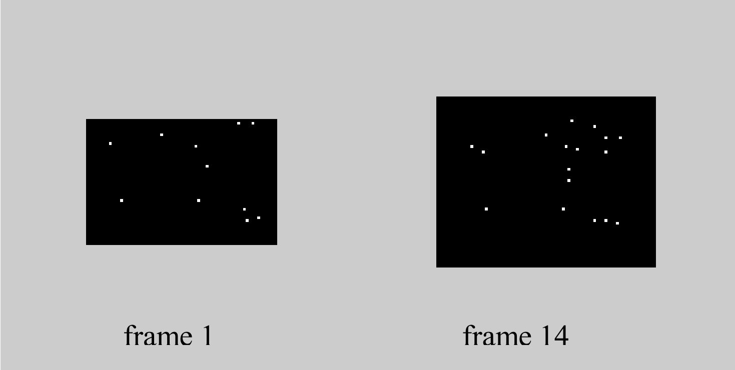


Figure 45- Feature detections for trial 4. This figure is given to illustrate how detections are effected when the *fspan* value, determining the minimum separation allowed between detected corners, is decreased.

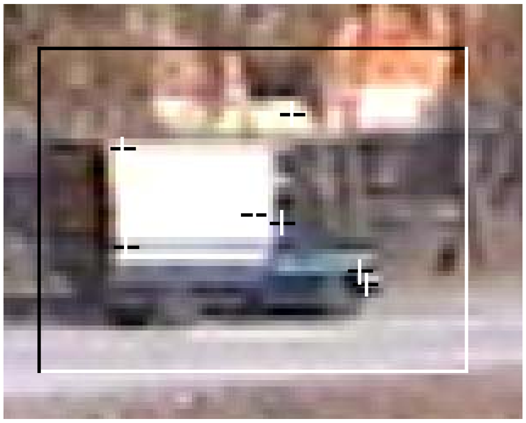


Figure 46- Member updates ("+") for frame 14 of trial 4. This figure illustrates that with decreasing *fspan* value, we increased ambiguity because we allowed detections, which are close to each other, some of which may be clutter. Therefore compared with frame 14 of previous trials, the feature at upper right is heavily affected by background clutter.

CHAPTER 6

CONCLUSION

6.1. Conclusion

In this study, a group tracker is designed for tracking single visual targets. The applicability of the approach is not unique to visual targets with features and method can be used for applications where target dynamics can be modeled as a group motion. In other words, if the dynamics of multiple targets are strongly coupled with each other, group model used with this technique is more proper than using individual models and filters for targets. Figure 47 shows how multiple target tracking algorithms can be classified considering where our algorithm stands.

The important feature of our method is the use of PDA like (all neighborhoods) state estimation to increase accuracy of estimation considering the NN alternatives used by traditional group tracking methods. Different from the other group trackers, centroid of group members is not used in this method because accurate tracking of all members is performed. This leads to an important consequence on applicability of this technique. That is, differing from other group tracking approaches, the members of the group should not necessarily be closely spaced. Another important point is that, it is developed directly for visual targets unlike radar and sonar tracking based algorithms adapted to visual tracking applications. Also using a group tracking approach for single target tracking is a distinctive property of our method.

Since the group member dynamics is estimated in this method, particularly disappeared members can be tolerated with the smoothing effect of the group tracker. Long term loss of members causes deletion of that member track from tracker to keep the track accuracy and similarly, new features are accepted during the track for continuity of the tracking process.

Computational and memory requirements are closely related to number of members being tracked and therefore some parameters are defined and used in feature detection phase of the tracker. They also affect the stability of selected features for tracking.

It should also be stated that Gaussian distribution assumption both for clutter and system noise is an approximation. Furthermore, in visual tracking, parameters such as detection and false alarm should be tuned considering the possible application environment of the tracker and type of the sensor (air or ground targets, IR or day TV, *et cetera*).

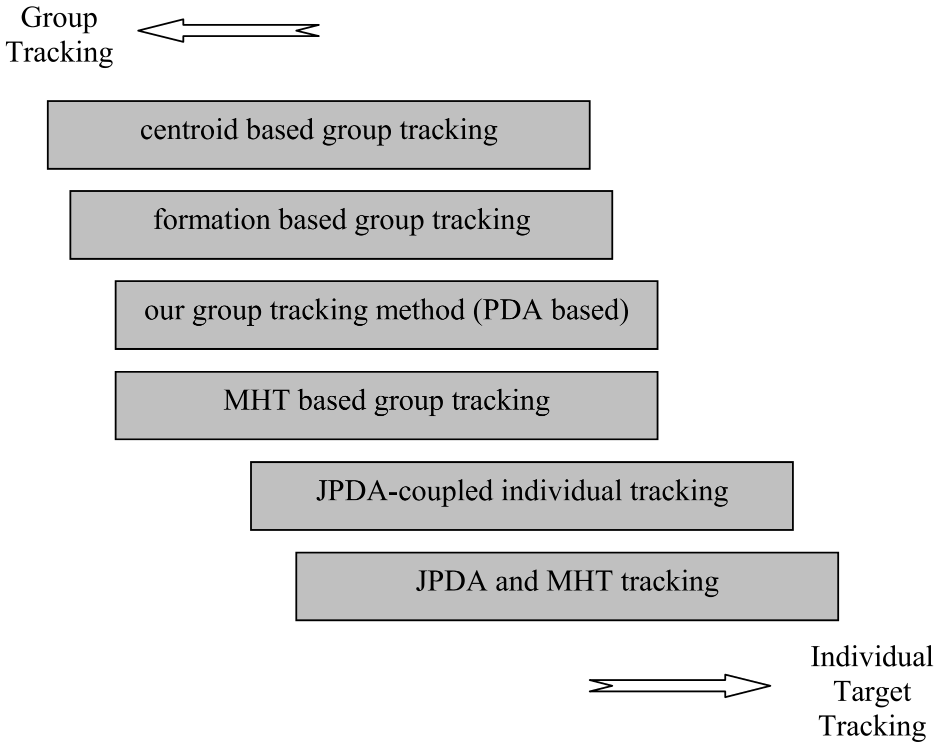

Group Tracking
centroid based group tracking
formation based group tracking
our group tracking method (PDA based)
MHT based group tracking
JPDA-coupled individual tracking
JPDA and MHT tracking
Individual Target Tracking

Figure 47- Classification of multiple target tracking algorithms.

6.2. Future Work

Some improvements are possible for our approach. Criteria for addition and removal of features can be improved. Secondly, the output of this tracker can feed a higher level reasoning block. Member addition and removal can also be improved by the aid of this reasoning block. Improvements on initialization phase are also possible to increase accuracy.

New pixel processing strategies to increase feature stability is also possible. Feature matching is used in only initialization phase but it can also be included in the tracking phase as an additional attribute.

Background motion due to imager platform is ignored in this work. For the case of background motion, additional image processing methods can be developed or platform motion measurements can be used to modify the state equations.

As stated before independent false alarm (clutter) and feature detection probabilities are assumptions. However, persistent clutter is inevitable for visual targets in many applications. Considering these facts, dynamic parameter tuning may be investigated.

Order of the group model can be increased depending on the target dynamics or switching multiple model approaches may be used with this method.

REFERENCES

[1] Robert E. Nasburg (Michael Dudzig, Editor), *The Infrared and Electro-Optical System Design Handbook, Volume 4: Electro-Optical System Design, Analysis and Testing, Chapter 5,* SPIE, 1993

[2] Gillian K. Groves, Spencer W.White, Michael D. Vahey, John A.Harding, “Reconfigurable Video Tracker”, *Acquisation Tracking and Pointing XIII, pp. 216-225,* Proceedings of SPIE, 1999

[3] Teresa L.P. Olson, Carl W. Sanford, “A Real-Time Multistage IR Image-Based Tracker”, *Acquisation Tracking and Pointing XIII, pp. 226-233,* Proceedings of SPIE, 1999

[4] Jerry W. Bukley, Robert M. Cramblitt, “Comparison of Image Processing Algorithms for Tracking Illuminated Targets”, *Acquisation Tracking and Pointing XIII*, pp. 234-243, Proceedings of SPIE, 1999

[5] Michael K. Masten, *Selected Papers on Precision Stabilization & Tracking Systems for Acquisation, Pointing and Control Applications*, SPIE, 1996

[6] H. M.Shertukde, Y. Bar-Shalom, *Multitarget-Multisensor Tracking, Volume II, Chapter 5:Precision Tracking of Small Extended Targets with Imaging Sensors,* Artech House, 1992

[7] R.W.Sitter, “An Optimal Data Association Problem in Surveillance Theory”, *IEEE Trans. Military Electronics,* Vol. MIL-8, pp. 125-139, April 1964

[8] Y. Bar-Shalom, T. E. Fortmann, *Tracking and Data Association*, Academic Press, Inc., 1988

[9] Y. Bar-Shalom, Xiao-Rong Li, *Multitarget-Multisensor Tracking,1995*

[10]Neil Gordon, David Salmond, David Fischer, “Bayesian Tracking after group pattern distortion,” *Signal and Data Processing of Small Targets*, Proc. SPIE, Vol. 2759, pp.279-292, Apr.1996

[11]Samuel S. Blackman, *Multiple-Target Tarcking with Radar Applications*, Artech House, 1986

[12]Samuel Blackman, Robert Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999

[13]Jonathen Michael Roberts, “Attentive Visual Tracking and Trajectory Estimation for Dynamic Scene Segmentation”, *Submitted for degree of Doctor of Philosophy,* University of Southampton, 1994

[14]John Christopher Clarke, “Applications of Sequence Geometry to Visual Motion”, *Submitted for degree of Doctor of Philosophy,* University od Oxford, 1997

[15]Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes,*Third edition, McGraw-Hill, Inc., 1991

[16]Robert L. Popp, Krisna R. Pattipati, Yaakov Bar-Shalom, Muralli Yeddanapudi, “Parallelization of a Multiple Model Multitarget Tracking Algorithm with Superior Speedups”, *IEEE Transactions on Aerospace and Electronic Systems* Vol.33, No.1 Jan.1997

[17]Robert L. Popp, Krisna R. Pattipati, Yaakov Bar-Shalom, “Dynamically Adaptable m-Best 2-D Assignment Algorithm and Multilevel Parallelization”, *IEEE Transactions on Aerospace and Electronic Systems* Vol.35, No.4, Oct.1999

[18]Robert L. Popp, Krisna R. Pattipati, Yaakov Bar-Shalom, “m-Best S-D Assignment Algorithm with Application to Multitarget Tracking”, *IEEE Transactions on Aerospace and Electronic Systems* Vol.37, No.1, Jan.2001

[19]Ingemar J.Cox, Matt L.Miller, “A Comparison Of Two Assignment Algorithms For Determining Ranked Assignments With Application to Multi-Target Tracking and Motion Correspondence” *NEC Research Institue Technical Report 95-07,* 1995

[20]Katta G. Murty, “An algorithm for ranking all the assignments in order of increasing cost”, *Operations Research,* vol.16 pp. 682-687,1968

[21]A. Murat Tekalp, *Digital Video Processing, Prentice Hall PTR,*1995

[22]Y. Bar-Shalom (editor) *Multitarget-Multisensor Tracking: Advanced Applications.* Artech House, Norwood, 1992

[23]J.K. Aggarwal, N. Nandhakumar “On the computation of motion from sequences of images – A Review, *Proceedings of the IEEE*,76(8):917-935, 1988

[24]Carlo Tomasi, Takeo Kanade, “Detection and Tracking of Point Features” *CMU-CS-91-132,* School of computer science, Carnegie Mellon University, April 1992

[25]Jianbo Shi, Carlo Tomasi, “Good Features To Track”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, June 1994

[26]A. Fusiello, E.Trucco, T.Tommasini, V.Roberto, “Improving Feature Tracking with Robust Statistics” *Pattern Analysis and Applications* 2:312-320, 1999