

PATH EXTRACTION OF LOW SNR DIM TARGETS
FROM GRAYSCALE 2-D IMAGE SEQUENCES

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

BY

SAİT ERGÜVEN

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

SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences.

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. İsmet ERKMEN
Head of the 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. Kerim DEMİRBAŞ
Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU (METU, EE)

Prof. Dr. Kerim DEMİRBAŞ (METU, EE)

Assoc. Prof. Dr. Çağatay CANDAN (METU, EE)

Assoc. Prof. Dr. Fahrettin ARSLAN (Ankara Univ., STAT)

M. Sc. İrfan OKŞAR (ASELSAN, MST)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Sait Ergüven

Signature :

ABSTRACT

PATH EXTRACTION OF LOW SNR DIM TARGETS FROM GRAYSCALE 2-D IMAGE SEQUENCES

Ergüven, Sait

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kerim Demirbaş

September 2006, 101 pages

In this thesis, an algorithm for visual detecting and tracking of very low SNR targets, i.e. dim targets, is developed. Image processing of single frame in time cannot be used for this aim due to the closeness of intensity spectrums of the background and target. Therefore; change detection of super pixels, a group of pixels that has sufficient statistics for likelihood ratio testing, is proposed. Super pixels that are determined as transition points are signed on a binary difference matrix and grouped by 4-Connected Labeling method. Each label is processed to find its vector movement in the next frame by Label Destruction and Centroids Mapping techniques. Candidate centroids are put into Distribution Density Function Maximization and Maximum Histogram Size Filtering methods to find the target related motion vectors. Noise related mappings are eliminated by Range and Maneuver Filtering. Geometrical centroids obtained on each frame are used as the observed target path which is put into Optimum Decoding Based Smoothing Algorithm to smooth and estimate the real target path.

Optimum Decoding Based Smoothing Algorithm is based on quantization of possible states, i.e. observed target path centroids, and Viterbi Algorithm. According to the system and observation models, metric values of all possible target paths are computed using observation and transition probabilities. The path which results in maximum metric value at the last frame is decided as the estimated target path.

Keywords: Dim Target, Super pixel, Likelihood Ratio Test, 4-Connected Labeling, Distribution Density Function Maximization, Maximum Histogram Size Filtering, Range and Maneuver Filtering, Optimum Decoding Based Smoothing Algorithm, Viterbi Algorithm, Observation and Transition Metrics

ÖZ

İKİ BOYUTLU GRİ TONLAMALI GÖRÜNTÜ DİZİSİNDEN DÜŞÜK SNR'LI SOLUK HEDEFLERE AİT YOL TAYİNİ

Ergüven, Sait

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Kerim Demirbaş

Eylül 2006, 101 sayfa

Bu tezde görsel olarak çok düşük sinyal gürültü seviyeli soluk hedefleri saptama ve izleme amaçlı bir algoritma geliştirilmiştir. Bu amaç için çevre ve hedef yeğlilik spektrumlarındaki yakınsaklık nedeniyle tek görüntü üzerinde görüntü işleme yöntemleri kullanılamaz. Bu nedenle, istatistiksel açıdan yeterli bilgiye sahip en küçük pixel grubu olan süper pikseller üzerindeki olabilirlik oran testine bağlı değişim algılaması yöntemi öne sürülmüştür. Geçiş noktaları olarak karar verilen süper pikseller ikili fark matrisi üzerine eşlemlenerek 4'lü bağlantı etiketleme yöntemi ile gruplanır. Her etiket hedefle ilişkili vektörel hareketleri bulmak için Etiket Parçalama ve Merkezi Nokta Eşleme teknikleri ile işlenir. Gürültü ilişkili hareketler Uzaklık ve Manevra Filtrelemesi ile elenir. Aday merkezi noktalar Görüntü Dağılım Yoğunluğu Fonksiyonu Enbüyültme ve En Büyük Histogram Büyüklük Filtrelemesi yöntemleri ile işlenir. Her görüntü üzerinde hedefe ait olası geometrik merkezi noktalar birleştirilerek elde edilen gözlem yolu bulunduğundan sonra eniyi kodçözümleyici yöntemine dayalı yumuşatma algoritması ile gerçek hedef yolu kestirmesi yapılır.

Bu algoritma hedef yoluna ait durum nicemlemesi ve Viterbi Algoritmasına dayalıdır. Bütün olası hedef yollarına ait metrik deęerleri; sistemin durum ve gözleme modellerine dayalı olan gözlem ve geçiş olasılıklarından yola çıkılarak hesaplanır. En büyük metrik deęerine sahip yol, hedefin izledięi yol olarak kestirilir.

Anahtar Kelimeler: Soluk Hedef, Süperpiksel, Olabilirlik Oran Testi, 4'lü bağlantı Etiketlemesi, Dağılım Yoęunluęu Fonksiyonu Enbüyültme, En Büyük Histogram Büyüklük Filtrelemesi, Uzaklık ve Manevra Filtrelemesi, Eniyi Kodçözümleyici Yöntemine Dayalı Yumuşatma Algoritması, Viterbi Algoritması, Gözlem ve geçiş metrikleri

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my supervisor Prof. Dr. Kerim DEMİRBAŞ for his guidance, advice, criticism, encouragements and insight throughout the research.

I would like to thank to ASELSAN INC. and my working friends for facilities provided for the completion of this thesis.

I am very grateful to my family for their assistance and encouragements.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	vi
ACKNOWLEDGMENT	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	xi
LIST OF FIGURES.....	xii
1. INTRODUCTION.....	1
2. DYNAMIC VISION	3
2.1 Change Detection	7
2.2 Connection Labeling Of Changing Super pixels.....	10
2.3 Label Destruction and Centroids Mapping	12
2.4 Maximization of Distribution Density Functions.....	16
2.5 Maximum Histogram Size Filtering.....	21
2.6 Range and Maneuver Filtering.....	23
2.7 Position and Direction Calibration.....	25
2.8 Assignment of the Next Iteration Label	27
2.9 Flowchart of the Dim Target Tracking Algorithm.....	28
3. OPTIMUM DECODING BASED SMOOTHING ALGORITHM.....	30
3.1 Quantization of States and Transition Probabilities.....	31
3.2 Approximation of an Absolutely Continuous Random Vector by a Discrete Random Vector	33
3.3 A Finite State Model for the Target Model.....	35
3.4 Approximate Observation Models	37
3.5 Minimum Error Probability Criterion	38
3.6 Optimum Decision Rule for the Target Paths	38
3.7 Optimum Decoding Based Smoothing Algorithm.....	40
3.8 An Example of the ODSA Algorithm.....	43
4. SIMULATION RESULTS FOR DIM TARGET TRACKING ALGORITHM....	46
4.1 Effect of Observation Parameters	49
4.1.1 Effect of Environment Noise Statistics	49
4.1.2 Effect of Image Size.....	51
4.1.4 Effect of Illumination	52
4.2 Effect of Target Parameters.....	53
4.2.1 Effect of Target Size and Shape.....	53
4.2.2 Effect of Target Motion Model	55
4.2.3 Effect of Target Motion Parameters.....	58
4.3 Effect of Risk Parameters.....	61
4.3.1 Effect of Superpixel Size.....	61
4.3.2 Effect of LRT Change Detection Threshold	64
4.3.3 Effect of Distribution Density Function.....	66
4.3.4 Effect of Minimum and Maximum Expected Target Areas.....	70

4.4 Effect of ODSA Parameters	74
4.4.1 Effect of the Gate Size	75
4.4.2 Effect of the Quantization Number of the Initial Target Path Vector	76
4.4.3 Effect of the Quantization Number of the Disturbance Noise	77
4.4.4 Effect of the Initial Target Path Vector Variance	78
4.4.5 Effect of the Disturbance Noise Variance.....	79
4.4.6 Effect of the Observation Noise Variance.....	80
4.4.7 Effect of Limiting the Maximum Possible Target Points Number	81
4.5 Comparison of Kalman Filter with ODSA for Different Target Dynamics.....	82
4.5.1 Linear Motion.....	82
4.5.1 Nonlinear Motion	84
4.5.1 Maximum Tolerable Noise.....	94
5. CONCLUSIONS.....	96
6. REFERENCES.....	100

LIST OF TABLES

Table 1 y and p values of discrete random variable with 8 possible values.....	35
Table 2 Simulation Algorithm Input Parameters	48
Table 3 Minimum Tolerable SNR for detection with different sizes and shapes ...	54
Table 4 Target Motion Models and Estimated RMS Errors	55
Table 5 Different Types of DDF with their Estimated Path RMS Errors	67

LIST OF FIGURES

Figure 1 Background Pixel Intensity Spectrum, N (100, 15).....	4
Figure 2 Target Pixel Intensity Spectrum, N (115, 15).....	5
Figure 3 Example of a dim air target in a foggy environment	7
Figure 4 4-Connected and 8-Connected Labeling Directions	11
Figure 5 4-Connected and 8-Connected Label Matrix with SNR: 0.81	11
Figure 6 Illustration of target and noise related neighbor super pixels	13
Figure 7 Centroids Distribution without Label Destruction.....	13
Figure 8 Centroids Distribution with Label Destruction.....	14
Figure 9 Illustration of Rectangular Label Destruction in the Bounding Box	16
Figure 10 Example of Exponential Distribution Density Function: e^{-r^2} with " $r^2 = x^2 + y^2$ "	17
Figure 11 Example of Linear Distribution Density Function: $\text{Max}(10, 10-r)$ with " $r^2 = x^2 + y^2$ "	17
Figure 12 A real target that appears on DDF Maximization.....	20
Figure 13 Differentiation of different targets for different size thresholds.....	22
Figure 14 Example of target range and maneuver filtering	24
Figure 15 The calibration of the real target position and direction	26
Figure 16 General Flowchart of the Dim Target Tracking Algorithm.....	29
Figure 17 Quantization and transition probabilities.....	32
Figure 18 PDF of continuous random variable x and approximated discrete random variable	35
Figure 19 Trellis diagram for the target motion	37
Figure 20 Trellis diagram for the target motion from time zero to time 2	43
Figure 21 Trellis diagram from time zero to time 2 at the end of first step	44
Figure 22 Trellis diagram from time zero to time 2 at the end of second step	45
Figure 23 Plot of Minimum Tolerable Difference of Target and Environment Mean Intensity vs. Noise Variance.....	49
Figure 24 Plot of $\Delta\mu$ Intensity vs. RMS Target Path Error	49
Figure 25 Plot of Image Size vs. Time Consumption in 50 frames for different SNR values.....	51
Figure 26 Plot of Probability of Detection for different illumination functions	52
Figure 27 Plot of Target Area for different shapes vs. Minimum Tolerable Target SNR	53
Figure 28 Plot of Estimated Path RMS Error for different Target Motion Models	57
Figure 29 Plot of Target Label (white) in the Difference Matrix Target Velocity: 20 Pixels / Frame SNR: 2.24 and Size: [8 8]	58
Figure 30 Plot of Target Label (white) in the Difference Matrix Target Velocity: 15 Pixels / Frame SNR: 2.24 and Size: [8 8]	58

Figure 31 Plot of Target Label (white) in the Difference Matrix Target Velocity: 10 Pixels / Frame SNR: 2.24 and Size: [8 8]	59
Figure 32 Plot of Target Label (white) in the Difference Matrix Target Velocity: 7 Pixels / Frame SNR: 2.24 and Size: [8 8]	59
Figure 33 Plot of Target Label (white) in the Difference Matrix Target Velocity: 20 Pixels / Frame SNR: 1.55 and Size: [8 8]	60
Figure 34 Plot of Target Label (white) in the Difference Matrix Target Velocity: 15 Pixels / Frame SNR: 1.55 and Size: [8 8]	60
Figure 35 Plot of Superpixel Size vs. Centroid Errors for different targets.....	61
Figure 36 Plot of Superpixel Size vs. Number pf Labels for different targets.....	62
Figure 37 Plot of Superpixel Size vs. Time of Analysis for different targets.....	62
Figure 38 Effect of Superpixel Size to Number of Centroids	63
Figure 39 Plot of Estimated Target Path with Motion Model: LCV, Start:[0,0] Finish:[500, 500]	64
Figure 40 Plot of Estimated Target Path with Motion Model: LCV, Start: [0, 0] Finish: [500, 500]	64
Figure 41 Plot of Estimated Target Path with Target Motion Model: LCV, SP: [0, 0] LP: [500 500]	65
Figure 42 Plot of Estimated Target Path with Target Motion Model: LCV SP: [0, 0] LP: [500 500]	65
Figure 43 Plot of Estimated Target Path RMS vs. Distribution Density Function Variance	66
Figure 44 Plot of Different Distribution Density Functions along X axis	67
Figure 45 Plot of scattered mapping centroids in the focus window	68
Figure 46 Plot of Step DDF outputs for Figure 45 with different bandwidths	68
Figure 47 Plot of Exponential DDF outputs for Figure 45 with different bandwidths	69
Figure 48 Plot of Exponential DDF outputs for Figure 45 with different bandwidths	69
Figure 49 Plot of Minimum Expected Target Area vs. Probability of Detection ...	70
Figure 50 Plot of Minimum Target Area vs. Average Number of Targets.....	71
Figure 51 Plot of Maximum Expected Target Area vs. Number of Target Candidates	72
Figure 52 Plot of Maximum Expected Target Size vs. Minimum Identification Distance.....	72
Figure 53 Plot of Frame Number vs. Average Estimation Error for Different Gate Sizes	75
Figure 54 Plot of Frame Number vs. Estimation Error with respect to Initial Quantization Number	76
Figure 55 Plot of Frame Number vs. Estimation Error with respect to Disturbance Noise Quantization Number.....	77
Figure 56 Plot of Frame Number vs. Estimation Error with respect to Initial Target Path Vector Variance	78
Figure 57 Plot of Frame Number vs. Estimation Error with respect to Disturbance Noise Variance	79

Figure 58 Plot of Frame Number vs. Estimation Error with respect to Observation Noise Variance	80
Figure 59 Plot of Frame Number vs. Estimation Error with respect to Maximum Target Points Number	81
Figure 60 Estimated Path for Linear Motion using ODSA	83
Figure 61 Estimated Path for Linear Motion using Kalman	84
Figure 62 Estimated Path for Circular Motion using ODSA	86
Figure 63 Estimated Path for Circular Motion using Kalman	86
Figure 64 Plot of Angular Velocity per frame $w(k)$ with respect to time k	87
Figure 65 Estimated Path for Circular Motion with Changing Angular Velocity using ODSA	87
Figure 66 Estimated Path for Circular Motion with Changing Angular Velocity using Kalman.....	88
Figure 67 Estimated Path for Parabolic Motion using ODSA	89
Figure 68 Estimated Path for Parabolic Motion using Kalman	90
Figure 69 Estimated Path for Elliptic Motion using ODSA.....	91
Figure 70 Estimated Path for Elliptic Motion using Kalman.....	92
Figure 71 Estimated Path for Changing Maneuver Motion using ODSA.....	93
Figure 72 Estimated Path for Changing Maneuver Motion using Kalman.....	94
Figure 73 Plot of Minimum Tolerable Difference of Target and Environment Mean Intensity vs. Noise Variance for ODSA and Kalman Filter.....	94

CHAPTER 1

INTRODUCTION

Most target tracking algorithms focus on large targets, namely, targets large enough for traditional automatic target recognition with a single frame of data. In addition, SNR values of interested targets in these algorithms are high enough to extract required information such as edge detection. The aim of this thesis is to develop and implement a new visual dim single target tracking algorithm in low observable and dense environments with a stationary camera looking at a fixed region of interest. The target to be tracked has in general a size of maximum 100 resolution elements and an SNR value less than 2 dB. The related application areas can be military and civilian depending on the targets to be tracked. The image pixels are assumed to be gray scaled and each of them is affected by Additive White Gaussian Noise with known statistics. The statistics of AWGN can be estimated by different parameter estimation methods but these methods are out of the scope of this thesis. The algorithm and simulation of environment is implemented by *MATLAB* programming language.

In Chapter 2, *Dynamic Vision* [2] is represented, an image processing technique developed for detecting very small and dim targets. It is based on *Likelihood Ratio Testing, Labeling of Super pixels, Label Destruction and Centroids Mapping, Maximization of Distribution Density Functions, Maximum Histogram Size Filtering and Range and Maneuver Filtering* methods. The output of dynamic vision algorithm is the observation of target centroids on each frame. The effect of the noise at the output is the target false alarms and centroids coordinate variation from the real target path.

In Chapter 3, *Optimum Decoding Based Smoothing Algorithm* [1] (ODSA) is represented. This algorithm obtains a trellis diagram for the target motion and estimates the target path by using the *Viterbi Decoding Algorithm* supporting both linear and nonlinear models. The input of ODSA for dim target tracking is the output of dynamic vision algorithm given in Chapter 2. The use of ODSA in this thesis is to estimate the target path given observation points of target centroids on each frame.

In Chapter 4, simulation results of the tracking algorithm are given, which includes the performance under realistic conditions, the advantages and disadvantages of the algorithm, the effects of the system parameters due to environment, target, risk and ODSA parameters.

In Chapter 5, the conclusion of the thesis is given.

CHAPTER 2

DYNAMIC VISION

A vision system has to deal with moving and changing objects, changing illumination, and changing view points. The input to a dynamic scene analysis system is a sequence of image frames taken from a changing world. The camera used to acquire image sequence may also be in motion. In this thesis the camera is assumed to be stationary, looking to the same region continuously. This kind of camera usually has civil and military applications for ground and air observation systems in very low SNR situations that can be due to hard weather conditions, poor quality of image sensing or far pursuit applications like satellite image processing. Calibration of estimated target path has to be done with respect to camera movement. Each frame represents an image of the scene at a particular instant in time. The changes in the pixel values of the scene may be due to the motion of objects, environmental or sensor noise on the camera. The aim of the dynamic vision system is to characterize the motion and recognize moving objects. There are four possibilities for the system. [2]

- 1) Stationary Camera-Stationary Objects (SCSO)
- 2) Stationary Camera-Moving Objects (SCMO)
- 3) Moving Camera-Stationary Objects (MCSO)
- 4) Moving Camera-Moving Objects (MCMO)

In many applications, it may be enough to process a single image to obtain the necessary information. In this thesis, since environment intensity spectrum is assumed to be very close to moving object intensity spectrum, the single image processing methods are not applicable to characterize the motion. To give an application example, in foggy and rainy hard weather conditions, it is hard to detect moving objects far away from the stationary camera. In such cases, the object is buried into the noise and impossible to detect by single image processing techniques. Instead, comparison of two or more frames is necessary to detect motion.

To give an example of intensity spectrums in grayscale, the environments mean intensity maybe around 100 with noise variance 15 when the target intensity is around 115. Figure 1 and Figure 2 shows the pixel intensity spectrum of the target and background. In this example it is assumed that target is formed from one characterized piece. In general, different-featured target pieces may form a single target.

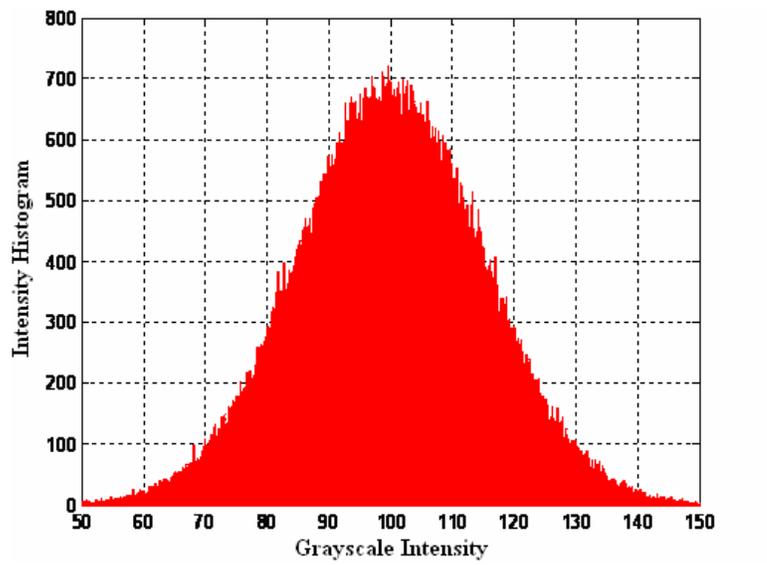


Figure 1 Background Pixel Intensity Spectrum, $N(100, 15)$

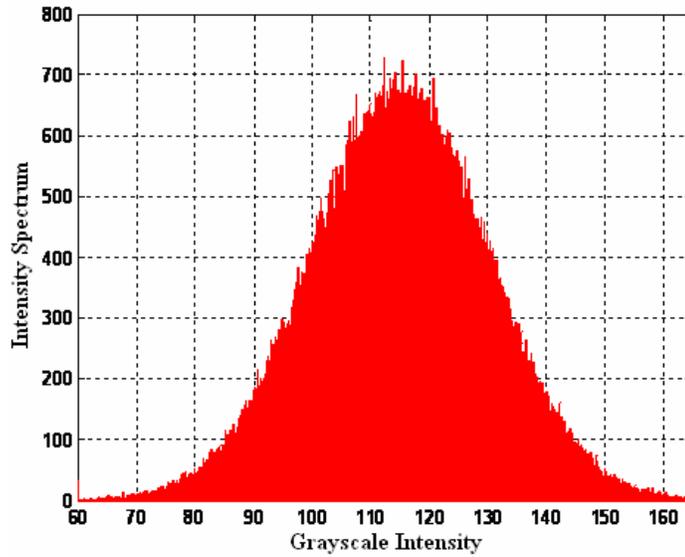


Figure 2 Target Pixel Intensity Spectrum, $N(115, 15)$

It is also possible that the background intensity spectrum can differ in different local areas on the frame. However, general characteristics of the distribution is assumed to be Gaussian in all sub regions, i.e. every pixel of the camera is exposed to Additive White Gaussian Noise with zero mean and known variance. This assumption covers almost all of the real world applications.

The moving objects are called as dim when a threshold value for single image cannot be used to characterize the object properties. Although sequence of frames gives much more information to aid in understanding a scene, it brings together significantly increasing processing time. Fortunately, research in dynamic scene analysis has shown that the recovery of information in many cases is easier in dynamic scenes than in static scenes.

In dynamic scene analysis, SCMO scenes have received the most attention. MCSO and MCMO scenes are very important in navigation applications. MCMO is the most general and possibly the most difficult situation in dynamic scene analysis.

Dynamic scene analysis has three phases:

- 1) Peripheral
- 2) Attentive
- 3) Cognitive

The peripheral phase is concerned with extraction of approximate information which is very helpful in later phases of analysis. This information indicates the activity in a scene and is used to decide which parts of the scene need careful analysis. The attentive phase concentrates analysis on the active parts of the scene and extracts information which may be used for recognition of objects, analysis of object motion, and preparation of a history of events taking place in the scene or other related activities. The cognitive phase applies knowledge about objects and other application dependent concepts to analyze the scene in terms of the objects present and the events taking place.

The input to a dynamic scene analysis system is a frame sequence, represented by $F(x, y, t)$ where x and y are the spatial coordinates in the frame representing the scene at time t . The value of the function represents the intensity of the pixel.

The Figure 3 shows an example of a dim air target in a foggy environment displayed by a mounted stationary camera on the ground. The target is surrounded by a square on the upper left side.

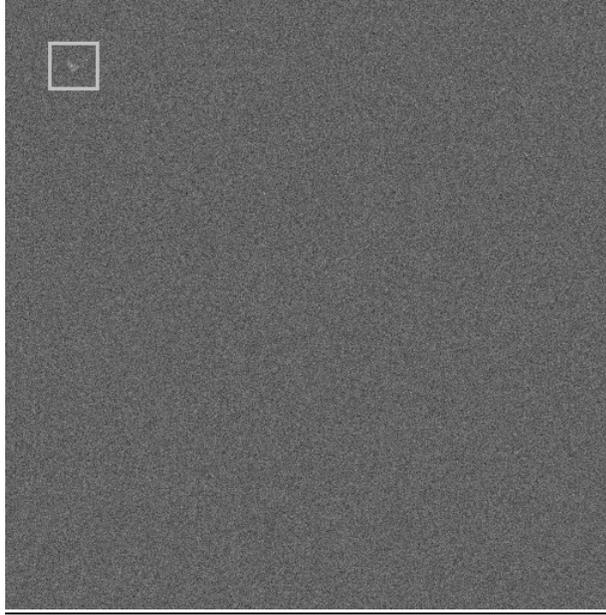


Figure 3 Example of a dim air target in a foggy environment

2.1 Change Detection

Detection of changes in two successive frames of a sequence is a very important step for many applications. Any perceptible motion in a scene results in some changes in the sequence of frames of the scene. A good quantitative estimate of the motion components of an object maybe obtained if the motion is restricted to the 2-D plane. In this thesis, motion is assumed to be parallel to the image plane. 3-D motion estimation is not the topics of this study. To decide whether a pixel value is changed due to motion, taking difference of pixel values one by one of two successive frames is a simple but most useful technique in many applications. However, since the aim of this study is to detect and track dim targets, difference picture method is not applicable for change detection of moving objects in very noisy environments. To make change detection more robust, intensity characteristics of groups of pixels at the same location in two frames maybe compared using either a statistical approach or an approach based on the local approximation of intensity distributions.

A straightforward domain independent method for comparing regions in images is to consider corresponding areas of the frames. These corresponding areas are the super pixels formed by pixels in no overlapping areas comprising of m rows and n columns. The values of m and n are selected to compensate for the aspect ratio of the camera. Therefore, a frame partitioned into disjoint super pixels is considered. Comparison is based on likelihood ratio test and the related calculation is given below:[2]

$$\lambda = \left(\frac{\frac{(\mu_1 - \mu_2)^2}{4} + \frac{(\sigma_1 + \sigma_2)}{2}}{\sigma_1 \cdot \sigma_2} \right)^2 \quad (2.1)$$

where μ_1 and μ_2 denote the mean intensity gray values and σ_1 and σ_2 denote the gray intensity variance for the super pixels at the same location of the camera in two successive frames.

After this computation, one can use a threshold value to decide whether a super pixel is changed or not. If λ is greater than the threshold, the algorithm decides that super pixel is changed and signs it. The threshold value is strongly related with SNR value of the target.

In this thesis, the SNR value of the target is defined as:

$$SNR_T = 10 \log \frac{\mu_{target}^2 + \sigma_{env}^2}{\mu_{env}^2 + \sigma_{env}^2} \quad (2.2)$$

where μ_{env} is the environment mean intensity, σ_{env} is the environment noise intensity variance and μ_{target} is the target mean intensity.

There are also different SNR definitions based on the intensity characteristics. One of them is given below [19]:

$$SNR_{motion} = \frac{\mu_{target} - \mu_{background}}{\sigma_{background}} \quad (2.3)$$

where μ_{target} is the average motion energy of pixels in the ground truth target region. Likewise $\mu_{background}$ is the average of motion energy of pixels in the ground truth region and $\sigma_{background}$ is the standard deviation of motion energy of pixels in the ground truth background region.

A measure of motion energy is estimated by considering a circular region centered at that pixel and computing the sum of contributions from individual pixel intensities in the region. Thus, for a given pixel location, the average motion energy computed from the circular region is:

$$e = \frac{1}{N} \sum_{i=1}^N (u_i^2 + v_i^2) \quad (2.4)$$

where the summation is carried out over the circular region, the index i specifies individual pixel location in the region and N is the total number of pixels in the region. This kind of computation makes use of non-directional nature of the circular region which makes it a robust choice to account for unknown target motion. However, this method is used for SNR of the motion and varies in time. Therefore, Equation 2.2 is chosen as the target SNR definition.

In very low SNR values, low threshold values may cause unnecessary false alarms of change detection and give false motion directions. Therefore, robust change detection is necessary but not enough technique for tracking dim targets.

The performance of the likelihood ratio test can be increased significantly by using facets and quadratic surfaces to approximate the intensity values of pixels belonging to super pixels. These higher order approximations allow for better characterization of intensity values and result in more robust change detection.

2.2 Connection Labeling Of Changing Super pixels

After detection of change at super pixels, grouping of these super pixels is necessary to identify different target candidates. It is important to mention that very small targets whose sizes are less than “ $m \times n$ ”, i.e. super pixel size, cannot be detected. Similarly, targets that have very low velocities less than super pixel size cannot be detected. Therefore, according to the target characteristics desired to be tracked, an optimum super pixel size can be determined.

Small super pixel size brings small and slow target motion detection in low SNR situations while it brings together extra computational load to the algorithm.

Labeling of the binary difference matrix formed by LRT Change Detection can be defined as grouping connected changing super pixels with respect to the connection styles below:

- 1) 4-Connection
- 2) 8-Connection

4-connection means a super pixel is in the same label with its upper, lower, right and left neighbors. For 8-connection, its upper/lower and right/left neighbor super pixels are also included [15]. 8-connection gives more information about the target shape but is not applicable in low SNR detection. Local area of the target gets bigger with unnecessary noise related changing super pixels as it can be observed from Figure 5.

The

Figure 4 shows 4-Connection and 8-Connection directions used to find labels.

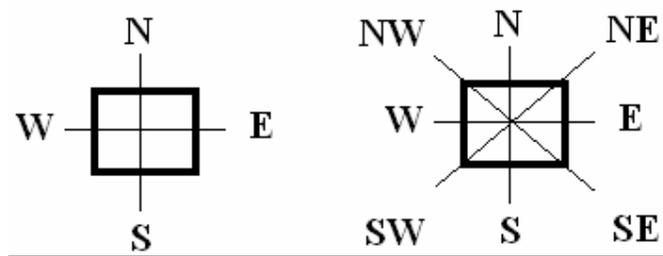


Figure 4 4-Connected and 8-Connected Labeling Directions

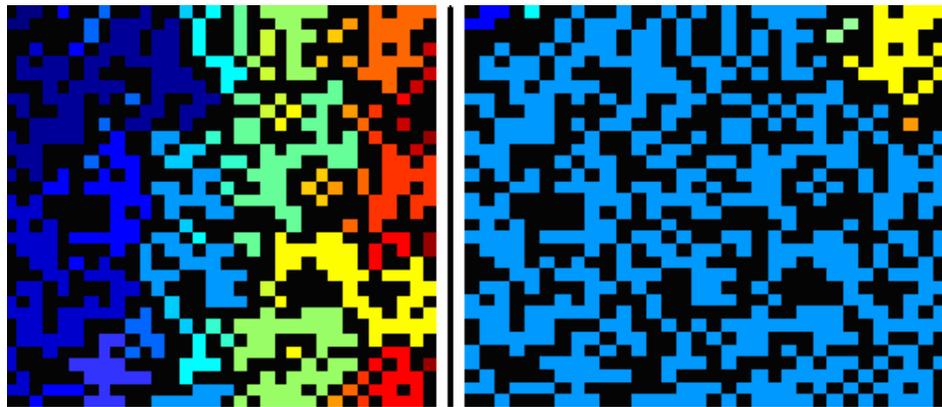


Figure 5 4-Connected and 8-Connected Label Matrix with SNR: 0.81

The pixels with the same color are in the same label and processed together. It is observed that the real target cannot be detected by single image processing techniques. The low SNR results in shading to real targets on the whole label matrix. Furthermore, 8-Connected labeling jams the label matrix and hides the real target. Therefore, in dim target applications 4-Connected labeling should be used.

2.3 Label Destruction and Centroids Mapping

Labels show candidate group movements at the scene. Analyzing labels includes extracting real target hidden in the label, mapping of the label to the next frame and filtering false alarms. Detecting changing super pixels in one label due to dim target motion is not necessary since the aim of the algorithm is to find motion not the target components. The algorithm doesn't search target pieces, rather it finds possible vector movement of the label.

Notice that all label movement doesn't give the real target motion and may cause false results. Therefore, instead of all label analysis with each super pixel, labels are divided into subgroups, i.e. groups of super pixels, and these subgroups are searched for vector motion. Notice also that vector motion is not required in its own label but also jump to other labels. Therefore, the focus area of the analysis must be at least target velocity. A safety region is necessary for this aim. One can give the entire frame as the focus area also. Destruction of label also decreases unnecessary noise clouds that spread all around the frame. The method to destruct the label can be in similar way as the super pixel creation, but since the label is a quadratic surface, it's meaningful to not increase the subgroup size. However, one super pixel is not enough to find real target movements since it doesn't give the real target motion characteristics, which is "If two neighbor super pixels are pieces of a target, they map to same or close location in the next frame." Therefore, minimum expected target area is used as the subgroup size. Changing super pixels that result from noise are highly probable to map to different locations.

The aim of the centroids mapping is to find new candidate positions of each element of the subgroup in the next frame by using LRT Comparison and labeling. Figure 6 illustrates the centroids mapping of the target and noise related two neighbor super pixels.

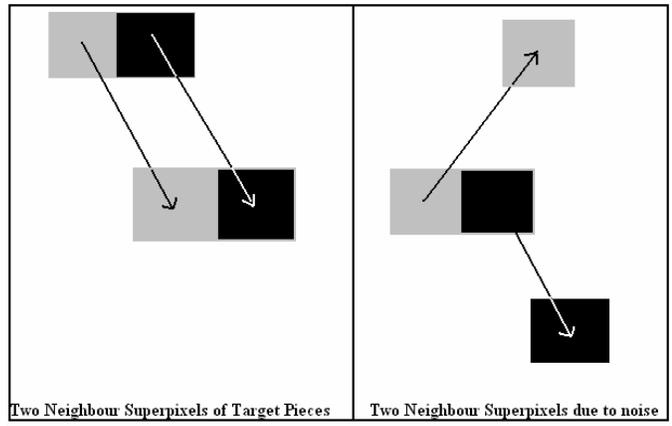


Figure 6 Illustration of target and noise related neighbor super pixels

Figure 7 shows an illustration of centroids that result from a label which includes the real target without destruction on it. The points that gather around the same center are the candidate target vectors. The points that become distinct from the others will be eliminated. Notice that noise related centroids are very probable to cause false alarms.

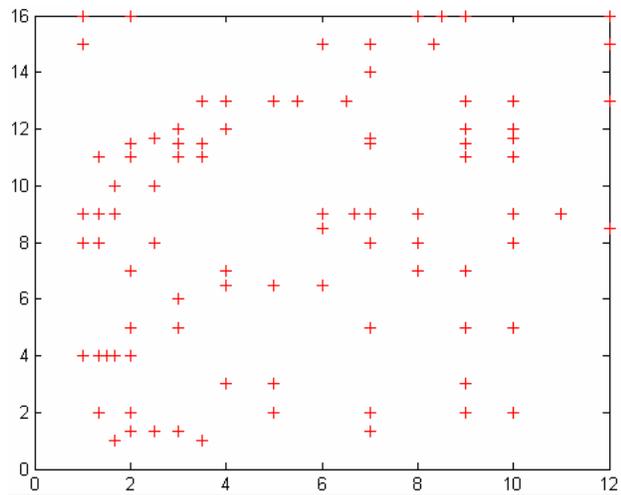


Figure 7 Centroids Distribution without Label Destruction

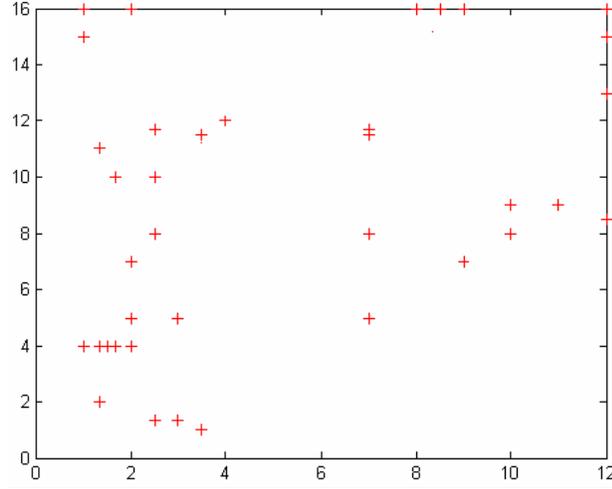


Figure 8 Centroids Distribution with Label Destruction

In Figure 8, it is clear that with label destruction, false alarms are eliminated considerably.

The mathematical definitions for labeling and destruction are given below:

Definition 1: Let D_n be the binary difference matrix whose elements are decided by LRT Change Detection method in n^{th} and $(n-1)^{th}$ and image frames. An element of this matrix $d_{i,j}$ is defined by: [2]

$$d_{i,j} = \begin{cases} 0 & \lambda \leq \tau & i = 1, 2, \dots, m \\ 1 & \lambda > \tau & j = 1, 2, \dots, k \end{cases} \quad (2.5)$$

where λ is the calculated LRT value of the super pixel which is given by the Equation 2.1, τ is the threshold value for comparison, $m \times k$ is the fixed size of the rows and columns in image frames.

Definition 2: Let L_r be the set of coordinates of the r^{th} label which is decided by using 4-connected labeling in the set D_n . An element of this set l_i is defined by:

$$L_i = \{(x_i, y_i) \mid D_n(x_i, y_i) = r\} \quad (2.6)$$

where (x_i, y_i) are the Cartesian coordinate values of the super pixels in the set D_n .

Definition 3: $S_{j,r}$ is the j^{th} subgroup of the label L_r , which is formed of groups of superpixels (x_i, y_i) that are interconnected with a rule ξ_i . This rule can be in shape or size manner geometrically.

From this definition, $S_{j,r}$ can be written as:

$$S_{j,r} = \left\{ \forall (x_i, y_i) \in S_{j,r} \mid \xi(x_i, y_i) \equiv \xi_i \right\} \quad (2.7)$$

where (x_i, y_i) is the Cartesian coordinates of the superpixels in D_n .

To give an example for ξ_i :

$$\xi_i : ((x_i - r_x)^2 + (y_i - r_y)^2) \leq \tau^2 \quad (2.8)$$

where (r_x, r_y) is the centre coordinates of the circle with radius τ .

The destruction method used in the algorithm is based on rectangular pieces of the bounding box of the label. If a piece includes a target element, it is evaluated for centroid mapping, otherwise ignored.

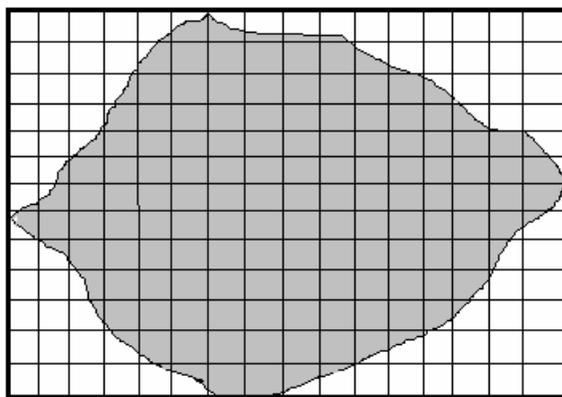


Figure 9 Illustration of Rectangular Label Destruction in the Bounding Box

2.4 Maximization of Distribution Density Functions

Every destructed label may give different, unrelated motion direction. This may be due to different targets in the label or false alarms. Subgroups of the label are analyzed to find their most probable direction. First of all, mapping of each superpixel of a subgroup in the next frame has to be done by the method similar in likelihood ratio test which is used for change detection. However, at this time it's necessary to find similar statistical super pixels instead of searching dissimilarity in the superpixels. For each method, the same threshold value can be used due to same noise variance. Mapping of each super pixel gives many candidates of vector movements in terms of the centroids. Therefore, a selection method along mapping vectors has been offered. At this time, distribution density functions take very important role. Distribution density functions are used to find most probable vector movement of a subgroup in the searched label. The input to the distribution density function is the centroids of the candidate mapping labels. It is important to notice that the distribution density functions mentioned in this thesis is not related with the probability density functions which are used to characterize the probability of random variables.

The probability of one centroid is dependent on how close the other centroids gather around it. Distribution density functions are symmetric around the origin in both x and y directions and monotonically decreasing. Notice also that distribution density functions should have an optimum bandwidth with respect to

the noise variance in the environment. The bandwidth definition can be in different ways in terms of the energy percentage that fall into it.

The Figure 10 and Figure 11 show three dimensional graphics of different types of distribution density functions in two axis with different bandwidths.

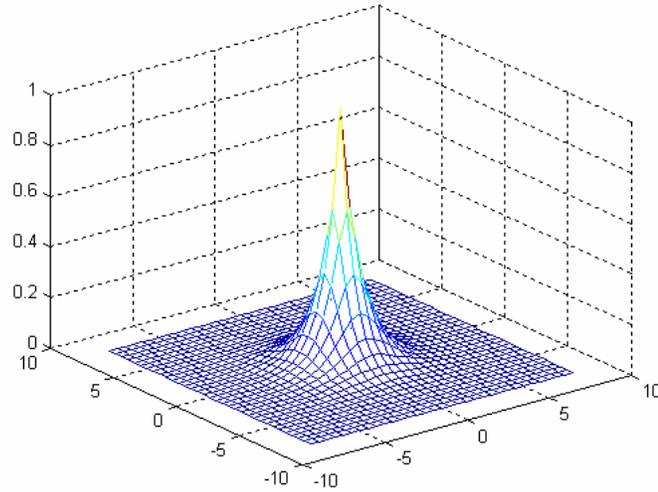


Figure 10 Example of Exponential Distribution Density Function: e^{-r} with " $r^2 = x^2 + y^2$ "

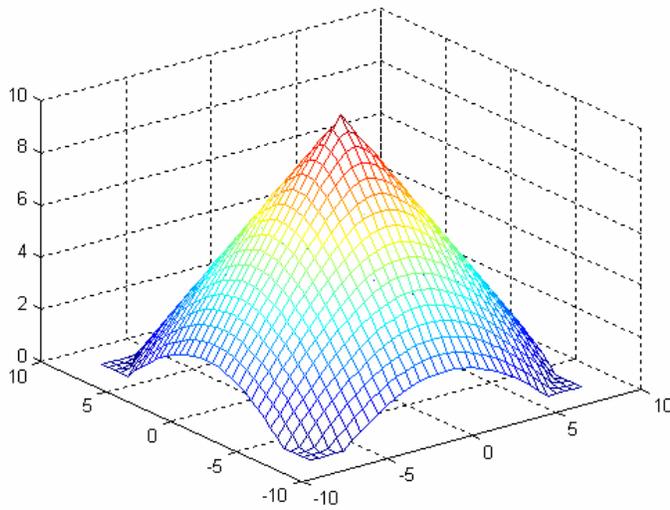


Figure 11 Example of Linear Distribution Density Function: $\text{Max}(10, 10-r)$ with " $r^2 = x^2 + y^2$ "

Definition 4: Let M_t be the set of total mapping centroid coordinates of the subgroup S_t and m_p be the subset of mapping centroid coordinates of the single superpixel in this subgroup.

$$\bigcup_p m_p = M_t \quad \text{where } p = 1, 2, \dots, n_s \quad (2.9)$$

and n_s is the number of elements in the subgroup S_t . It is important to mention that m_p sets that result from the elements of the subgroup S_t don't have to be mutually exclusive.

Definition 5: Let $\psi_{x,y}$ be the distribution density function (DDF) of the mapping centroid (x, y) given by the equation below:

$$\psi_{x,y} = \sum_n \psi(x_n, y_n) \quad n = 1, 2, 3, \dots, m_s \quad (2.10)$$

where, n is the centroid index number excluding the index number of (x, y) in M_t which has total number of elements m_s . It is also possible to filter out number of centroids instead of evaluating all of them, since DDF take very low accumulated values far away from the origin.

M_t is the observation set which includes necessary and unnecessary mapping centroids. The aim of DDF is to filter out noise related unnecessary mapping points. In the area of mapping centroids, we can find a region that has the maximum number of hit points with minimum entropy. The entropy of a point in the area is defines as the summation of distances of the all points to the related point.

$$E\{(x_i, y_i)\} = \sum_{i=2}^n d_i \quad n = 1, 2, \dots, k \quad (2.11)$$

in the set of (x_i, y_i) coordinates with total number of k elements.

The combination of minimum entropy with maximum number of hit points is given by DDF. As the number of centroids gets bigger and entropy gets smaller, DDF output increases in the same manner.

It is important to mention that coordinates of maximum likelihood point in the area does not have to be one of the elements in M_t . However, since the computational load for optimization increases very much and the distance error when one of the M_t elements is chosen is very small, it is more useful to search the maximum likelihood centroid point in the set M_t . The point found from DDF maximization is the centroid of the superpixels of the subgroup analyzed which have similar intensity characteristics.

In order to find maximum likelihood mapping centroids point of the subgroup S_t , it is necessary to define the probability of each element of the set M_t .

Definition 6: Let $P_{M_t}(x, y)$ be the probability of mapping for the mapping centroid (x, y) in M_t .

$$P_{M_t}(x, y) = \frac{1}{\sum_{i=1}^n \psi_{x_i, y_i}} \cdot \psi_{x, y} \quad (2.12)$$

$\psi_{x, y}$ is directly related with the mapping probability. As the distribution of the mapping centroids gets closer to the mapping centroid (x, y) , the probability of vectoral movement of the subgroup m_p to that point gets bigger. The mapping centroid which maximizes Ψ is chosen as the maximum probable subgroup vectoral movement \bar{S}_{\max} .

$$\bar{S}_{\max} = \arg \max(P_{M_t}) \quad (2.13)$$

The Figure 12 shows graphical result when a frame part that includes the real target has been put into an exponential distribution density function.

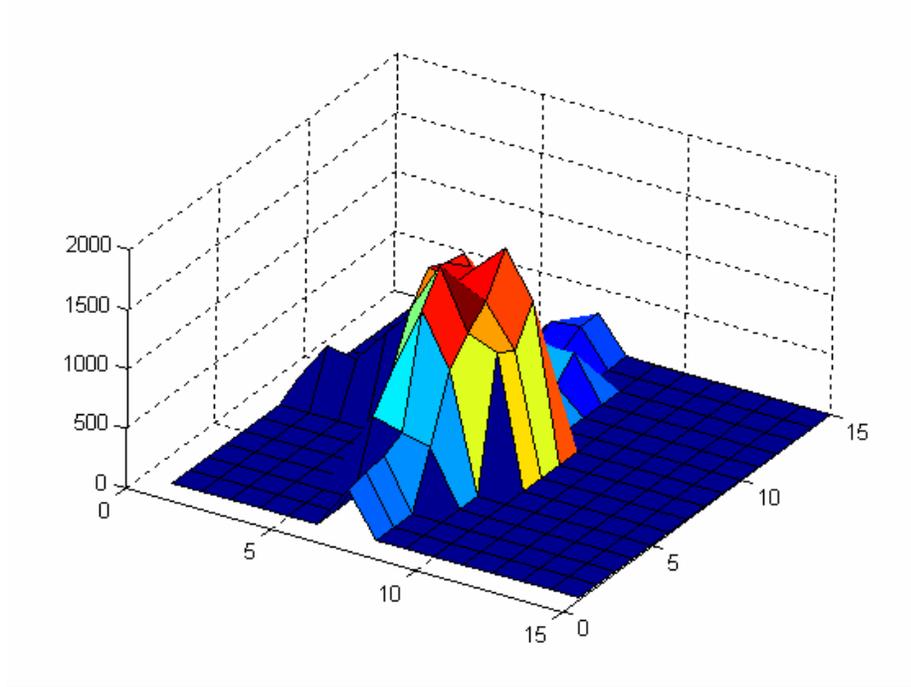


Figure 12 A real target that appears on DDF Maximization

It is clear that the regions that include the real target results in higher DDF values which are related with number of hits to the same region and minimum entropy.

To summarize DDF Maximization, each candidate centroids of the mapped labels of the subgroup elements are taken as the origin of DDF for the evaluation. All other centroids that fall in the bandwidth are put into DDF to get the final summation value. The one that has the maximum value in DDF is taken as the maximum likelihood subgroup vector.

It is important to notice that, if there does not exist enough superpixels that come from the target, DDF maximization gives false alarms. However it is possible to identify different targets in different locations by Maximum Histogram Size Filtering which is given in the next chapter.

2.5 Maximum Histogram Size Filtering

The next step in the algorithm is to find different target candidates by using subgroup mapping points \bar{S}_{\max} determined by DDF maximization. This step is necessary to differentiate different targets that gather around the same centroid. It's also possible that noise clouds result in subgroup mapping points as if they are target elements. The mapping points of this kind of subgroups may be in the real target centroids, however it is not important for the algorithm to differentiate them from the real target vector movements because the algorithm uses the final filtered vector movement as the start point that result from subgroups for the next iteration. It is necessary to emphasize that the mapping points are the only important and enough result of the DDF maximization. The mission of finding target start point is related with the previous iteration of the algorithm. It takes importance only for the initial state but the algorithm uses ODSA for estimation. From the point of ODSA, the maximum metric in the last is the result of the estimated path not the initial point.

The method used to differentiate different target groups strongly depends on determining different target mapping point clouds. The resolution to differentiate different targets can be increased with a good selection of grouping technique. This technique depends also on maximum target size. Maximum histogram size filtering has been developed for this aim.

This filtering method generates new classes of centroids and groups them by going back from the point that has the maximum value in the member histogram.

The Figure 13 shows an example of differentiation of different targets on the image with maximum histogram size filtering.

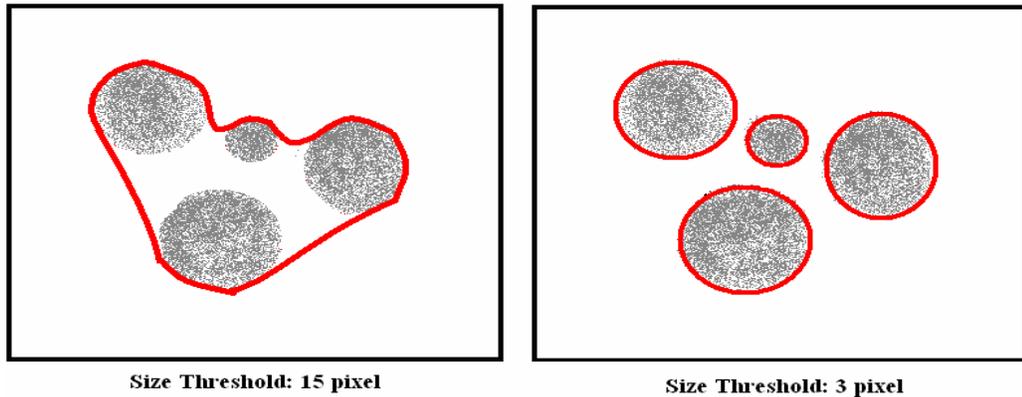


Figure 13 Differentiation of different targets for different size thresholds

The general flowchart of the Maximum Histogram Size Filtering is given below:

- Step 1. Create a group for each centroid. Each group consists of other centroids which are closer than the size threshold value.
- Step 2. Find the number of elements for each group
- Step 3. Assign the group which has the maximum number of elements as a target.
- Step 4. Kill the group by zeroing the histograms of the centroids included in it.
- Step 5. Go back to Step 3 until all of the groups are killed.

2.6 Range and Maneuver Filtering

The target candidates found by maximum histogram size filtering method can be in different directions and velocities with respect to last vector movement. Therefore, the algorithm takes maximum possible range and maneuver index acceptable for a target by using Equations 2.13 and 2.14.

Range and maneuver index definitions are related to difference between two successive frames. The sampling time of the image sensor is important from this point of view. As the sampling time of the image sensor gets bigger, it's not reliable to filter out targets that fall into maximum range and maneuver.

Let r_m and m_i be the maximum target movement range and maneuver index of the tracked target. The centroids of possible target movements, C_m , are filtered by the following equations:

$$m_i \geq f \left(\left| \tan^{-1} \left(\frac{y_{m,k+1} - y_{m,k}}{x_{m,k+1} - x_{m,k}} \right) - \tan^{-1} \left(\frac{y_{m,k} - y_{m,k-1}}{x_{m,k} - x_{m,k-1}} \right) \right| \right) \quad (2.14)$$

$$r_m \geq g \left(\frac{\sqrt{(y_{m,k+1} - y_{m,k})^2 + (x_{m,k+1} - x_{m,k})^2}}{\sqrt{(y_{m,k} - y_{m,k-1})^2 + (x_{m,k} - x_{m,k-1})^2}} \right) \quad (2.15)$$

where k indicates the related frame numbers. (x, y) is the Cartesian coordinate of the centroid. $f(\cdot)$ is the maneuvering filter function and $g(\cdot)$ is the range filter function. These functions are based on the system state transition model. $f(\cdot)$ defines the angle change and $g(\cdot)$ defines the relative velocity change, i.e. acceleration, of the target between three successive frames. Based on the system model defined, these functions are used to filter out non possible target movements.

Notice that “f” and “g” functions are defined adaptively, that is to say the inputs of these functions in the next step are the outputs of the last step. Therefore, slow target transition changes along the frames can be detected by changing the acceleration and maneuver parameters with respect to the last observed data.

It is also possible to use Kalman or Extended Kalman Filters to predict the current and future states of the target. However, it is found to be enough to use second order adaptive filter to eliminate non possible target movements. The order of the filter is related with the maximum frame number difference between the current and past frames, which are used in “f(.)” and “g(.)” functions. The Equation 2.14 and 2.15 use second order adaptive filter for this aim.

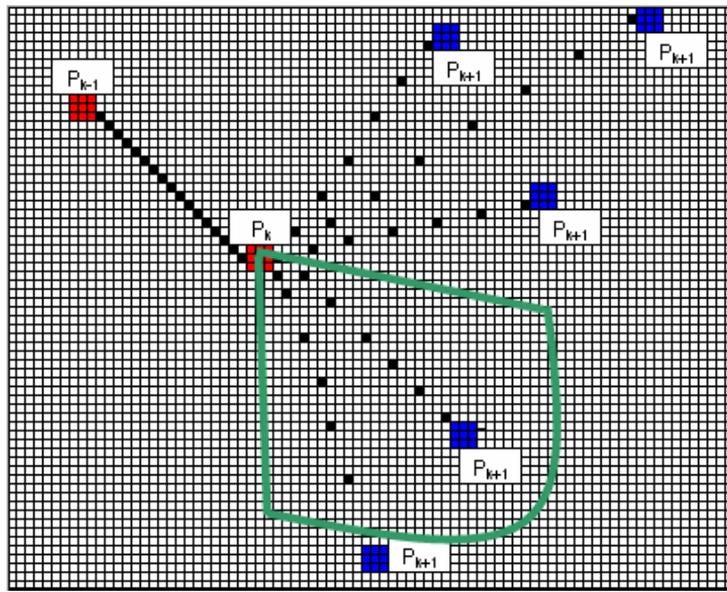


Figure 14 Example of target range and maneuver filtering

The surface that is enclosed by the green line is the possible target movement region for a linear state transition. Notice that system can be defined in any manner, i.e. constant or variable acceleration and maneuver parameters.

Below is the list of possible target state transition models:

1. Constant Acceleration Constant Maneuver
2. Variable Acceleration Constant Maneuver
3. Constant Acceleration Variable Maneuver
4. Variable Acceleration Variable Maneuver

The algorithm cannot find the targets which has acceleration and maneuvering bigger than the limits. Even though giving more tolerance for the limits seems to increase probability of real target cover area, it is more realistic and useful to limit them to prevent false selection of the real target along the candidates. From this point of view, it can be said that there exists maximum acceleration and maneuver limit for a given SNR value to extract the path.

2.7 Position and Direction Calibration

The algorithm uses center windowing method to focus on target. The method is based on locating the last observed target position at the center of the window with a predetermined size which is the maximum tolerable target velocity. The superpixels in the frame that are not in this window are ignored for the analysis. The reason of center windowing is to increase the time performance of the algorithm and prevent false alarms that cannot be related with focused target. This relation is about the maximum target velocity tolerable in the system. There exists a tradeoff between the performance of the algorithm and risk of missing the target that is not in the focus window.

Centering the window for focusing on the target to be tracked necessitates position and direction calibration. Dim target tracking algorithm finds the maximum probable target movement in the focused window. This vector has to be mapped into real main frame of observation.

A reference of the target with respect to the focus window is used for calibration. This reference is the left-upper coordinates of the focus window at the main frame. At each observation interval, the focus window jumps to the new target-centered position and new observation matrix is given as the input to change detection. After dim target tracking algorithm finds the relative target movement with respect to the window position and then maps the target vector to the stationary camera coordinates.

Figure 15 shows the vector sum of the relative target movement and focus window vectors.

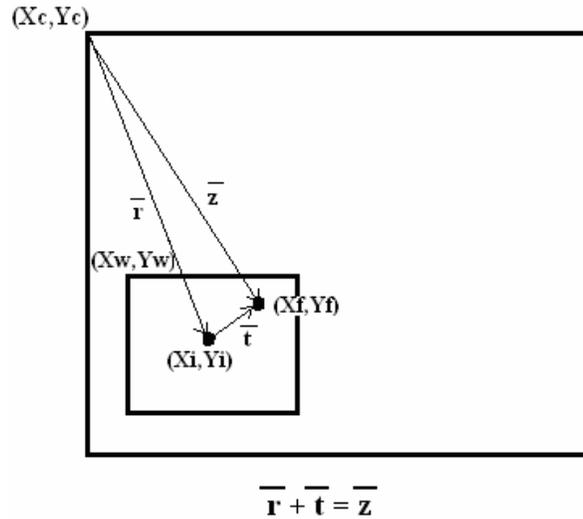


Figure 15 The calibration of the real target position and direction

where “r” is the vector between the center of the focus window and main frame left-upper corner, “t” is the relative motion of the target with respect to focus window, “z” is the final calibrated target position vector with respect to the left-upper corner of the main frame.

- (X_c, Y_c) : Cartesian coordinates of the main frame left-upper corner
- (X_w, Y_w) : Cartesian coordinates of the focus window left-upper corner
- (X_i, Y_i) : Cartesian coordinates of the center of the focus window, also the final observed target position
- (X_f, Y_f) : Cartesian coordinates of the relative vector movement of the target

2.8 Assignment of the Next Iteration Label

The final step of the dim target tracking algorithm is to provide continuity of the iteration by assigning the last observed target centroid to the most probable label in the next step. Due to the low SNR values, the point that is found to be the centre of the last label can no longer be seen by the algorithm as the centre of the next difference matrix label. The labels change their sizes and shapes in the next frame because the frames that are used for comparison have already been changed. Therefore it is necessary to match the target centroid with the most probable label. For this aim, the following criteria are presented:

Let Δr_l be the distance between the last observed target centroid and the centroid of the l^{th} label of the next difference matrix and A_l be the area of the label which is defined as the number of pixels included in the label.

The labels which have error tolerance of the last label area are filtered and the label which has the closest centroid to the target centroid is chosen as the next label for iteration.

Let label L_x be the final decided label for the iteration. L_x has the following properties:

$$L_x : \left\{ |A_x - A_s| \leq \varepsilon_s \cap \Delta r_x = \min(\Delta r_l) \right\} \quad l = 1, 2, \dots, N_l \quad (2.16)$$

where A_s is the area of the last analyzed label and ε_s is the maximum tolerable target size change between two successive frames. N_l is the number of labels in the binary difference matrix D_n .

2.9 Flowchart of the Dim Target Tracking Algorithm

The aim of the dim target tracking algorithm is to get reasonable target path observation from the image sequences given. The result of the algorithm has at least the distance error as much as the superpixel size. Furthermore, there occur big fluctuations along the path from the real target centroid due to the low SNR condition. Therefore, the output of the dim target tracking algorithm has to be put into a smoothing algorithms to find more reliable path estimation. The effect of the high noise variance reflects to the deviation of the centroid of the observed path from the real target path, which usually results in discontinuities in the path. The risk of missing the target centroid in the next frame gets bigger if the observed path is not smoothed. Furthermore, it increases the performance of the algorithm from the computation time point of view.

The Figure 16 shows the general flowchart of the dim target tracking algorithm and the relation of the algorithm with the Optimum Decoding Based Smoothing algorithm.

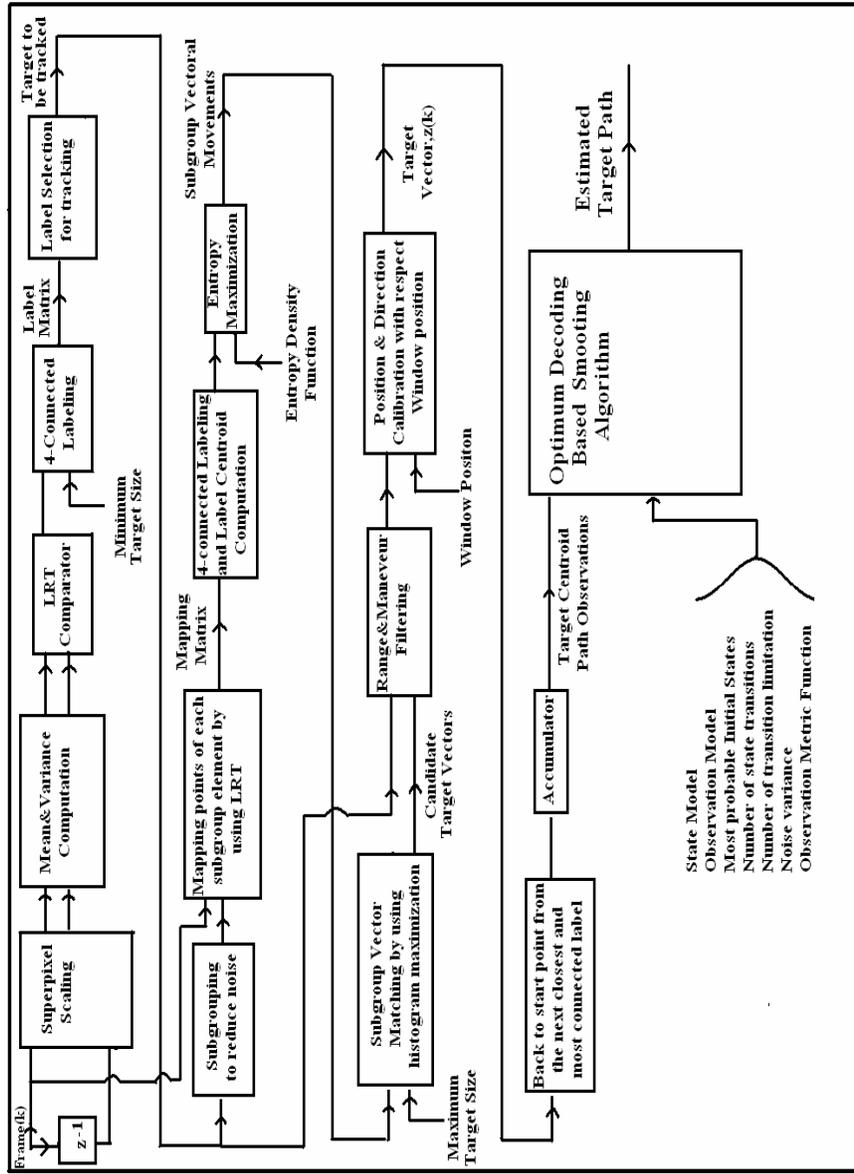


Figure 16 General Flowchart of the Dim Target Tracking Algorithm

CHAPTER 3

OPTIMUM DECODING BASED SMOOTHING ALGORITHM

Optimum decoding based smoothing algorithm is an estimation algorithm which can be applied for both linear and nonlinear estimation problems modeled as below:

$$\text{Motion model,} \quad x(k+1) = f(k, x(k), u(k), w(k)) \quad (3.1)$$

$$\text{Observation model,} \quad z(k) = g(k, x(k), v(k))$$

Where $x(0)$ is an $nx1$ initial state Gaussian distributed random vector (which determines the considered target location at time 0); $x(k)$ is an $nx1$ (target) state vector at time k ; $u(k)$ is a $qx1$ input at time k with known statistics; $w(k)$ is a $px1$ Gaussian distributed disturbance noise vector at time k with zero mean and known statistics; $v(k)$ is an $lx1$ Gaussian distributed observation noise vector at time k with zero mean and known statistics; $z(k)$ is an $rx1$ observation vector at time k . Time k is time $t_0 + kT_0$ where t_0 and T_0 are the initial time and the observation interval respectively. The random vectors $x(0)$, $w(k)$ and $v(k)$ are assumed to be independent for all k . The goal is to estimate the state sequence $\{x(0), x(1) \dots x(L)\}$ by using the observation sequence $\{z(1), z(2), \dots, z(L)\}$. [1]

3.1 Quantization of States and Transition Probabilities

Due to some analytical difficulties, calculations should be performed numerically. Let $x(k)$ be a random vector whose range is in the space R^n (n dimensional Euclidian space). Let us divide R^n into nonoverlapping subspaces R_i^n and assign a unique value x_{qi} to each subspace R_i^n , where the subscript q is quantization. [1]

Definition 3.1: A function $x_q(.) \triangleq Q\{x(.)\}$ is a quantizer for the state $x(.)$ if the following hold:

- 1) A function $x_q(.) \triangleq Q\{x(.)\} = x_{qi}$ whenever $x(.) \in R_i^n$; and
- 2) x_{qi} is unique for each R_i^n

Definition 3.2: The function $x_q(.)$ is the quantized state vector at time $(.)$, and its possible values are called quantization levels of the state $x(.)$.

Definition 3.3: Subspace R_i^n is called gate R_i^n .

Definition 3.4: The value x_{qi} is called the quantization level for the gate R_i^n .

Definition 3.5: The transition probability $\Pi_{jm}(k)$ is the probability that the state $x(k+1)$ will lie in the gate R_m^n when the state $x(k)$ is in the gate R_j^n ; i.e.,

$$\Pi_{jm}(k) \triangleq \text{Prob} \{ x(k+1) \in R_m^n \mid x(k) \in R_j^n \} \quad (3.2)$$

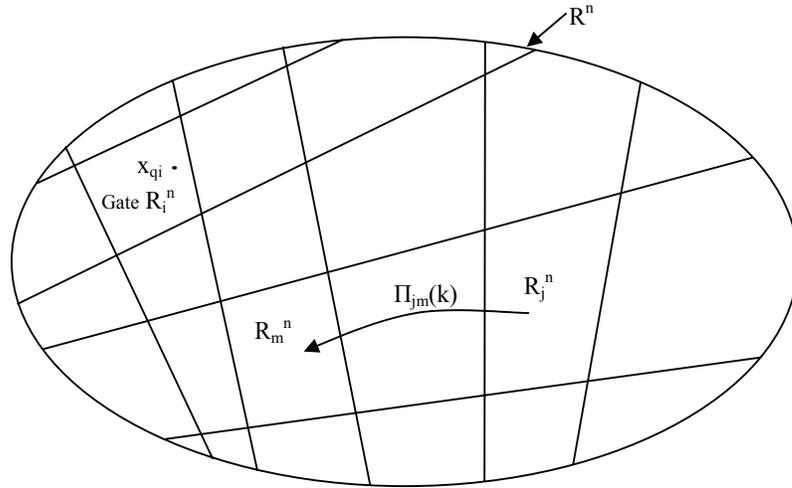


Figure 17 Quantization and transition probabilities

Since the initial state vector $x(0)$ and $w(k)$ are Gaussian distributed and the random vectors $x(0)$, $w(k)$, and $w(l)$ are assumed to be statistically independent for all k, l ; $x(k+1)$ and $x(k)$ are linear transformations of the Gaussian random vectors $x(0)$, $w(1)$, ..., and $w(k)$. Thus, $p(x(k))$ and $p(x(k+1)|x(k))$ are normal density functions. Therefore, the evaluation of $\Pi_{jm}(k)$ is difficult. For this reason, we define a finite-state model described in section 2.3 by approximating the disturbance noise vector $w(k)$ and the initial state vector $x(0)$ by discrete random vectors, and by quantizing $x(k)$ for all k . For this finite-state model, the transition probabilities can be easily calculated.

3.2 Approximation of an Absolutely Continuous Random Vector by a Discrete Random Vector

In order to find the optimum discrete random variable with n possible values that approximates an absolutely continuous random variable x with distribution function $F_x(\cdot)$, we must find a distribution function $F_{y_0}(\cdot)$ which minimizes the objective function $J(\cdot)$:

$$\begin{aligned} J(F_{y_0}(\cdot)) &= \min_{F_y(\cdot)} J(F_y(\cdot)) \\ &= \min_{g(\cdot)} J(g(\cdot)) \end{aligned} \quad (3.3)$$

Where

$$J(F_y(\cdot)) = \int_{-\infty}^{\infty} [F_x(a) - F_y(a)]^2 da \quad (3.4)$$

The aim is to find a step function $g_0(\cdot)$ which minimizes the objective function $J(\cdot)$:

$$\begin{aligned} J(g(\cdot)) &= \int_{-\infty}^{y_1} F_x^2(a) da + \int_{y_1}^{y_2} [F_x(a) - P_1]^2 da + \int_{y_2}^{y_3} [F_x(a) - P_2]^2 da + \dots \\ &+ \int_{y_{n-1}}^{y_n} [F_x(a) - P_{n-1}]^2 da + \int_{y_n}^{\infty} [F_x(a) - 1]^2 da \end{aligned} \quad (3.5)$$

$$g_0(x) = \begin{cases} 0, & x < y_{1,0}, \\ P_{i,0}, & y_{i,0} \leq x < y_{i+1,0}, \quad i = 1, 2, \dots, n-1 \\ 1, & x \geq y_{n,0}, \end{cases} \quad (3.6)$$

If $g_0(x)$ is a step function which minimizes (3.5), it must satisfy the following set of equations:

$$\begin{aligned}
 P_{1,0} &= 2F_x(y_{1,0}) \\
 P_{i,0} + P_{i+1,0} &= 2F_x(y_{i+1,0}), \quad i = 1, 2, \dots, n-2 \\
 1 + P_{n,0} &= 2F_x(y_n) \\
 P_{i,0}(y_{i+1,0} - y_{i,0}) &= \int_{y_{i,0}}^{y_{i+1,0}} F_x(a) da \quad i = 1, 2, \dots, n-1
 \end{aligned} \tag{3.7}$$

The discrete random variables which approximate the normal random variable with zero mean and unity variance are given in the References [20]. If the mean (μ) and the variance (σ) of the random variable are different than 0 and 1 respectively, the new discrete values are computed according to the mean and variance of the random variable by using the formula given in 3.8.

$$y' = \sigma \cdot y_{i,0} + \mu \quad P'_{i,0} = P_{i,0} \quad i = 1, 2, \dots, n \tag{3.8}$$

The Figure 18 shows the distribution functions of the absolutely continuous normally distributed random variable x with zero mean and unity variance; and the distribution of the optimum discrete random variable which approximates x with 8 possible values.

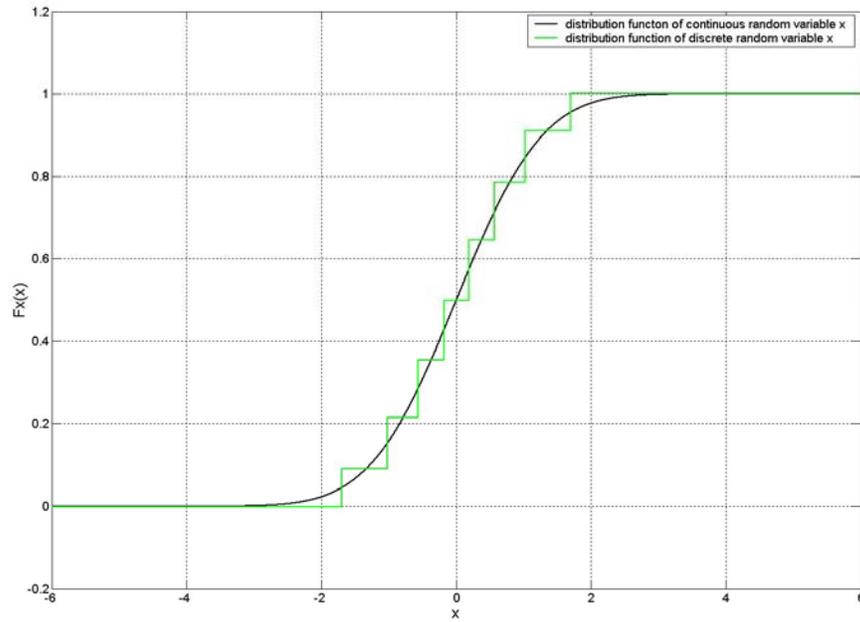


Figure 18 PDF of continuous random variable x and approximated discrete random variable

The y and p values of approximated x are given at Table 1.

Table 1 y and p values of discrete random variable with 8 possible values

	1	2	3	4	5	6	7	8
y	-1.6990	-1.0250	-0.5700	-0.1840	0.1840	0.5700	1.0250	1.6990
p	0.0922	0.1240	0.1394	0.1460	0.1460	0.1394	0.1240	0.0922

3.3 A Finite State Model for the Target Model

Gates are assumed to be generalized rectangles with origin R_0^n . The quantization levels for gates are assumed to be the center of the gates.

$$X_q(\cdot) \triangleq Q(x(\cdot)) = x_{qi} \quad \text{if } x(\cdot) \in R_i^n, \quad (3.9)$$

For each k the disturbance noise vector $w(k)$ is approximated by a discrete random vector $w_d(k)$ and the initial state vector $x(0)$ is approximated by a discrete random vector $x_d(0)$ as described in section 3.2. The number of discrete values for $w_d(k)$ and $x_d(k)$ are chosen such that $w(k)$ and $x(k)$ are satisfactorily approximated. Furthermore, by replacing $w(k)$ and $x(0)$ with discrete random vectors $w_d(k)$ and $x_d(0)$ respectively, and then quantizing the states by (3.9), the target-motion model is reduced to a finite state model.

$$X_q(k+1)=Q(f(k, x_q(k), u(k), w_d(k)), \quad (3.10)$$

The number of quantization levels for $k=0$ is the number of discrete values of the initial state vector $x(0)$, and the number of quantization levels for $k=1, 2 \dots L$ is limited because of the increase in the complexity of the algorithm. If the number of states were not limited, than the program computation time would increase dramatically with large k values.

The transition probability $\Pi_{jl}(k)$, which is defined by the conditional probability that the quantized state vector $x_q(k+1)$ will be equal to the quantization level x_{ql} for gate R_l^n , given that the quantized state vector $x_q(k)$ is equal to the quantization level x_{qj} for gate R_j^n is determined as follows:

$$\Pi_{jl}(k) \triangleq Prob \{ x_q(k+1)=x_{ql} \mid x_q(k)=x_{qj} \} \quad (3.11)$$

Assume that the $x_q(k)$ is equal to x_{qj} for gate R_j^n . The transitions from this quantization level to others are determined by $w_d(k)$ and the function $Q(f(k, x_q(k), u(k), w_d(k)))$. If $w_d(k)$ has m discrete values ($w_{d1}(k), w_{d2}(k), \dots, w_{dm}(k)$), then $x_q(k+1)$ can take at most m quantization levels. The probability of $\Pi_{jl}(k)$ is determined by the probability of $w_d(k)$ value (determined by corresponding p value) which results x_{ql} . If more than one values of $w_d(k)$ result x_{ql} than probabilities are added.

3.4 Approximate Observation Models

Since the target-motion model is reduced to a finite-state model which uses the quantized state vector $x_q(\cdot)$, the observation model is approximated as equation (3.12).

$$z(k) = g(k, x_q(k), v(k)) \quad (3.12)$$

The motion model can be represented by a trellis diagram as shown in Figure 19. When the trellis diagram is drawn from time 0 to L , the target motion occurs along one of the possible paths through the trellis diagram. Based on the observation sequence $z(1), z(2), \dots, z(L)$, the most probable path followed by the target is determined by using the *minimum error probability* criterion, which is a special case of *Bayes' criterion* in detection theory. Using this criterion reduces the problem of finding the path most likely followed by the target to a multiple hypothesis testing problem.

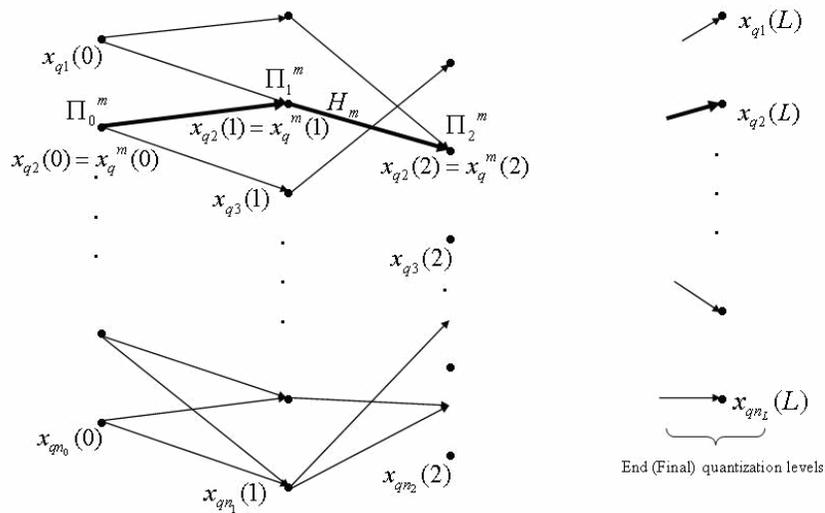


Figure 19 Trellis diagram for the target motion

3.5 Minimum Error Probability Criterion

Let the number of possible paths in the trellis diagram be M . Then, we define M hypothesis for the M possible paths. Using the minimum error probability criterion, we decide on one of the hypothesis to be the actual target path.

Let us develop a decision rule that assigns each point in the observation space D into M subspaces D_1, D_2, \dots, D_M . If the observation fall in the subspace D_i , we decide that the target (most likely) followed path H_i . Subspace D_i is called the decision region for hypothesis H_i . We must choose the decision regions in such a way that the overall error probability is minimized. The overall error probability, sometimes called Bayes' risk is defined by:

$$R \triangleq \sum_{j=1}^M \sum_{\substack{i=1 \\ i \neq j}}^M \left\{ \int_{z^L \in D_i} p(H_j) p(z^L | H_j) dz^L \right\} \quad (3.13)$$

where

- $p(H_j)$: Probability that the hypothesis H_j (path H_j) is true,
- $p(z^L | H_j)$: Conditional probability of the observation sequence z^L ($z(1), z(2), \dots, z(L)$) given that hypothesis H_j is true.

The optimum decision rule is

$$\text{Choose } H_i \text{ if } p(H_i) p(z^L | H_i) > p(H_j) p(z^L | H_j) \text{ for all } j \neq i, \quad (3.14)$$

3.6 Optimum Decision Rule for the Target Paths

Let us consider the motion model in equation (3.10) and the observation model in equation (3.12). Since the disturbance noise vector $w(k)$ is assumed to be independent of $w(j)$ and $x(0)$ for all $j \neq k$, the a priori probability of hypothesis H_i can be rewritten as:

$$p(H_i) = \prod_{k=0}^L \Pi_k^i, \quad (3.15)$$

where $\Pi_k^i = \text{prob}(x_q(k)=x_q^i(k)|x_q(k-1)=x_q^i(k-1))$, and $x_q^i(k-1)$ and $x_q^i(k)$ are the quantization levels for the gates in which the target lies at time $k-1$ and k respectively when it follows path H_i .

The function $p(z^L|H_i)$ in equation (3.14) can be rewritten as:

$$p(z^L|H_i) = \prod_{k=1}^L p(z(k)|x_q^i(k)) \quad (3.16)$$

Where $p(z(k)|x_q^i(k)) = p(z(k)|x_q(k) = x_q^i(k))$

Since $z(k)$ is a Gaussian distributed random variable, $z(k)|x_q^i(k)$ is also a Gaussian distributed random variable with mean $g(k, x_q^i(k), 0)$ and variance σ_v^2 . Therefore, $p(z(k)|x_q^i(k))$ can be computed according to the formula given in equation (3.17).

$$p(z(k)|x_q^i(k)) = \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp\left(\frac{-(z(k) - g(k, x_q^i(k), 0))^2}{2\sigma_v^2}\right) \quad (3.17)$$

Substituting equation (3.15) and (3.16) into the optimum decision rule of equation (3.14), we obtain the following:

Choose H_i if

$$\Pi_0^i \prod_{k=1}^L \Pi_k^i p(z(k)|x_q^i(k)) > \Pi_0^j \prod_{k=1}^L \Pi_k^j p(z(k)|x_q^j(k)) \quad (3.18)$$

for all $j \neq i$.

Since it is more convenient to perform summations than multiplications, and the natural logarithm function is monotonically increasing, taking the natural logarithms of both sides of the inequality in equation (3.18), we get the following:

Choose H_i if

$$\ln(\Pi_0^i) + \sum_{k=1}^L \{ \ln(\Pi_k^i) + \ln(p(z(k)|x_q^i(k))) \} > \ln(\Pi_0^j) + \sum_{k=1}^L \{ \ln(\Pi_k^j) + \ln(p(z(k)|x_q^j(k))) \} \quad (3.19)$$

for all $j \neq i$.

3.7 Optimum Decoding Based Smoothing Algorithm

The optimum decision rule for selecting the target path is the path in the trellis diagram which has the maximum probability. This can be handled by Viterbi decoding algorithm, which is the optimum decoding algorithm. The algorithm which obtains a trellis diagram for the target motion model, and which finds the path most likely followed by the target by using the Viterbi decoding algorithm is referred as the optimum decoding based smoothing algorithm. This method finds the most probable path by comparing the metric values of the quantization values of the states from time 0 to time L . Metric values are defined as below:

Definition 3.6: The metric, denoted by $MN(x_{qi}(0))$, of the initial node $x_{qi}(0)$ is defined by

$$MN(x_{qi}(0)) = \ln [prob(x_q(0) = x_{qi}(0))] \quad (3.20)$$

Consequently, $MN(x_q^m(0)) = \ln (\Pi_0^m)$.

Definition 3.7: The metric, denoted by $M(x_{qj}(k-1) \rightarrow x_{qi}(k))$, of the branch which connects the quantization level $x_{qj}(k-1)$ to the quantization level $x_{qi}(k)$ is defined by

$$M(x_{qj}(k-1) \rightarrow x_{qi}(k)) \triangleq \ln [\text{prob}(x_q(k)=x_{qi}(k)|x_q(k-1)=x_{qj}(k-1))] + \ln p(z(k)|x_{qi}(k)) \quad (3.21)$$

Definition 3.8: The metric of a path from time 0 to time i is the summation of the metric of the initial node from which the path starts and the metrics of the branches of which the path consists of.

The value of the input vector $u(k)$ used in the motion model is assumed to be constant from time $k=0$ to time $k=L$ and its probability density function $f_u(u)$ given in equation (3.22) is composed of discrete values with known probabilities.

$$f_u(u) = \sum_{i=1}^N P_i \delta(u - u_i) \quad (3.22)$$

Where

N : the number of possible values of $u(k)$,

u_i : the possible value of $u(k)$,

P_i : the probability of u_i .

The vectors $x(k)$, $w(k)$, $z(k)$, $u(k)$ and $v(k)$ are chosen to be one-dimensional for simplicity. Furthermore, the initial state $x(0)$, distribution noise $w(k)$, and the observation noise $v(k)$ are chosen to be normally distributed random variables with given mean and variance. The mean of $w(k)$ and $v(k)$ are assumed to be 0. The probability density functions of $x(0)$, $w(k)$ and $v(k)$ are given in equations (3.23), (3.24) and (3.25) respectively.

$$f_{x_0}(x) = \frac{1}{\sqrt{2\pi\sigma_{x_0}^2}} e^{-\frac{(x-\mu_{x_0})^2}{2\sigma_{x_0}^2}}, \quad (3.23)$$

$$f_w(w) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{w^2}{2\sigma_w^2}}, \quad (3.24)$$

$$f_v(v) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{v^2}{2\sigma_v^2}} \quad (3.25)$$

The program gives opportunity to the user to modify the algorithm parameters such as the gate size, the number of quantization values of $x(0)$ and $w(k)$ and the maximum number of states at each time step. These parameters are directly related with the algorithm performance. However, these values also determine the computational time. For example, smaller gate size value means higher precision, but it also means longer computation time. Moreover, increasing the number of quantization values of the initial state $x(0)$ or the distribution noise $w(k)$ improves the performance, but resulting slower program run. For high L values, the number of states at each time step augments dramatically which results more and more complex program. By discarding the quantization steps with lower metrics at each time step, the computational burden can be reduced. This can be achieved by limiting the maximum number of states and preserving only the most probable states which have the highest metric values.

3.8 An Example of the ODSA Algorithm

Let us consider a target whose motion from time zero to time 2 is described by Figure 20. Using the ODSA, we would like to find the path in the trellis diagram which was most likely followed by the target from time zero to time 2.

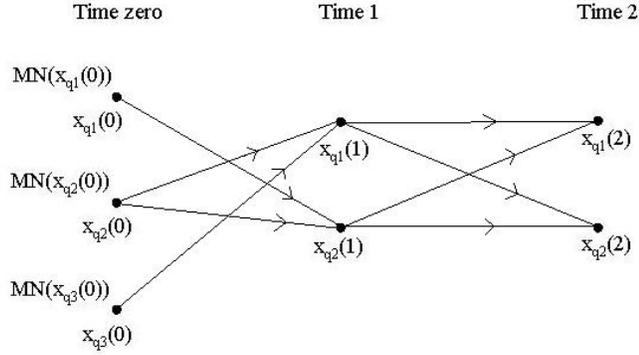


Figure 20 Trellis diagram for the target motion from time zero to time 2

Preliminary step: To each initial node, assign its metric, i.e., $MN(x_{qi}(0)) = Prob\{x_q(0) = x_{qi}(0)\}$, where $i=1, 2, 3$. From now on, the metric of the node $x_{qi}(k)$ is represented by $MN(x_{qi}(k))$.

Step 1: Consider the node $x_{q1}(1)$. The branches $x_{q2}(0)x_{q1}(1)$ and $x_{q3}(0)x_{q1}(1)$ are the only ones connecting the nodes at time zero to the node $x_{q1}(1)$. Hence calculate the metrics of these branches, then add these metrics to the metrics of the nodes $x_{q2}(0)$ and $x_{q3}(0)$ and obtain the following:

$$A_{11} \triangleq M(x_{q2}(0) \rightarrow x_{q1}(1)) + MN(x_{q2}(0)), \quad (3.26)$$

$$A_{12} \triangleq M(x_{q3}(0) \rightarrow x_{q1}(1)) + MN(x_{q3}(0)). \quad (3.27)$$

Further, assuming that $A_{11} = A_{12}$, the path $x_{q2}(0)x_{q1}(1)$ is chosen as the best path for the node $x_{q1}(1)$, and A_{11} is assigned to the node $x_{q1}(1)$ as its metric, i.e., $MN(x_{q1}(1)) = A_{11}$. The path $x_{q3}(0)x_{q1}(1)$ is then discarded. Let us now assume that the following are similarly found for the node $x_{q2}(1)$: $x_{q1}(0)x_{q2}(1)$ is the best path for $x_{q2}(1)$, and $MN(x_{q2}(1)) = M(x_{q1}(0) \rightarrow x_{q2}(1)) + MN(x_{q1}(0))$. Hence, we have Figure 21 at the end of step 1.

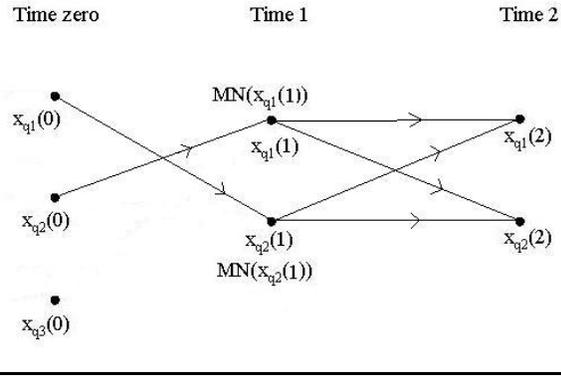


Figure 21 Trellis diagram from time zero to time 2 at the end of first step

Step 2: Consider the node $x_{q1}(2)$. The branches $x_{q1}(1)x_{q1}(2)$ and $x_{q2}(1)x_{q1}(2)$ are those connecting the nodes at time 1 to the node $x_{q1}(2)$. Hence, calculating the metrics of these branches and adding these metrics to the metrics of the nodes $x_{q1}(1)$ and $x_{q2}(1)$, we obtain the following:

$$A_{21} \triangleq M(x_{q1}(1) \rightarrow x_{q1}(2)) + MN(x_{q1}(1)), \quad (3.28)$$

$$A_{22} \triangleq M(x_{q2}(1) \rightarrow x_{q1}(2)) + MN(x_{q2}(1)). \quad (3.29)$$

Further, assuming that $A_{22} \geq A_{21}$, the path $x_{q1}(0)x_{q2}(1)x_{q1}(2)$ is chosen as the best path for the node $x_{q1}(2)$, and A_{22} is assigned to the node $x_{q1}(2)$ as its metric, i.e., $MN(x_{q1}(2)) = A_{22}$. The path $x_{q2}(0)x_{q1}(1)x_{q1}(2)$ is then discarded. Let us now assume that the following are similarly found for the node $x_{q2}(2)$: $x_{q2}(0)x_{q1}(1)x_{q2}(2)$ is the best path for $x_{q2}(2)$, and $MN(x_{q2}(2)) = M(x_{q1}(1) \rightarrow x_{q2}(2)) + MN(x_{q1}(1))$. Hence, we have Figure 22 at the end of step 2. In addition, assuming that $MN(x_{q2}(2)) \geq MN(x_{q1}(2))$, the path $x_{q2}(0)x_{q1}(1)x_{q2}(2)$ is chosen as the path followed by the target from time zero to time 2.

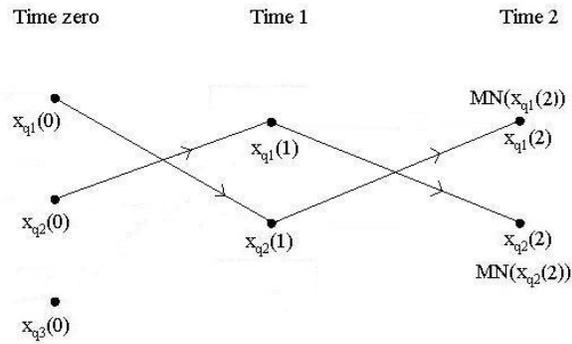


Figure 22 Trellis diagram from time zero to time 2 at the end of second step

CHAPTER 4

SIMULATION RESULTS FOR DIM TARGET TRACKING ALGORITHM

In order to check the performance of the algorithm, the actual dim targets are necessary with different parameters. The parameters are grouped in four parts in the list below with respect to their relations.

- a. Observation Parameters
- b. Target Parameters
- c. ODSA Parameters
- d. Risk Parameters

Observation parameters are related to image sensor and environment specifications. The list below gives observation parameters.

- i. Environment Noise Statistics
- ii. Image Size
- iii. Illumination

Target parameters are related to dim moving objects that is to be tracked by the image sensor. These parameters are unknown and can't be measured by the algorithm. The list below gives target parameters.

- i. Target Size and Shape
- ii. Target Motion Model
- iii. Target Motion Parameters

Risk parameters are related to missing of targets to be tracked. These parameters are application dependent and chosen by the user knowing what kind of targets to be tracked. The list below gives risk parameters.

- i. Superpixel Size
- ii. LRT Change Detection Threshold
- iii. Distribution Density Function
- iv. Minimum and Maximum Expected Target Areas

ODSA parameters are related to estimation of the path based on the observations that is given by the dim target tracking algorithm. These parameters are performance dependent and can be changed to get more accurate results. There exists a tradeoff between the time consumption of ODSA and average estimation error. The list below gives ODSA parameters.

- i. Gate Size
- ii. Quantization Number of the Initial Target Path Vector
- iii. Quantization Number of the Disturbance Noise

- iv. Initial Target Path Vector Variance
- v. Disturbance Noise Variance
- vi. Observation Noise Variance
- vii. Limiting the Maximum Possible Target Points Number

In order to check the performance of the algorithm, the actual observations and targets with desired parameters are needed. Therefore, a simulation algorithm is implemented to handle the parameters given at

Table 2 with the help of *MATLAB* function *randn(.)*.

Table 2 Simulation Algorithm Input Parameters

TARGET PARAMETERS	SENSOR AND ENVIRONMENT PARAMETERS
Motion Model	Image Size
Start and Last Point	Pixel Size
Velocity	Maximum Number of Frames
Acceleration	Environment Intensity Mean and Variance
Initial Direction	Illumination Function and Rate
Maneuver Ratio	
Intensity Mean and Variance	
Shape and Size	

The simulation results given below show the effects of the parameters of the algorithm described in Chapter 2 and 3. Simulations are performed for 250 program executions. For each execution, the related parameters are regenerated with the same values. Moreover, results acquired from different model parameters or SNR values are plotted on the same graph to increase the comprehension.

4.1 Effect of Observation Parameters

4.1.1 Effect of Environment Noise Statistics

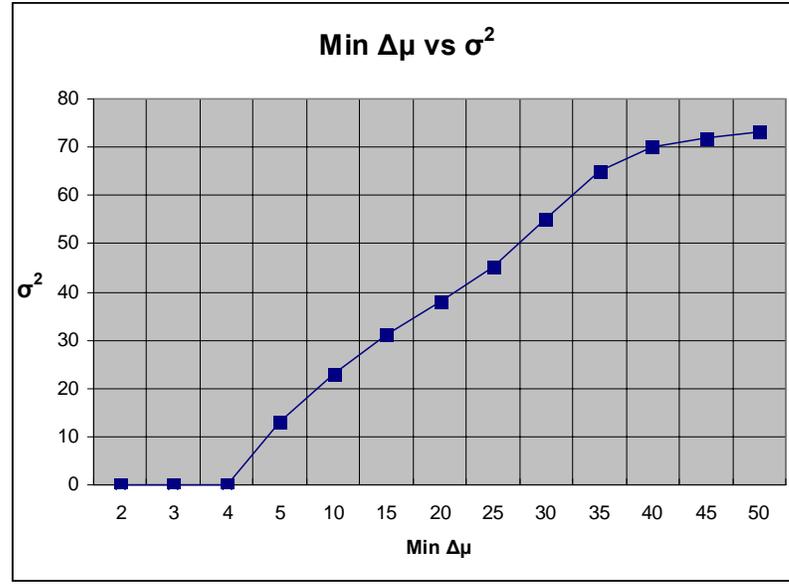


Figure 23 Plot of Minimum Tolerable Difference of Target and Environment Mean Intensity vs. Noise Variance

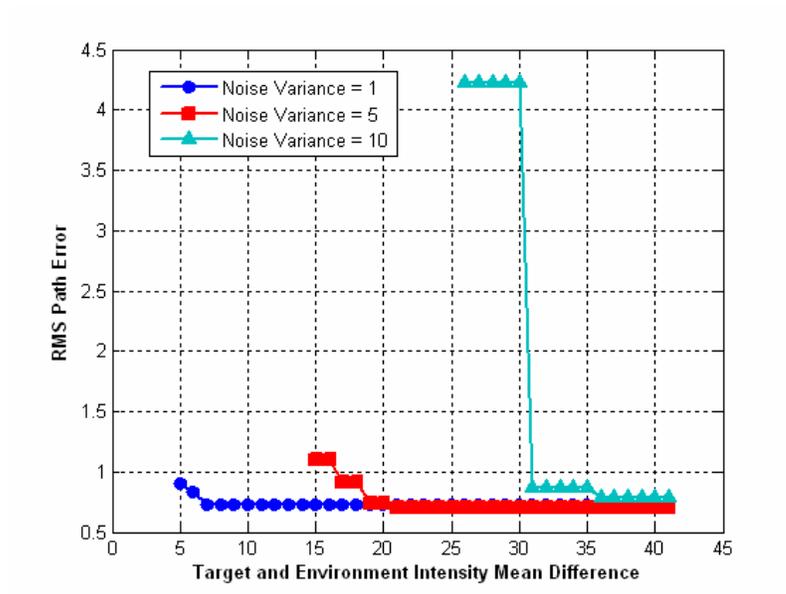


Figure 24 Plot of $\Delta\mu$ Intensity vs. RMS Target Path Error

Comment:

It can be seen from Figure 23 that minimum tolerable difference of target and environment mean intensity is linearly related with the noise variance. As the mean difference increases the noise variance also increases up to a saturation value which is approximately around the value “40”. After this point on, it is not important for the algorithm to increase the mean difference. It can handle the target detection and tracking independent of noise variance. Furthermore, the minimum mean difference that can be handled is “2” with no noise variance. If the environment noise takes into account, the minimum noise difference is “5”. The linear interpolation of the graphic in the non saturated region results in the equation below:

$$\Delta\mu_{\min} \approx 2\sigma^2 \quad (4.1)$$

From Figure 24, it is observed that estimation error of the algorithm significantly depends on the noise variance. The tolerance of the mean difference between target and environment degrades as the noise variance increase. It is also clear that after a satisfactory mean difference, the RMS error saturates around “0.7”.

Furthermore, if the mean difference is under a tolerable value with respect to the noise variance, it is impossible to track the target.

4.1.2 Effect of Image Size

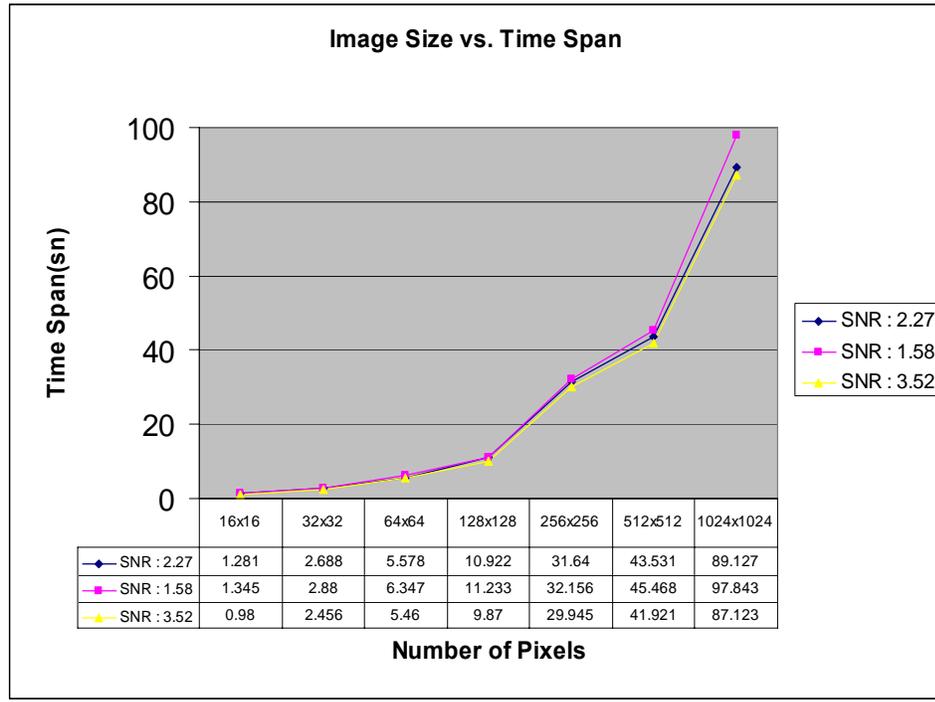


Figure 25 Plot of Image Size vs. Time Consumption in 50 frames for different SNR values

Comment:

It can be seen from Figure 25 that image size of the observation frames significantly effects the time to complete the analysis. The time span formula in seconds with respect to the image size when a curve fitting algorithm applied for an exponential approximation is given below:

$$T_s = 1.03.e^{19.82X} \quad (4.2)$$

where X is the number of pixels along one dimension of the frame and T_s is the time spanned for the analysis.

From Figure 25, it is also observed that The SNR value of the target for a fixed image size effects slightly the time spanned by the algorithm. The reason of change is due to the increase in possible target candidates for low SNR values. Therefore, the algorithm is more useful for non real time applications such as satellite image analysis for space researches.

4.1.4 Effect of Illumination

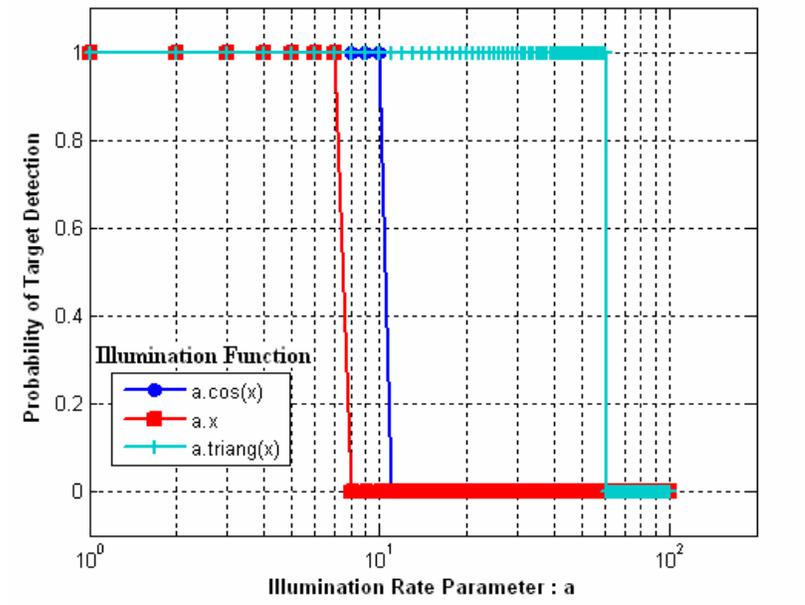


Figure 26 Plot of Probability of Detection for different illumination functions

Comment:

It can be seen from Figure 26 that if there exists an illumination source on the path of the target, the probability of target detection degrades significantly after some illumination rate value. The rate of illumination is defined as the pixel intensity change in two successive frames. The parameter X is the frame number in Figure 26.

The reason of the performance reduction is due to the illumination clouds that hide the real target in LRT Difference Matrix. In average, after a pixel intensity change of approximately 10, it is impossible to identify the target in the illumination cloud. It is also observed that probability of target detection can differ according to the illumination function. *triang(.)* is the *Matlab* function to create triangular waveforms.

4.2 Effect of Target Parameters

4.2.1 Effect of Target Size and Shape

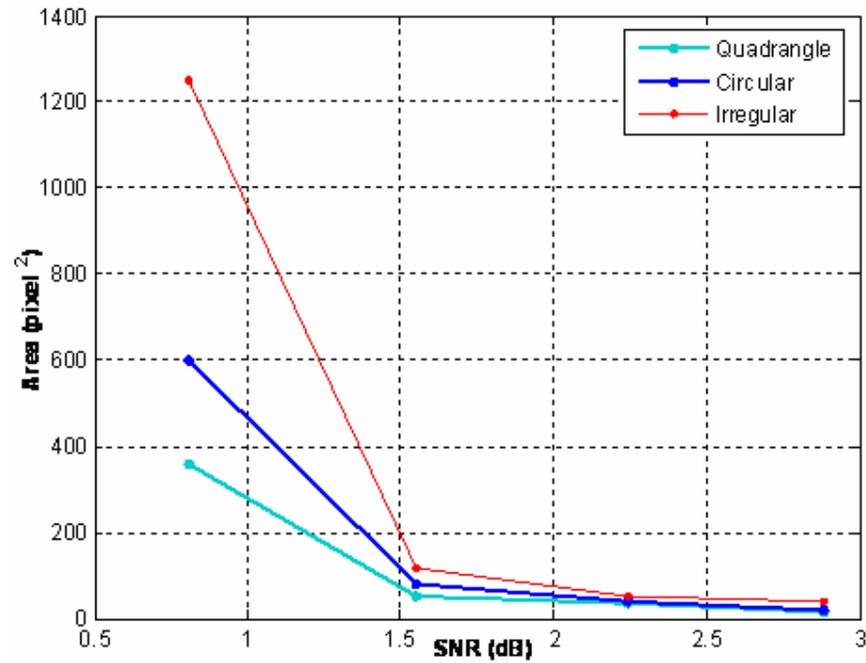


Figure 27 Plot of Target Area for different shapes vs. Minimum Tolerable Target SNR

Table 3 Minimum Tolerable SNR for detection with different sizes and shapes

	SNR	Dimensions	Area	Diagonal
Quadrangle	0.81	[19 19]	360	26.87
	1.55	[7 7]	50	9.89
	2.24	[6 6]	35	8.48
	2.88	[4 4]	15	5.65
Circular	0.81	radius = 14	600	28
	1.55	radius = 5	80	10
	2.24	radius = 3.5	40	7
	2.88	radius = 2,5	20	5
Irregular	0.81	Extent ratio = 0.4	1250	70
	1.55	Extent ratio = 0.4	115	15
	2.24	Extent ratio = 0.4	50	10
	2.88	Extent ratio = 0.4	40	8

The meaning of parameters used in Table 3 is given below.

1. Diagonal: Maximum distance along the peripheral coordinates of the target
- 2.

$$\text{Extent Ratio} = \frac{\text{Area of the target}}{\text{Area of the bounding box}} \quad (4.3)$$

where bounding box is the minimum rectangular area that encapsulates the target.

Comment:

It can be seen from Figure 27 that minimum tolerable target SNR for the dim target tracking algorithm is indirectly proportional with target area. It is also clear that quadrangle targets result in the best performance of detection. As the complexity of the shape of the target increases, it is hard to detect in low SNR situations. The reason for this comes from the superpixel size and shape. At the target edges, the probability of covering the target in the related superpixel gets so small that it becomes almost impossible to extract the maximum distribution of the related target piece in the second frame. However, it is also important to notify that minimum tolerable target area for all SNR values is constant, approximately 25 pixel².

4.2.2 Effect of Target Motion Model

The target motion models studied are listed below:

- I. LCV : Linear Constant Velocity
- II. LCA : Linear Constant Acceleration
- III. CMCV : Constant Maneuver Constant Velocity
- IV. CMCA : Constant Maneuver Constant Acceleration
- V. VMCV : Varying Maneuver Constant Velocity
- VI. VMVV : Varying Maneuver Varying Velocity

Table 4 Target Motion Models and Estimated RMS Errors

MOTION MODEL	TARGET PARAMETERS		RMS ERROR
LCV	Target Intensity mean	125	1.631
	Target size	[8 8]	
	Target velocity	25	
	Target maneuver Index	NA	
	Target acceleration	NA	
	Image Size	[1024 1024]	
	Number of Frame	30	
	Noise Intensity mean	100	
	Noise Intensity variance	15	
LCA	Target Intensity mean	125	1.759
	Target size	[8 8]	
	Target velocity	25	
	Target maneuver Index	NA	
	Target acceleration	2	
	Image Size	[1024 1024]	
	Number of Frame	30	
	Noise Intensity mean	100	
	Noise Intensity variance	15	
CMCA	Target Intensity mean	125	2.234
	Target size	[8 8]	
	Target velocity	25	
	Target maneuver Index	0.05	
	Target acceleration	2	
	Image Size	[1024 1024]	
	Number of Frame	30	
	Noise Intensity mean	100	
	Noise Intensity variance	15	

MOTION MODEL	TARGET PARAMETERS		RMS ERROR
CMCV	Target Intensity mean	125	1.815
	Target size	[8 8]	
	Target maneuver Index	0.05	
	Target acceleration	NA	
	Target size	[8 8]	
	Target velocity	25	
	Image Size	[1024 1024]	
	Number of Frame	30	
	Noise Intensity mean	100	
	Noise Intensity variance	15	
VMCV	Target Intensity mean	125	2.261
	Target size	[8 8]	
	Target velocity	25	
	Target maneuver Index	0.05+randn (.)	
	Target acceleration	NA	
	Image Size	[1024 1024]	
	Number of Frame	30	
	Noise Intensity mean	100	
	Noise Intensity variance	15	
MOTION MODEL	TARGET PARAMETERS		RMS ERROR
VMVV	Target Intensity mean	125	2.349
	Target size	[8 8]	
	Target velocity	25	
	Target maneuver Index	0.05+randn (.)	
	Target acceleration	2+randn (.)	
	Image Size	[1024 1024]	
	Number of Frame	30	
	Noise Intensity mean	100	
	Noise Intensity variance	15	

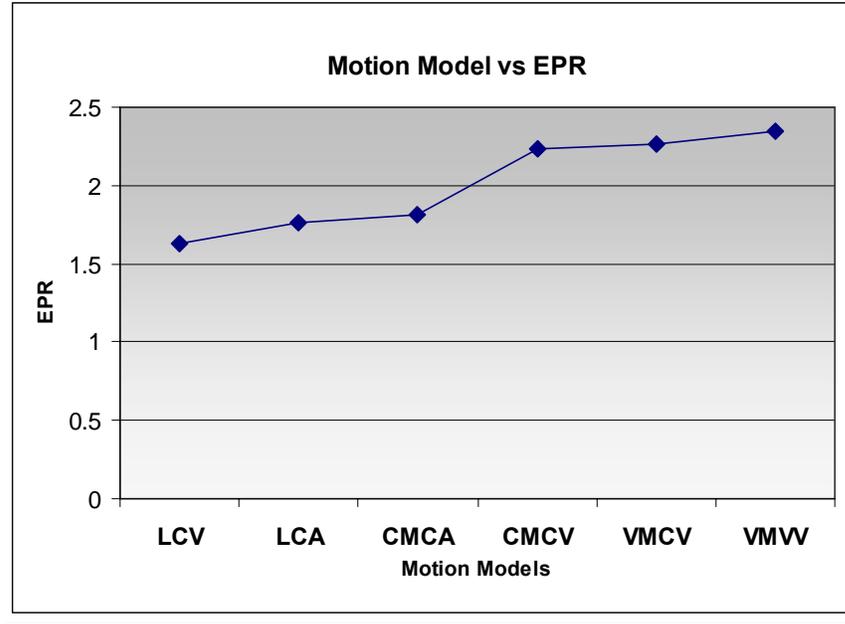


Figure 28 Plot of Estimated Path RMS Error for different Target Motion Models
 EPR : Estimated Path RMS Error

Comment:

It can be seen from Table 4 and Figure 28 that as the complexity of target motion increases, the error that results from the algorithm also increases. The function used for variable acceleration and maneuver is the *Matlab* function “*randn(.)*” which generates random numbers with zero mean and unit variance.

The effect of maneuver index is given by the formula below:

$$\Delta\Phi = \frac{\beta_m \cdot 180}{\pi} \quad (4.4)$$

where “ $\Delta\Phi$ ” is the angle change in degrees between two consecutive frames and “ β_m ” is the maneuver index written at

In Table 4 the velocity and acceleration parameters are in terms of pixel distance. “*NA*” is used for “*Not Applicable*”.

4.2.3 Effect of Target Motion Parameters

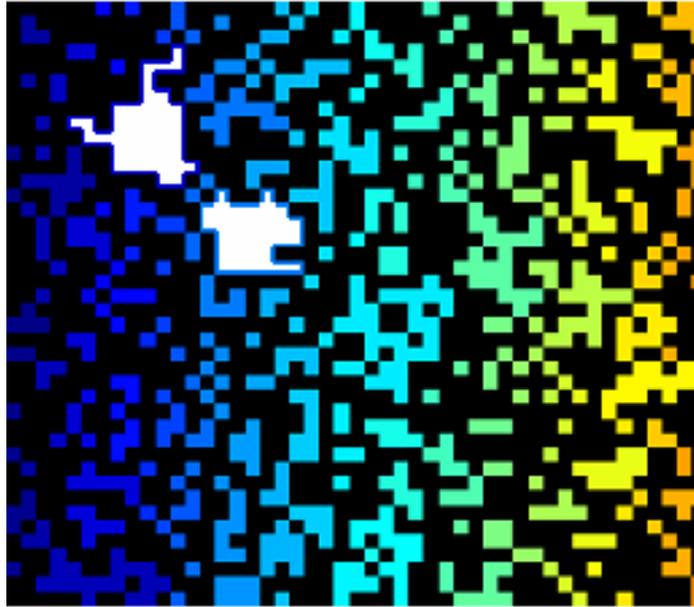


Figure 29 Plot of Target Label (white) in the Difference Matrix Target Velocity: 20 Pixels / Frame
SNR: 2.24 and Size: [8 8]

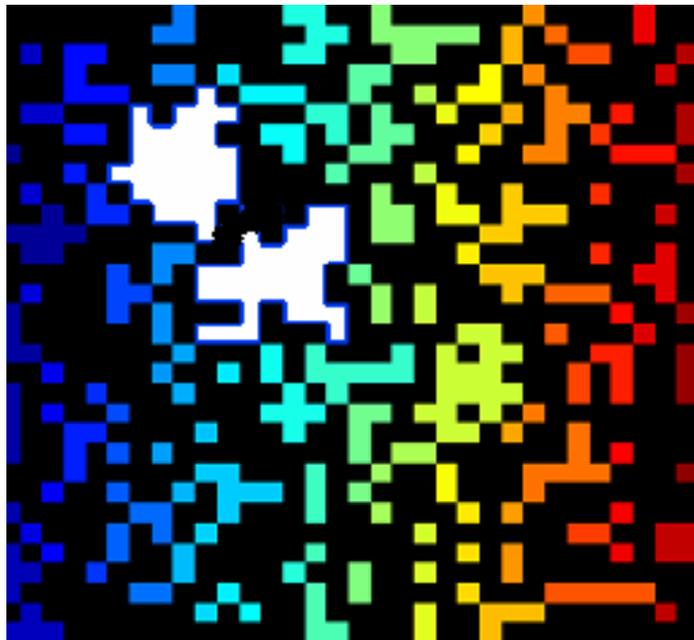


Figure 30 Plot of Target Label (white) in the Difference Matrix Target Velocity: 15 Pixels / Frame
SNR: 2.24 and Size: [8 8]



Figure 31 Plot of Target Label (white) in the Difference Matrix Target Velocity: 10 Pixels / Frame
SNR: 2.24 and Size: [8 8]



Figure 32 Plot of Target Label (white) in the Difference Matrix Target Velocity: 7 Pixels / Frame
SNR: 2.24 and Size: [8 8]

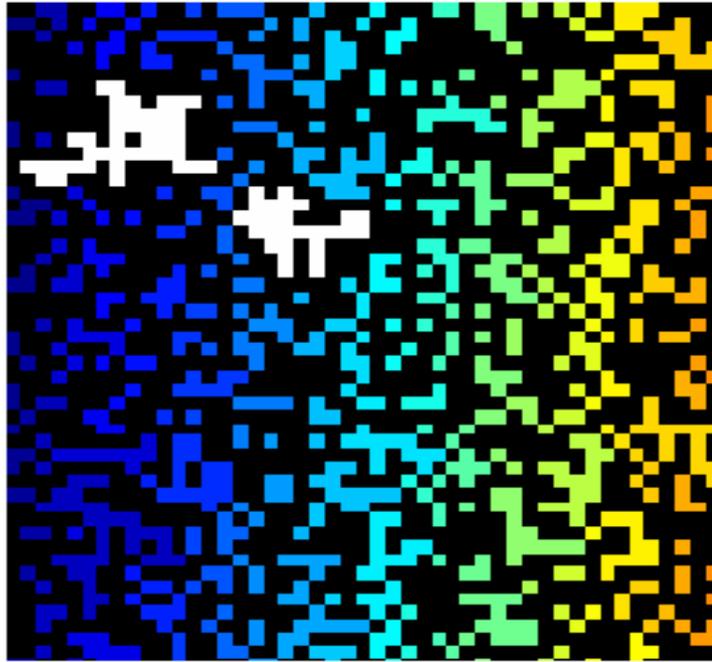


Figure 33 Plot of Target Label (white) in the Difference Matrix Target Velocity: 20 Pixels / Frame
SNR: 1.55 and Size: [8 8]

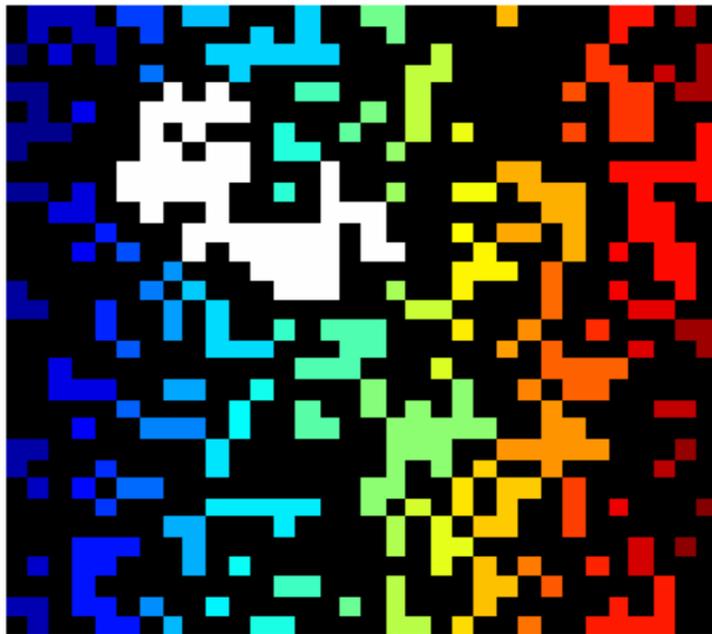


Figure 34 Plot of Target Label (white) in the Difference Matrix Target Velocity: 15 Pixels / Frame
SNR: 1.55 and Size: [8 8]

Comment:

It can be seen from Figure 29-Figure 34 that as the SNR is decreased for low velocity targets, old and new target positions gets closer to each other in the label matrix and finally become a single target label. Therefore for each target SNR value, there exists a breakpoint that the algorithm can separate the old and new target positions from the image frames. As the SNR value is decreased, it can be also seen from Figure 33 and Figure 34 that the target velocity must increase above a critical value to differentiate the target labels.

4.3 Effect of Risk Parameters

4.3.1 Effect of Superpixel Size

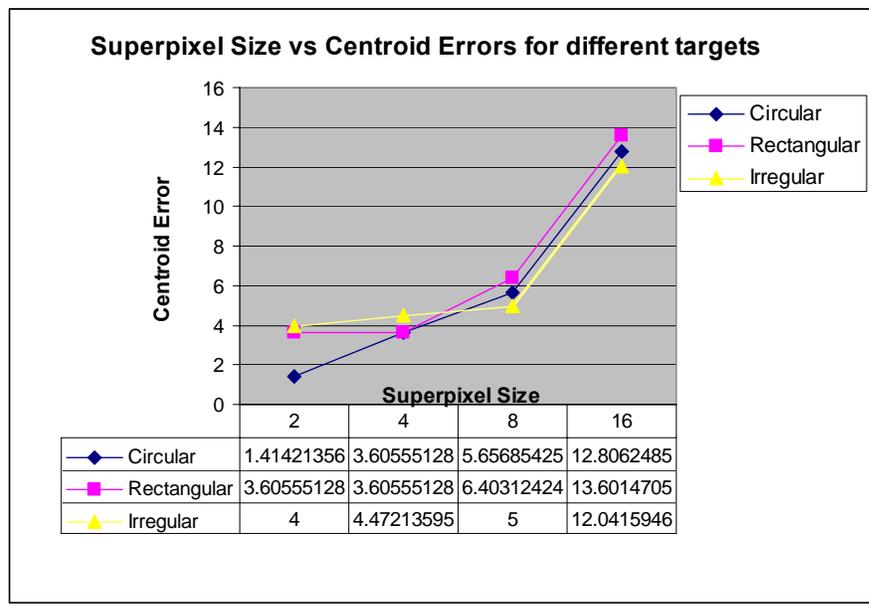


Figure 35 Plot of Superpixel Size vs. Centroid Errors for different targets

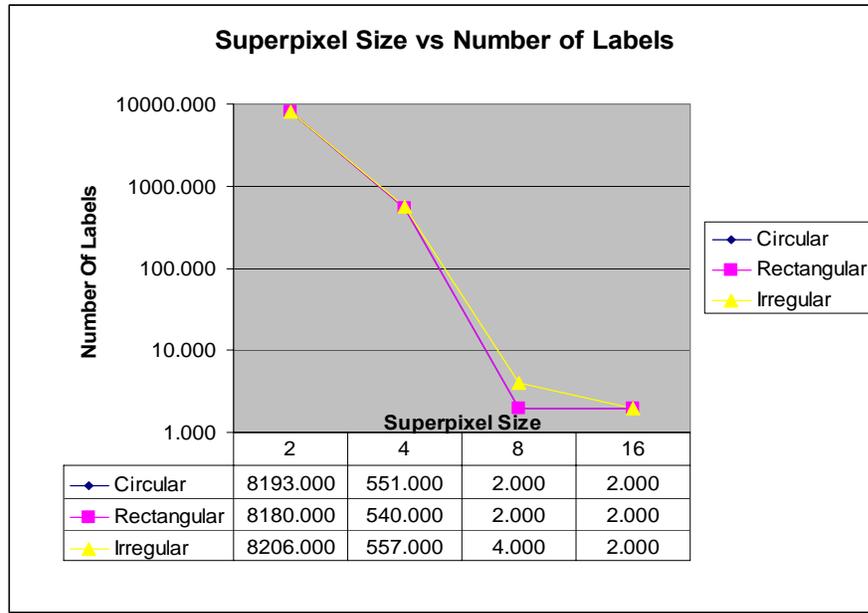


Figure 36 Plot of Superpixel Size vs. Number of Labels for different targets

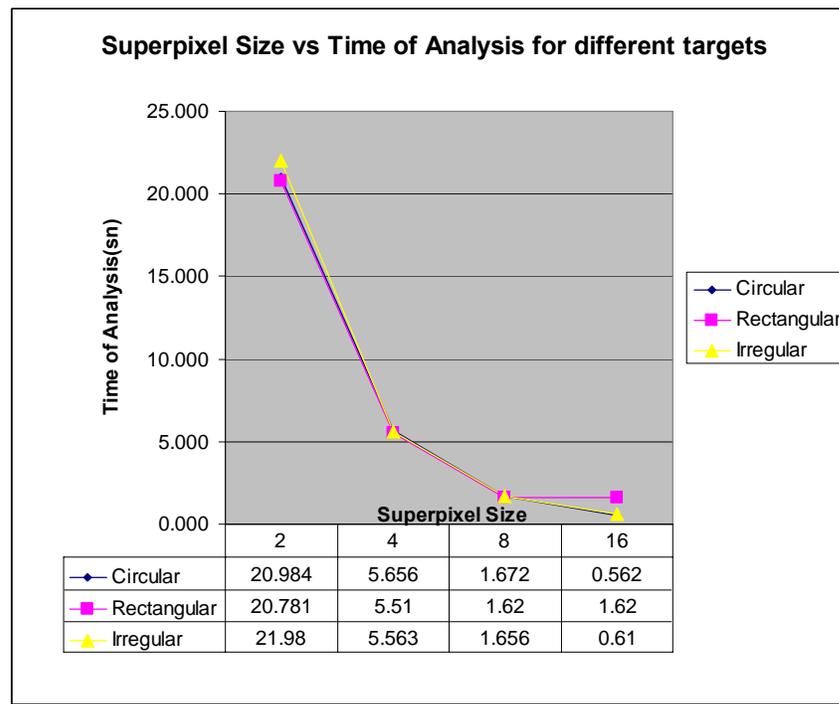


Figure 37 Plot of Superpixel Size vs. Time of Analysis for different targets

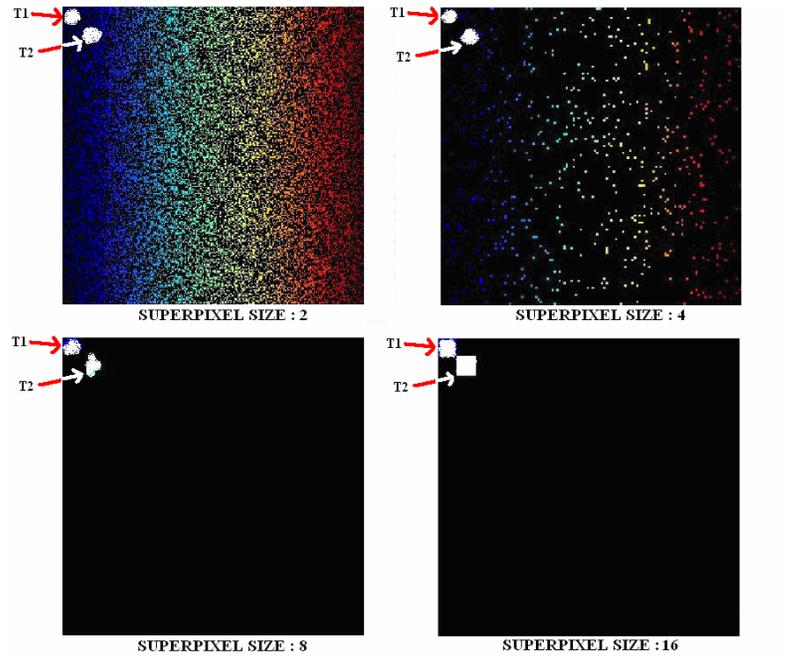


Figure 38 Effect of Superpixel Size to Number of Centroids
T1: Target Position at time k, T2: Target Position at time k+1

Comment:

It can be seen from Figure 35 that as the superpixel size used for motion detection algorithm increases, the centroid error which is defined as difference between the label geometric centroid and real target centroid increases. Therefore, the probability of missing the target in the next iteration step also increases rapidly. However, as it can be seen from Figure 36 and Figure 37 that the increase of superpixel size decreases number of candidate labels that the target can move to. Furthermore, the time spanned by the algorithm used for motion detection analysis also decreases sharply. In Figure 38, the difference matrix caused by LRT detection in two successive frames is given for different superpixel size values. According to these results, it is clear that the superpixel size has to be optimized with respect to the tradeoff between centroid error and number of possible target label candidates. Too small superpixel size causes more time consumption while results in more accurate centroid estimations.

4.3.2 Effect of LRT Change Detection Threshold

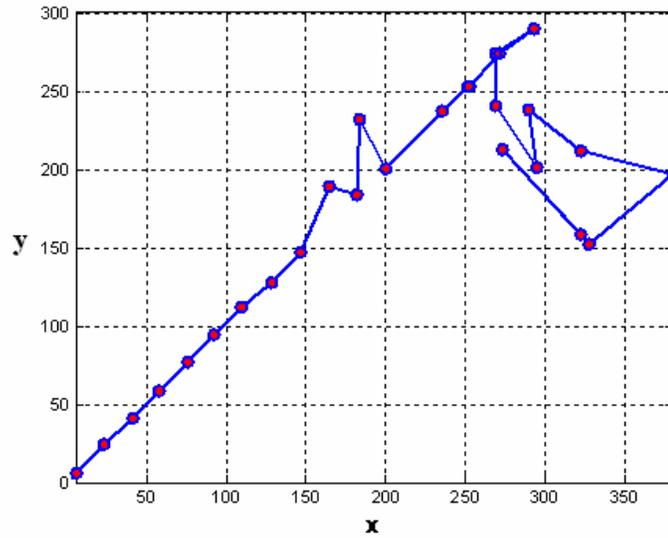


Figure 39 Plot of Estimated Target Path with Motion Model: LCV, Start:[0,0] Finish:[500, 500]
Change Detection Threshold: 15 and SNR: 1.9 dB

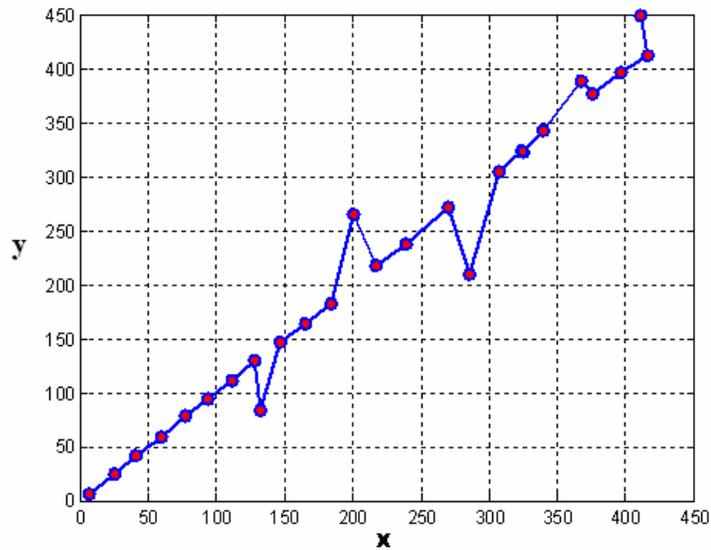


Figure 40 Plot of Estimated Target Path with Motion Model: LCV, Start: [0, 0] Finish: [500, 500]
Change Detection Threshold: 30 and SNR: 1.9 dB

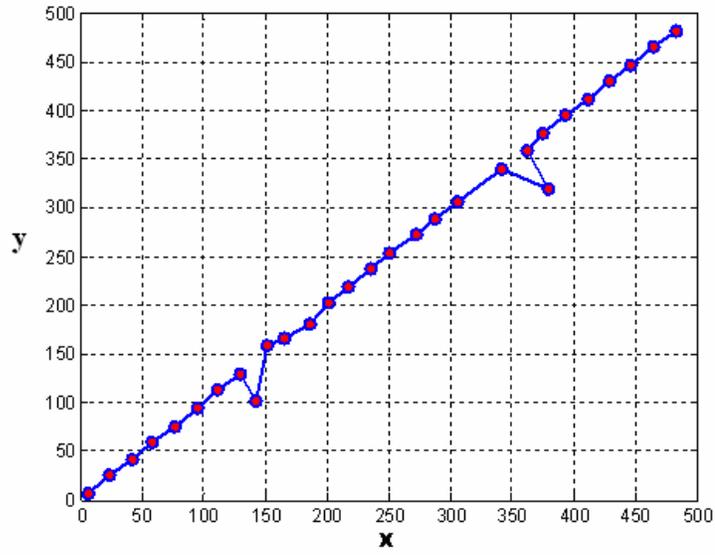


Figure 41 Plot of Estimated Target Path with Target Motion Model: LCV, SP: [0, 0] LP: [500 500]
Change Detection Threshold: 40 and SNR: 1.9 dB

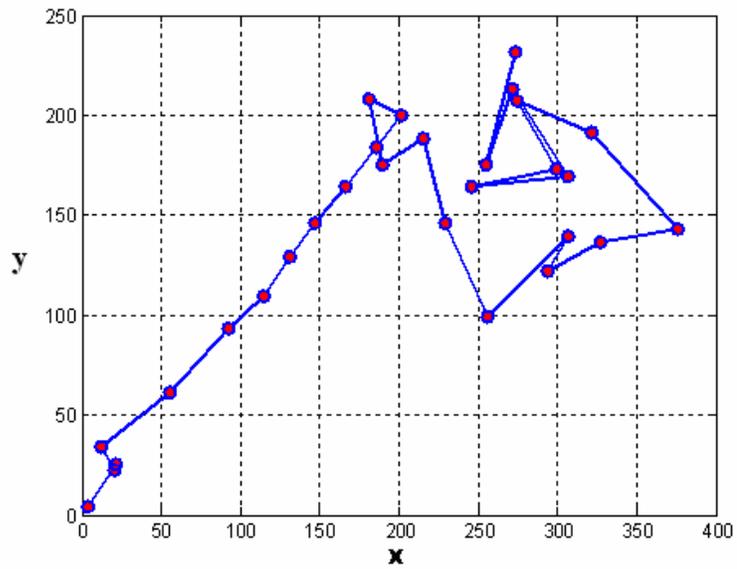


Figure 42 Plot of Estimated Target Path with Target Motion Model: LCV SP: [0, 0] LP: [500 500]
Change Detection Threshold: 70 and SNR: 1.2 dB

Comment:

It can be observed from Figure 39 - Figure 42 that there exists an inverse proportion between SNR and LRT threshold. The false alarms of target centroids are decreased considerably by increasing the LRT threshold. However, it is also necessary to mention that increasing the threshold above a value which eliminates also the target, causes overall miss of the target. Therefore an optimum value for the threshold exists. The optimization criterion is between ODSA and detection. As the false alarms get effective with decreased threshold for low SNR targets, the ODSA can smooth the target path up to a tolerable value. In Figure 42, ODSA can not smooth the path since the target path has already been deflected too much from the real one.

4.3.3 Effect of Distribution Density Function

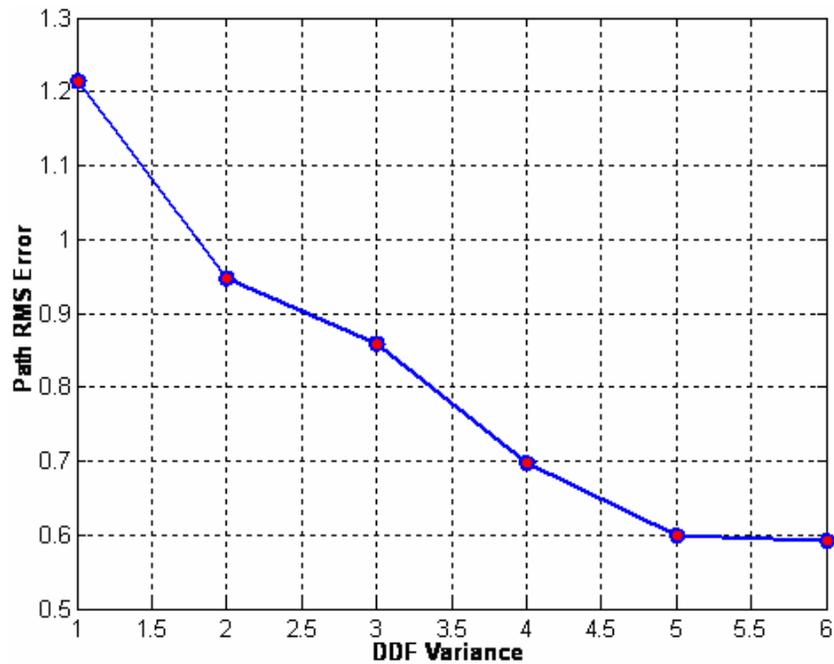


Figure 43 Plot of Estimated Target Path RMS vs. Distribution Density Function Variance

Table 5 Different Types of DDF with their Estimated Path RMS Errors

Distribution Density Function	Threshold	SNR	RMS Error
$\exp(-x)/(0.2)$	10	2.24	1.215
$\exp(-x)/(0.1)$	10	2.24	0.859
$\exp(-x)/(0.02)$	10	2.24	0.600
$\text{Max}(-10x+50,0)$	10	2.24	0.948
$\text{Max}(-20x+50,0)$	10	2.24	0.698
$\text{Max}(-50x+50,0)$	10	2.24	0.592
$45(1-\text{Stepfun}(x,10))$	10	2.24	1.356
$45(1-\text{Stepfun}(x,5))$	10	2.24	1.093
$45(1-\text{Stepfun}(x,1))$	10	2.24	0.932

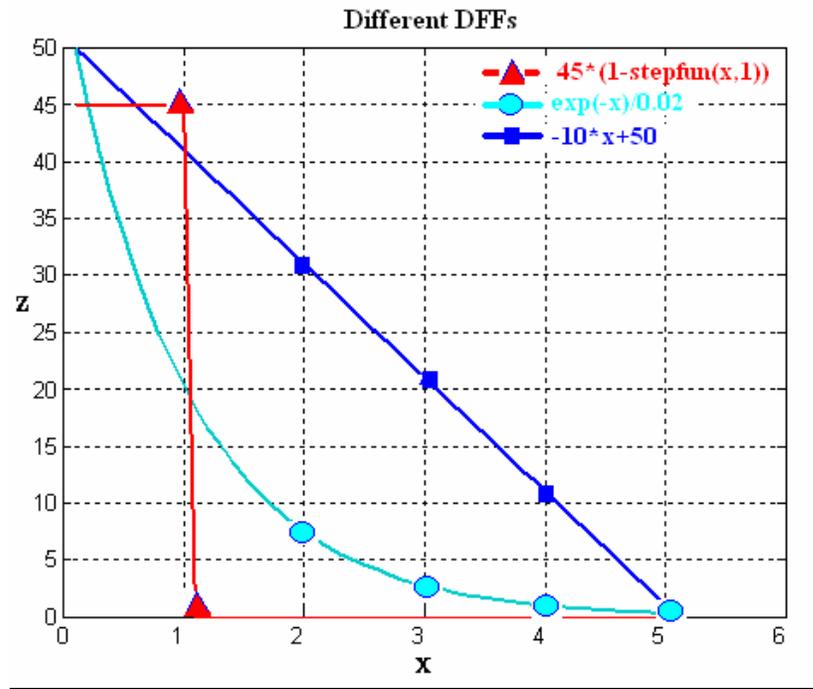


Figure 44 Plot of Different Distribution Density Functions along X axis

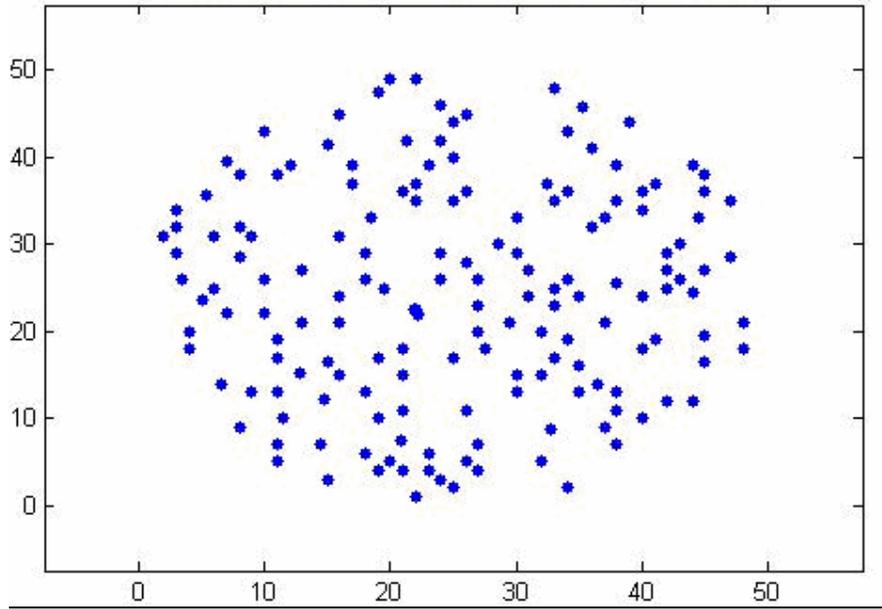


Figure 45 Plot of scattered mapping centroids in the focus window

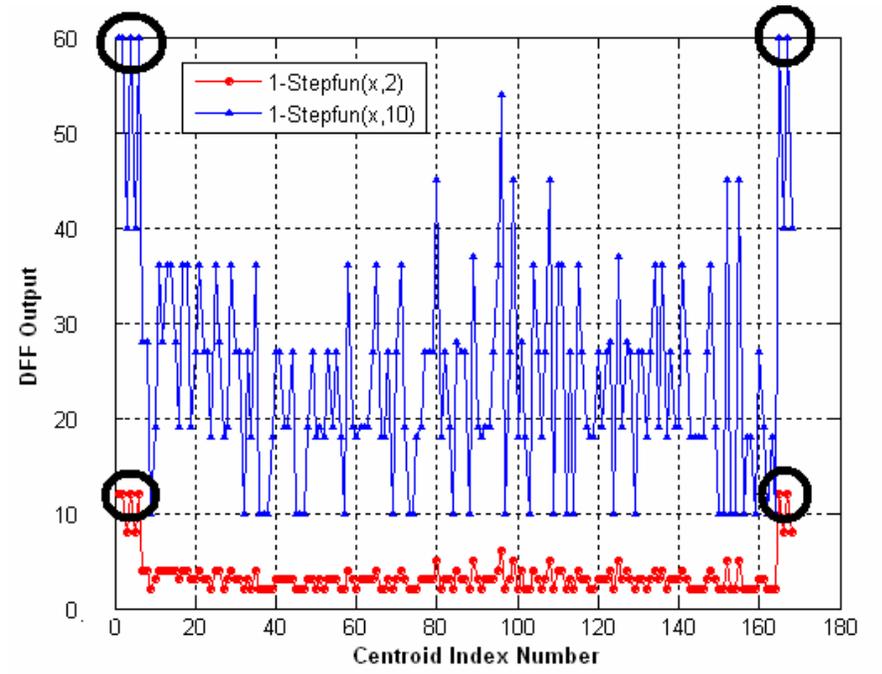


Figure 46 Plot of Step DDF outputs for Figure 45 with different bandwidths

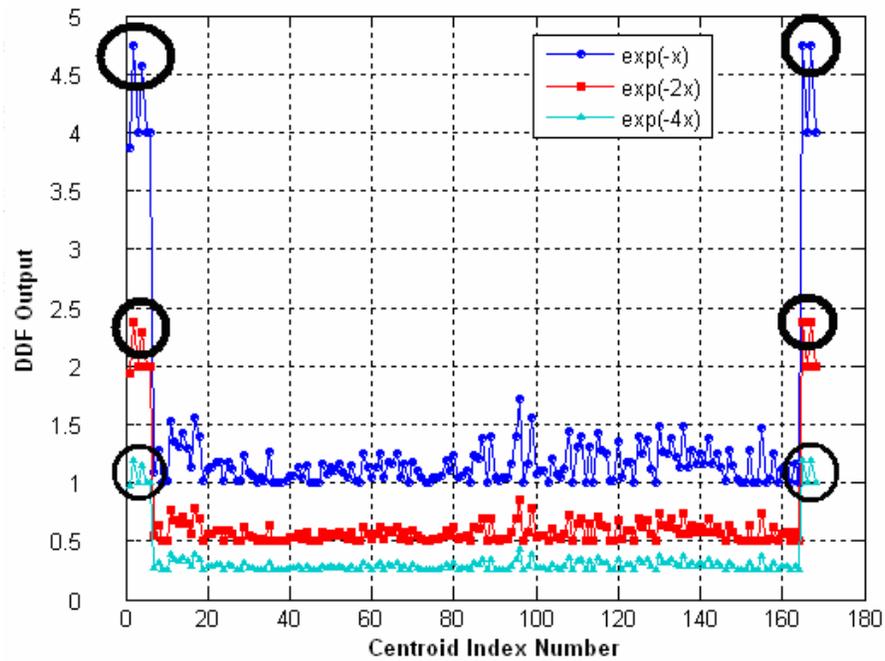


Figure 47 Plot of Exponential DDF outputs for Figure 45 with different bandwidths

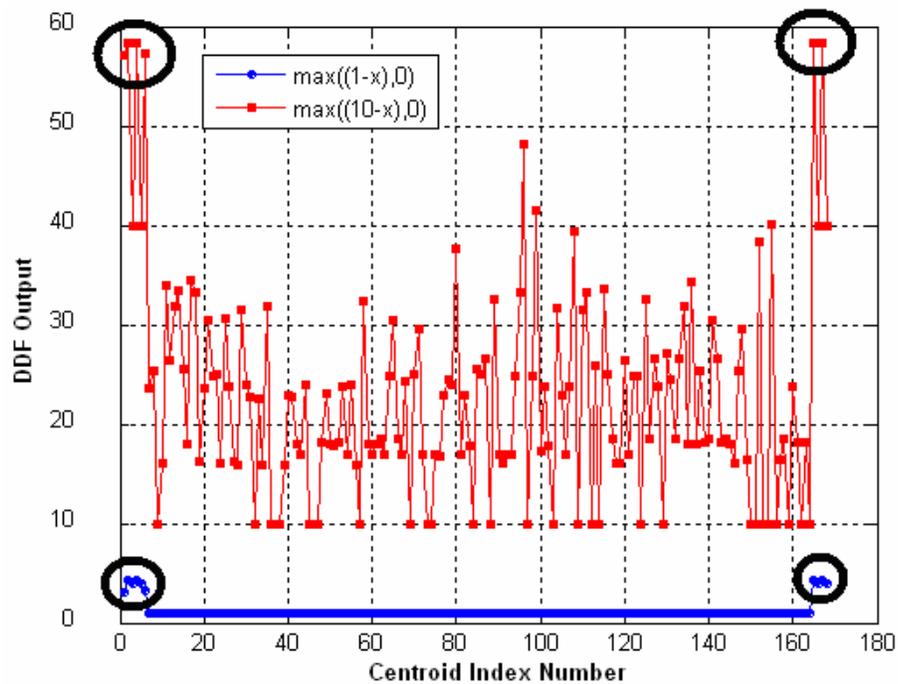


Figure 48 Plot of Exponential DDF outputs for Figure 45 with different bandwidths

Comment:

In Figure 43, it can be seen that as the energy of distribution function spreads around its central point estimated target path RMS error gets larger. Table 5 also shows that different types of Distribution Density Functions result in different performances in the algorithm. Some of the Distribution Density Functions used at Table 5 are given in Figure 44. The best performance along these three functions is given by the exponential function since the standard deviation of the exponential function is the minimum. The worst performance comes from the step function.

The second simulation result is related with Figure 45. When the scattered centroids that result from LRT matching of the subgroup elements is filtered with different kinds of DDF, the result of the maximization gives the same target centroid, which is shown by the black circles. The bandwidth of the DDF is very important to prevent false alarms.

4.3.4 Effect of Minimum and Maximum Expected Target Areas

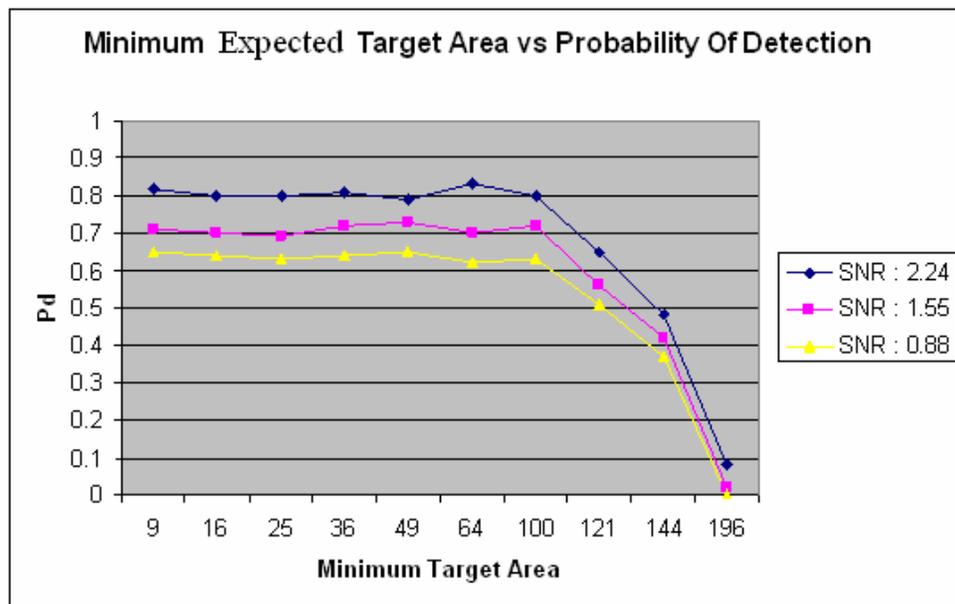


Figure 49 Plot of Minimum Expected Target Area vs. Probability of Detection
Target Area: 100 pixel²

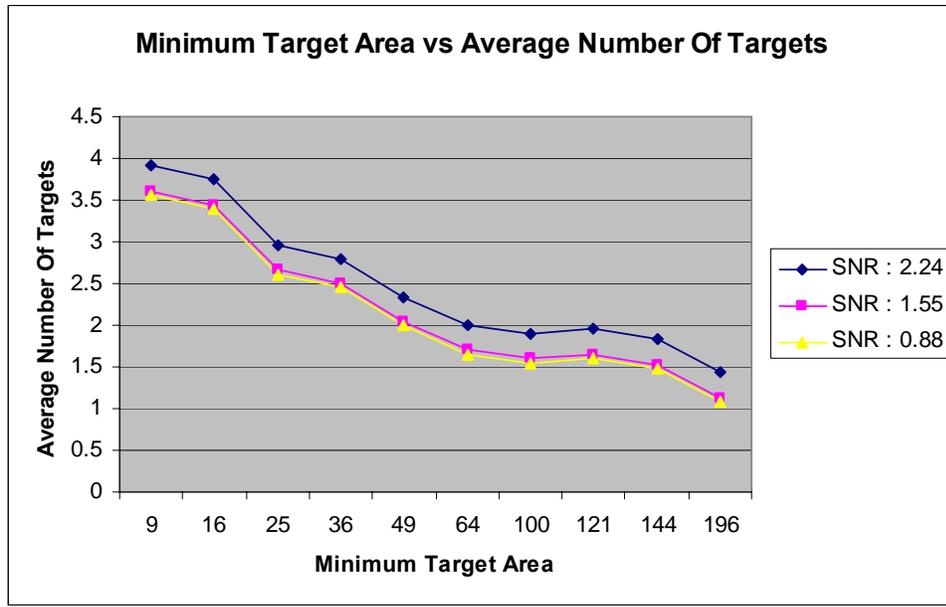


Figure 50 Plot of Minimum Target Area vs. Average Number of Targets

Comment:

In Figure 49, it can be seen that the probability of detecting the target for a fixed target SNR is independent of the minimum expected area up to the real target area. However, the number of possible targets decreases as the minimum expected target area gets closer to the real target area, which is given in Figure 50. The less the target candidates, the better performance for the algorithm is. The algorithm eliminates the targets that are smaller than the minimum expected target area. Therefore; according to the applications, the minimum expected target area should be less than the real target area to be in the safety region. It is also important to notice that as the SNR of the target degrades, the probability of detection for a fixed minimum expected target area reduces.

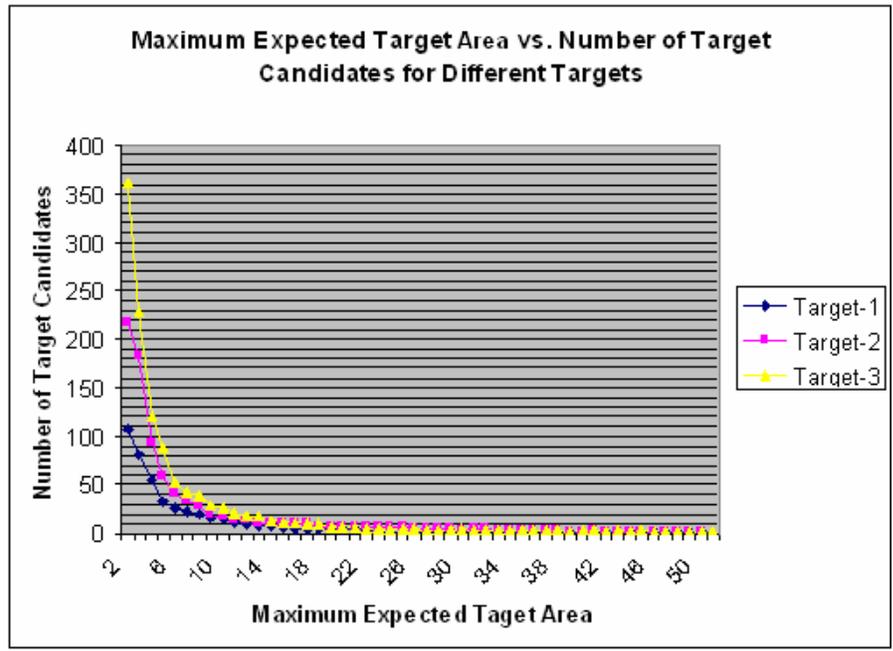


Figure 51 Plot of Maximum Expected Target Area vs. Number of Target Candidates
 Target 1: Circle, Target 2: Airplane, Target 3: Battle Tank



Figure 52 Plot of Maximum Expected Target Size vs. Minimum Identification Distance

Comment:

In Figure 51, it can be seen that the number of target candidates by using Maximum Histogram Size Filtering reduces significantly as the maximum expected target area decreases since this value is used as the size threshold for filtering in the algorithm. To get better performance from the algorithm without missing the target, it is necessary to decide an optimum maximum expected target area value. Notice that as the shape complexity of the target increases, the number of target candidates seen by the algorithm also increases approximately up to the real target size. However, from this point on, no matter what the target is, the algorithm can detect the target.

The target identification resolution in a group of targets strongly depends on maximum expected target area as it is observed from Figure 52. To get better target identification, it is necessary to reduce the maximum expected target area, although the time spanned for the analysis gets larger. If size threshold is less than the real target size, the target is broken into small sub targets by this filtering.

4.4 Effect of ODSA Parameters

The observation and state transition models used for the simulations are given by the equations below:

$$z_k = x_k + v_k \quad k = 1, 2, \dots, L \quad (4.5)$$

$$x_{k+1} = x_k + w_k \quad k = 0, 1, \dots, L-1 \quad (4.6)$$

where z_k is the observed target centroid, x_k is the real target centroid, v_k is the observation noise and w_k is the disturbance noise. The initial position x_0 is taken as the initial observation. No other affects such as extra input or interference are used in the models.

The parameter used for ODSA performance is the average estimation error which is given by the equation below:

$$AEE = \frac{\sqrt{\sum_i |C_i - \tilde{C}_i|^2}}{L} \quad i = 1, 2, \dots, L \quad (4.7)$$

where AEE is the average estimation error, C_i is the real target centroids along the path, \tilde{C}_i is the estimated target centroids and L is the total number of frames.

4.4.1 Effect of the Gate Size

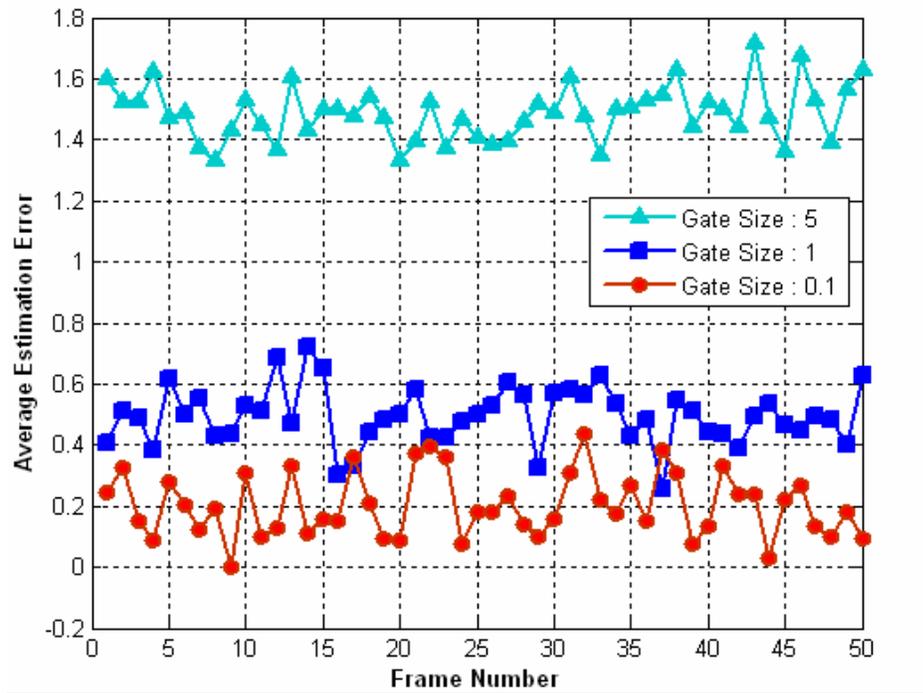


Figure 53 Plot of Frame Number vs. Average Estimation Error for Different Gate Sizes

Comment:

It can be observed from Figure 53 that the gate size is directly proportional with the algorithm performance. The target path estimation error reduces as the gate size becomes smaller. The gate size defined for image processing is related with the pixel size. Therefore, it is clear that as the pixel size of the image sensor gets smaller, the average estimation error reduces in the same manner. Gate size must be smaller than the superpixel size in order to smooth target centroid variation by ODSA.

4.4.2 Effect of the Quantization Number of the Initial Target Path Vector

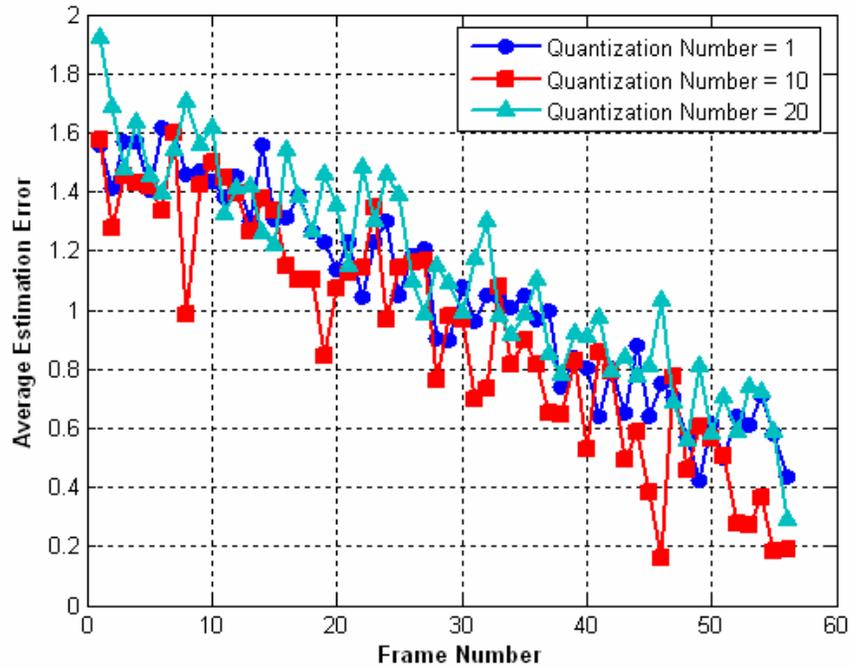


Figure 54 Plot of Frame Number vs. Estimation Error with respect to Initial Quantization Number

Comment:

It can be observed from Figure 54 that the number of the quantization levels of the initial target path vector slightly affects the performance of the algorithm. The mean value of the initial state should be chosen around the initial target movement coming from the dynamic vision algorithm which provides the observation values for ODSA.

4.4.3 Effect of the Quantization Number of the Disturbance Noise

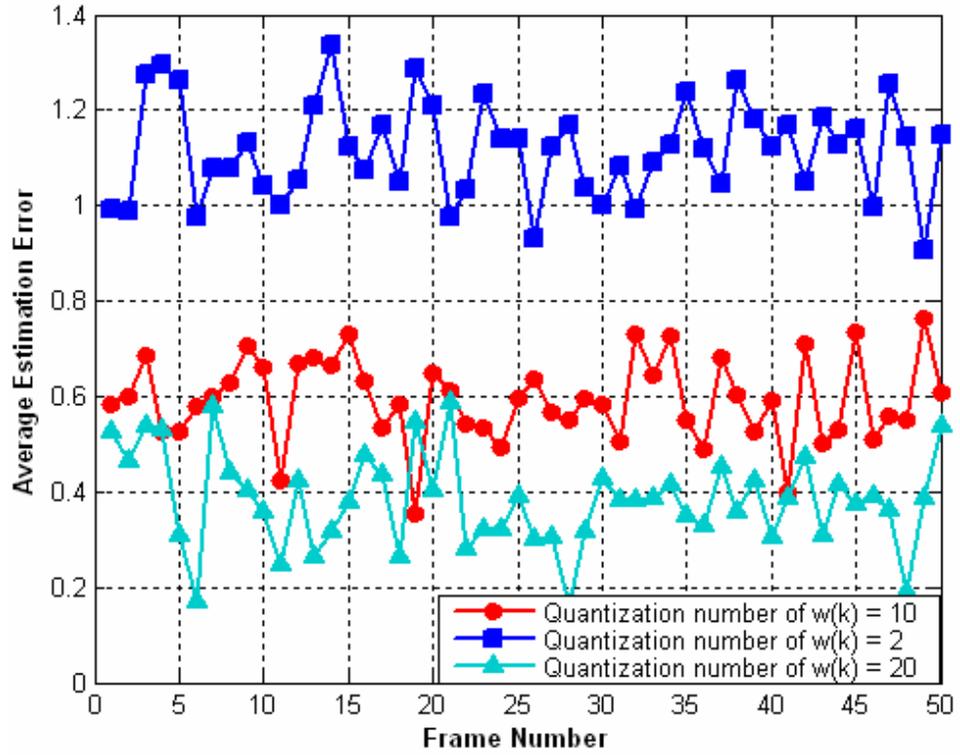


Figure 55 Plot of Frame Number vs. Estimation Error with respect to Disturbance Noise Quantization Number

Comment:

It can be observed from Figure 55 that increasing the number of the quantization levels of the disturbance noise improves the performance of the algorithm significantly. However, time consumption of the algorithm increases in the same manner. Therefore, as it is discussed in 4.4.7, limiting the maximum number of possible target points is necessary.

4.4.4 Effect of the Initial Target Path Vector Variance

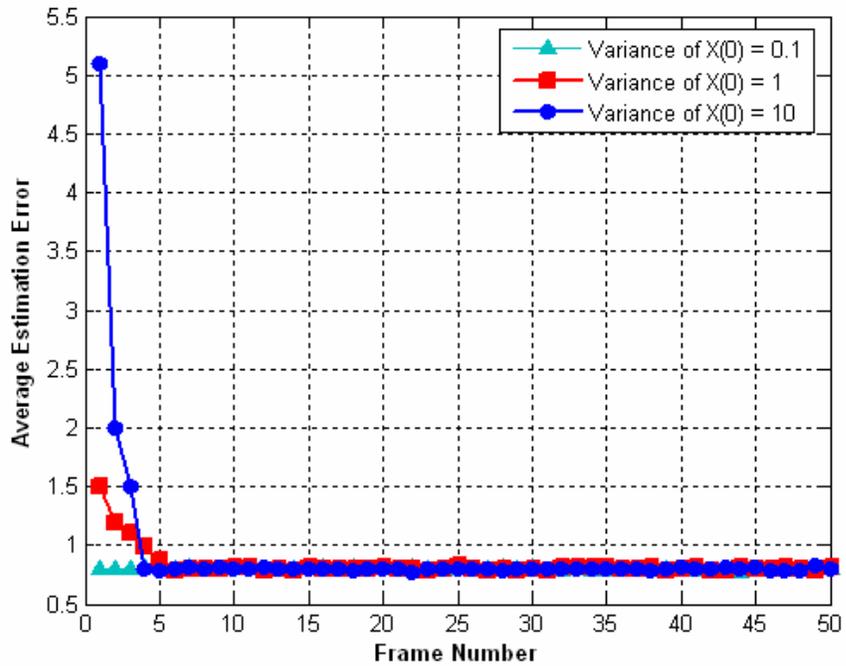


Figure 56 Plot of Frame Number vs. Estimation Error with respect to Initial Target Path Vector Variance

Comment:

It can be observed from Figure 56 that the variance of the initial target path vector only affects the initial time. As expected, the initial target path error increases with increasing the initial target path vector variance. For higher frame numbers, the average estimation error is not affected.

4.4.5 Effect of the Disturbance Noise Variance

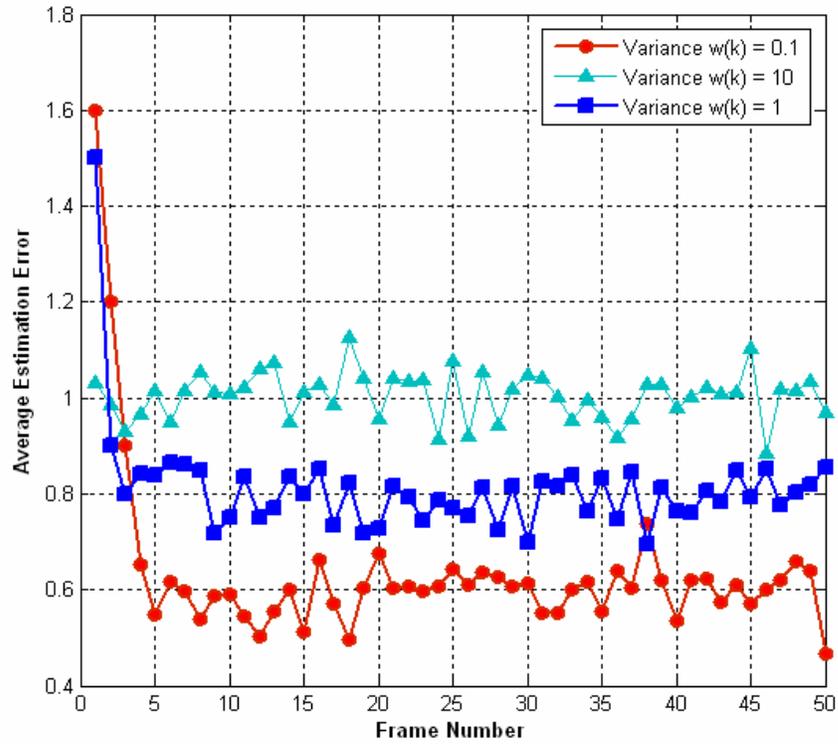


Figure 57 Plot of Frame Number vs. Estimation Error with respect to Disturbance Noise Variance

Comment:

When Figure 57 is analyzed, the effect of increasing the variance of the disturbance noise variance increases the average target path estimation error as expected.

4.4.6 Effect of the Observation Noise Variance

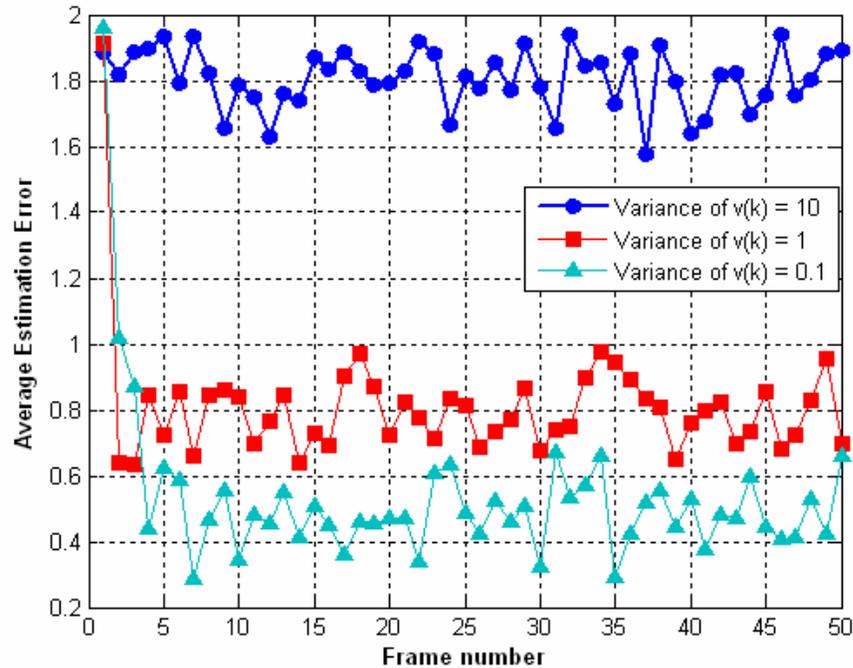


Figure 58 Plot of Frame Number vs. Estimation Error with respect to Observation Noise Variance

Comment:

Decreasing the observation noise variance reduces the average target path estimation error as expected at Figure 58. It is important to mention that for the dim target tracking algorithm, observation and disturbance noise statistics can be same, since the noise defined for the overall system only comes from AWGN which affects pixel intensity value. The result of pixel intensity change from the real value is the centroid variation along the target path. Furthermore, effect of noise variance for both types is very close in average estimation error.

4.4.7 Effect of Limiting the Maximum Possible Target Points Number

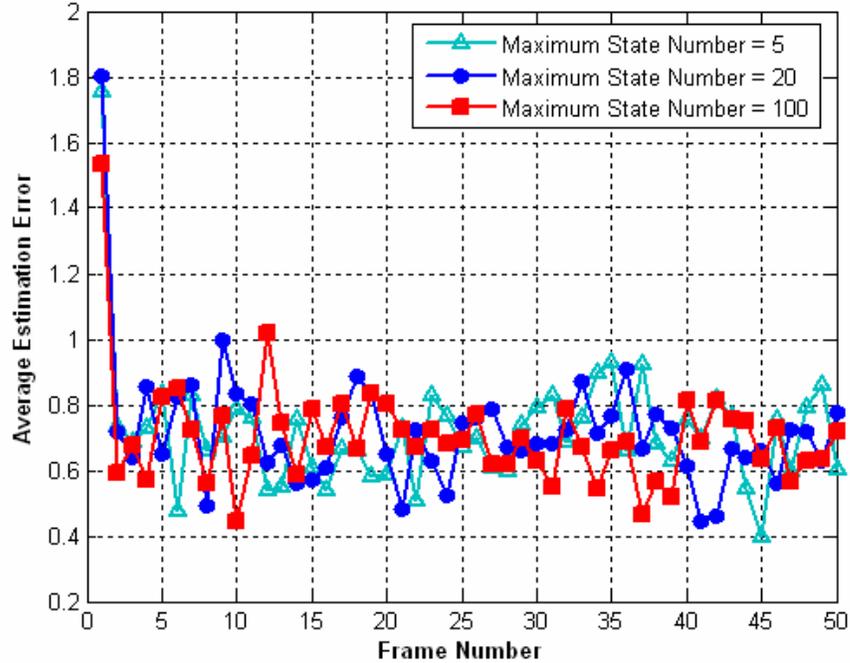


Figure 59 Plot of Frame Number vs. Estimation Error with respect to Maximum Target Points Number

Comment:

It can be observed from Figure 59 that changing the number of states did not affect the algorithm performance. The average estimation error does not degrade with limiting the possible target points. Since the program computation time is directly related with the Maximum Possible Target Points Number, the computation time can be reduced significantly by decreasing it.

4.5 Comparison of Kalman Filter with ODSA for Different Target Dynamics

4.5.1 Linear Motion

Let the target motion be given by the following state and observations equations for a “Linear Constant Velocity” Model:

$$\text{State: } \begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} \frac{r}{\sqrt{1+m^2}} & 0 \\ 0 & \frac{r.m}{\sqrt{1+m^2}} \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \quad (4.8)$$

$$\text{Observation: } \begin{bmatrix} ZX_k \\ ZY_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix} \quad (4.9)$$

where m and r are the constant angle and distance values of the target in two successive frames and can be given by the following equations:

$$m = \frac{Y_{k+1} - Y_k}{X_{k+1} - X_k} \quad \text{and} \quad r = \sqrt{(Y_{k+1} - Y_k)^2 + (X_{k+1} - X_k)^2} \quad (4.10)$$

The input is known for this model and given by the equation below:

$$\begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \forall k = 1, 2, \dots, n_s \quad (4.11)$$

where n_s is the number of observation frames.

The process and measurement noise covariance data are given by Equation 4.X.

$$E\left(\begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix}^T\right) = E\left(\begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix} \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix}^T\right) = E\left(\begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix}^T\right) = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad (4.12)$$

When the state and observation models are applied to ODSA and Kalman Filter with same parameters an observation values, the estimated path and the estimation error are given in the Figure 60 and Figure 61.

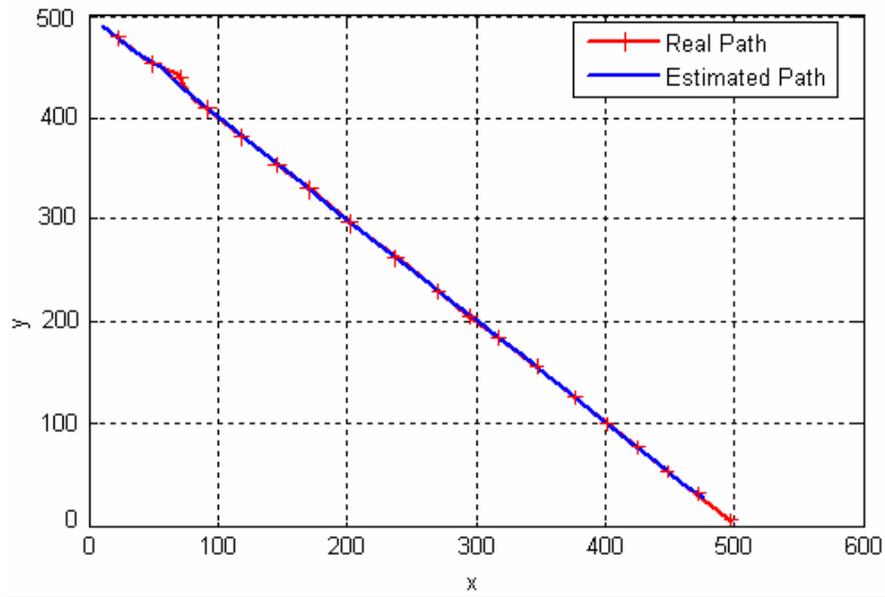


Figure 60 Estimated Path for Linear Motion using ODSA

Average Estimation Error: 0.996

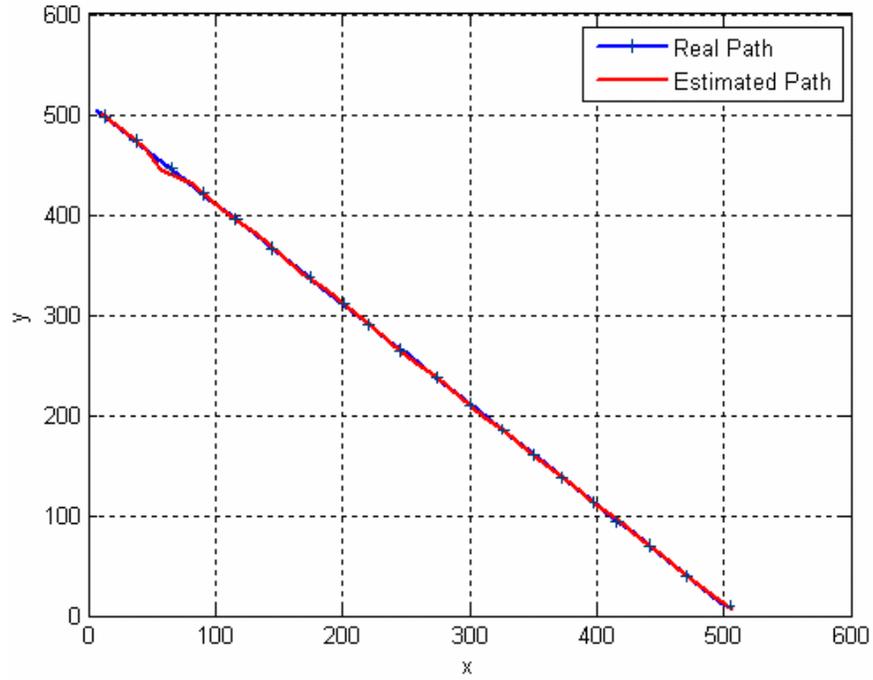


Figure 61 Estimated Path for Linear Motion using Kalman
Average Estimation Error: 1.023

4.5.1 Nonlinear Motion

For discrete nonlinear target motions, the path can be assumed to be piecewise linear between two time intervals. In this section, coordinates of the target is used as the state vector of the system. State and observation equations are driven according to piecewise linearity assumption.

- 1) Suppose that the target traverse a circle of radius r , starting at time $t = 0$ on the positive x axis. The direction of the target is counter clockwise and the angular velocity per frame, w , is constant. For n_s frames, the target path can be described parametrically by:

$$X_{k+1} - X_k = r. [\cos(w.k + w) - \cos(w.k)] \quad k = 1, 2, \dots, n_s \quad (4.13)$$

$$Y_{k+1} - Y_k = r. [\sin(w.k + w) - \sin(w.k)] \quad k = 1, 2, \dots, n_s \quad (4.14)$$

The state and observation models of the system can be given by the following equations:

$$\text{State: } \begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} \cos(w) - 1 & -\sin(w) \\ \sin(w) & \cos(w) - 1 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \quad (4.15)$$

$$\text{Observation: } \begin{bmatrix} ZX_k \\ ZY_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix} \quad (4.16)$$

$$\begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} = \begin{bmatrix} \cos(w.k) \\ \sin(w.k) \end{bmatrix} \quad (4.17)$$

When the state and observation models are applied to ODSA and Kalman Filter with same parameters an observation values, the estimated path and the estimation error are given in

Figure 62 and Figure 63.

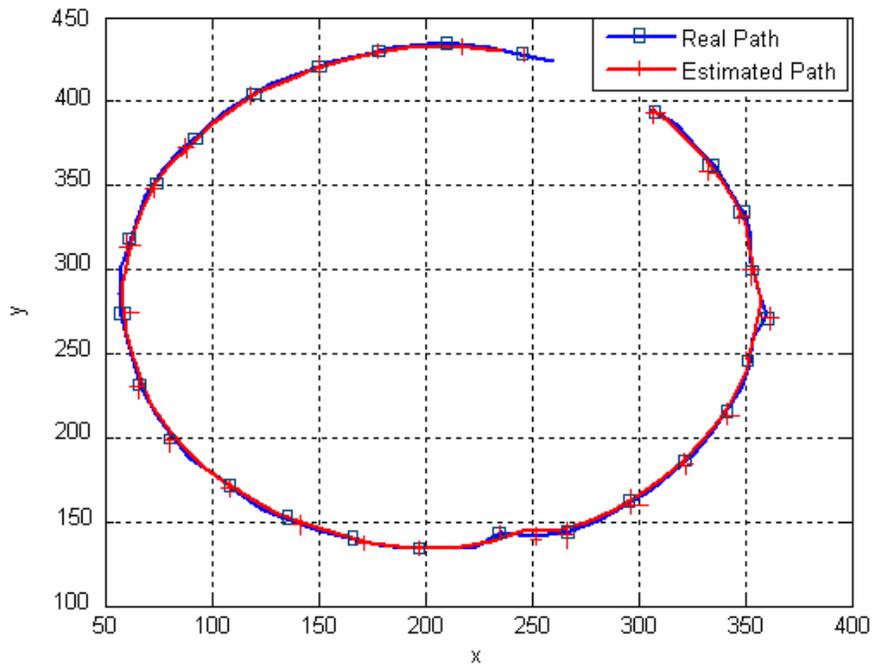


Figure 62 Estimated Path for Circular Motion using ODSA
Average Estimation Error: 0.644

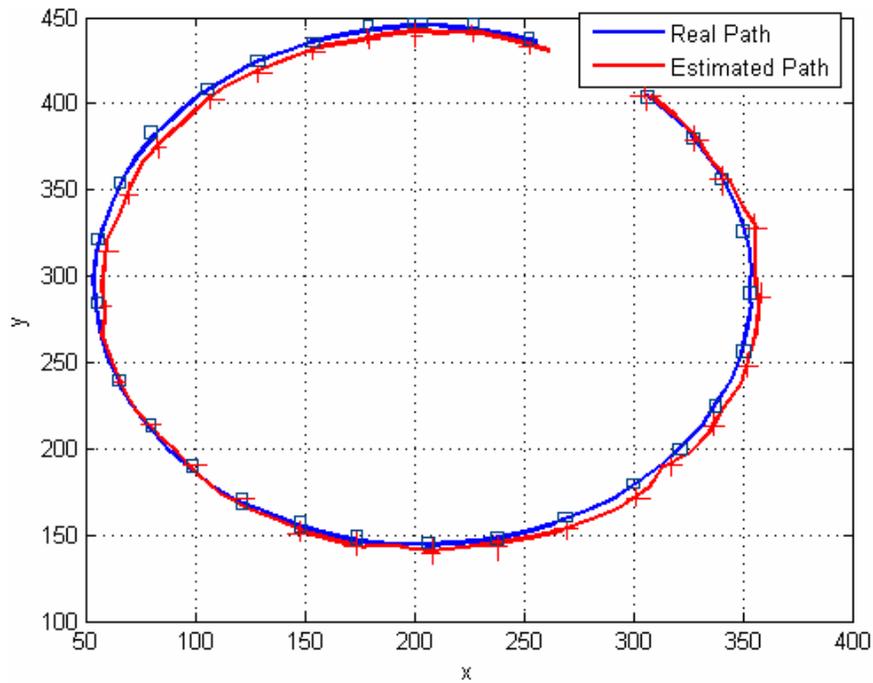


Figure 63 Estimated Path for Circular Motion using Kalman
Average Estimation Error: 3.067

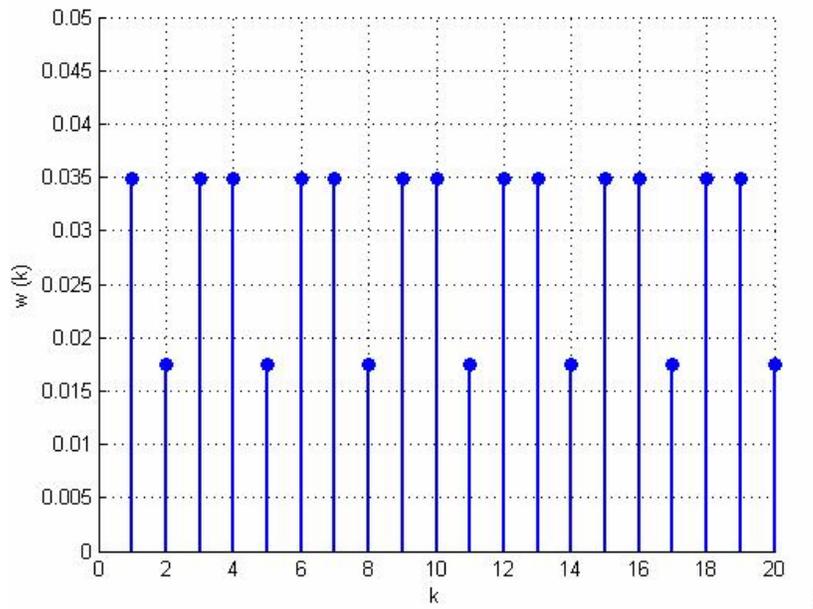


Figure 64 Plot of Angular Velocity per frame $w(k)$ with respect to time k

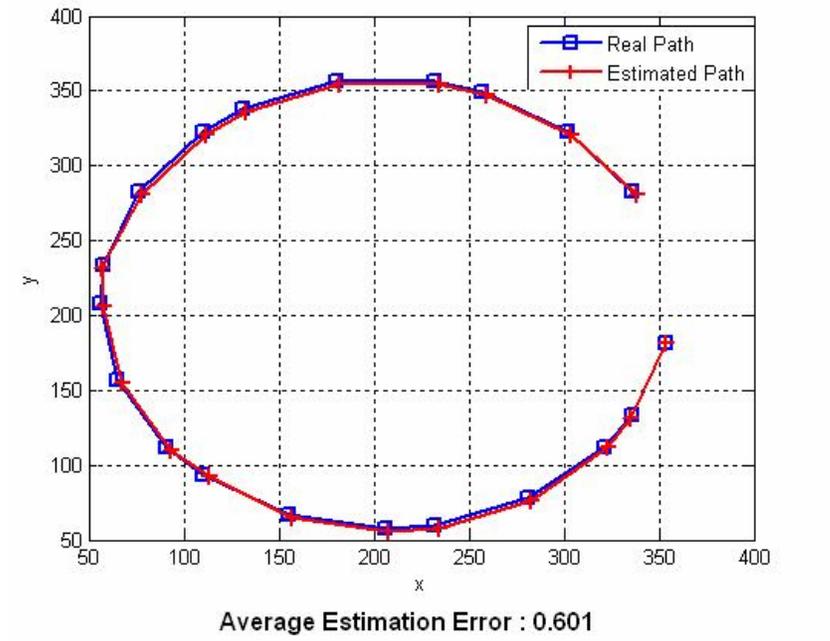


Figure 65 Estimated Path for Circular Motion with Changing Angular Velocity using ODSA
Average Estimation Error: 0.601

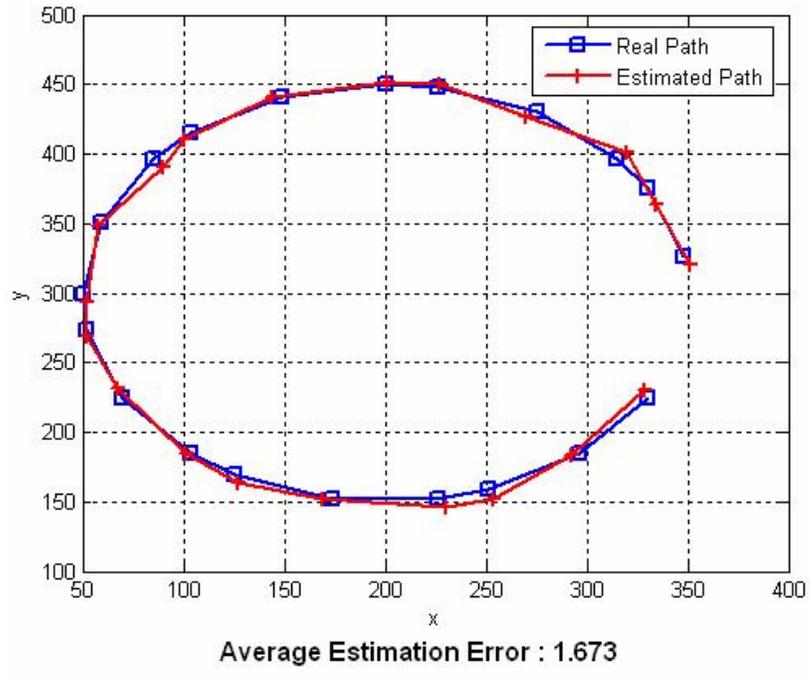


Figure 66 Estimated Path for Circular Motion with Changing Angular Velocity using Kalman
Average Estimation Error: 1.673

2) Suppose that the target traverse a parabolic path with its focus point (0, c). The equation of the parabola is given below:

$$y = \frac{x^2}{4.c} \quad (4.18)$$

Let the target start to motion from the point: $\left(m, \frac{m^2}{4.c}\right)$

The state and observation equations of the motion are given by the following equations:

$$\text{State: } \begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \quad (4.19)$$

$$\text{Observation: } \begin{bmatrix} ZX_k \\ ZY_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix} \quad (4.20)$$

$$\begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} = \begin{bmatrix} k+m \\ \frac{(k+m)^2}{4c} \end{bmatrix} \quad (4.21)$$

$k = 1, 2, \dots, n_s$ and n_s is the total number of frames.

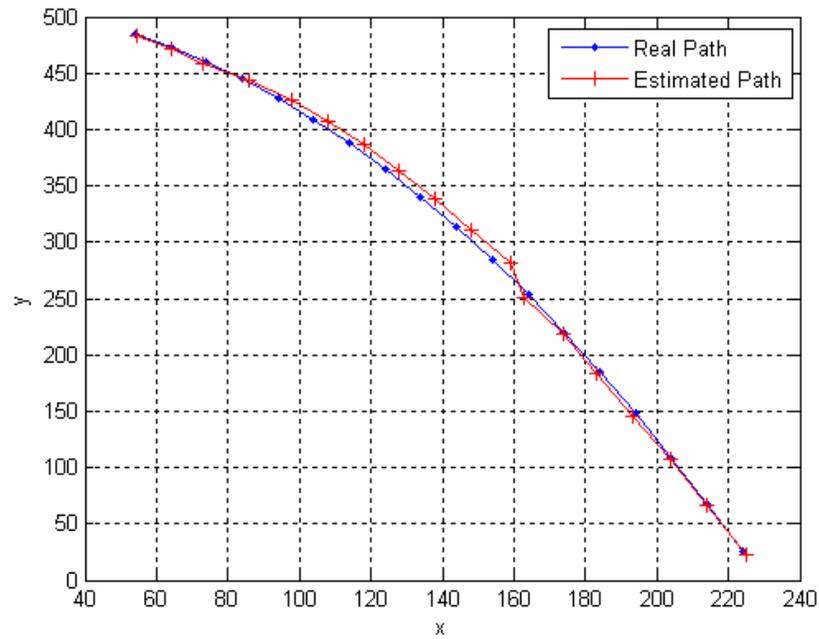


Figure 67 Estimated Path for Parabolic Motion using ODSA
Average Estimation Error: 0.918

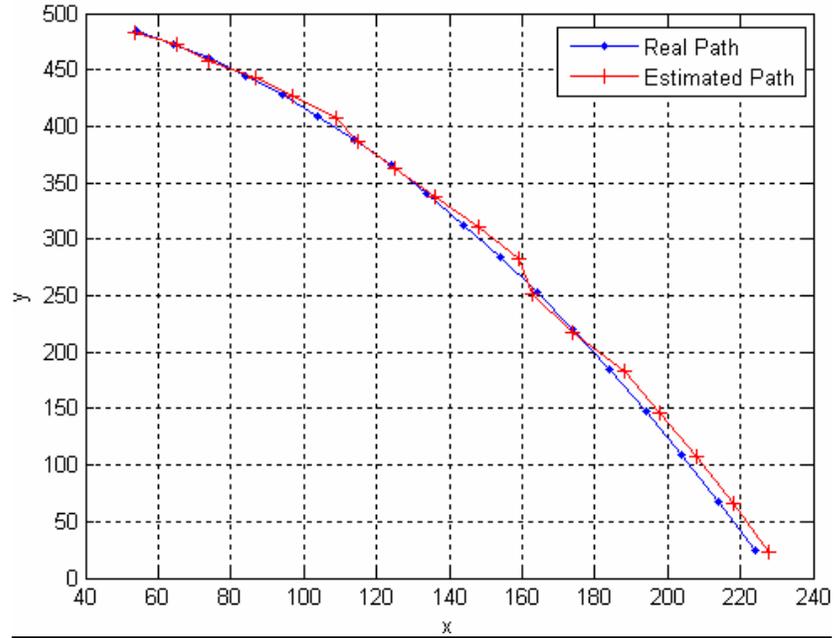


Figure 68 Estimated Path for Parabolic Motion using Kalman
Average Estimation Error: 1.837

- 3) Suppose that the target traverse a half elliptic path with its major and minor axis lengths a and b respectively. The equation of the upper half of the ellipse is given below:

$$y = b\sqrt{1 - \frac{x^2}{a^2}} \quad (4.22)$$

Let the target start to motion from the point: $(a,0)$

$$\text{State: } \begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \quad (4.23)$$

$$\text{Observation: } \begin{bmatrix} ZX_k \\ ZY_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix} \quad (4.24)$$

$$\begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} = \begin{bmatrix} -\frac{2a}{n_s}(k+1)+a \\ b\sqrt{1-\frac{\left(-\frac{2a}{n_s}(k+1)+a\right)^2}{a^2}} \end{bmatrix} \quad (4.25)$$

$k = 1, 2, \dots, n_s$ and n_s is the total number of frames.

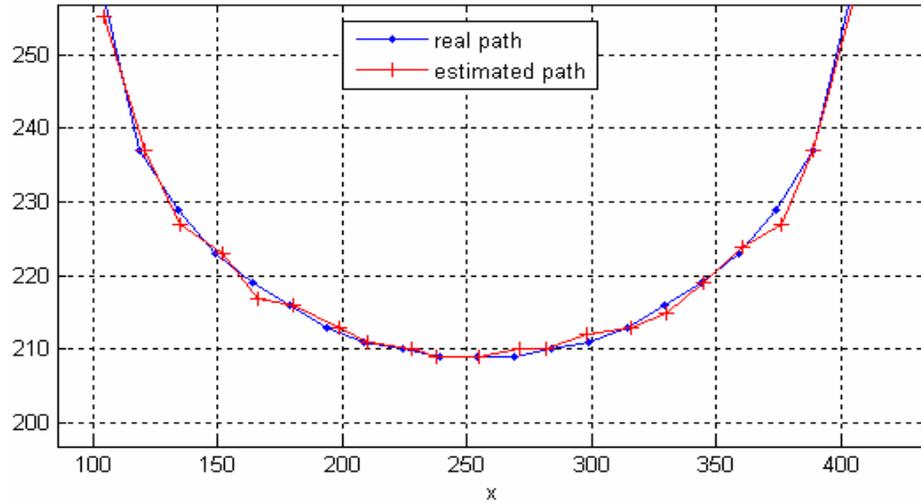


Figure 69 Estimated Path for Elliptic Motion using ODSA
Average Estimation Error: 0.591

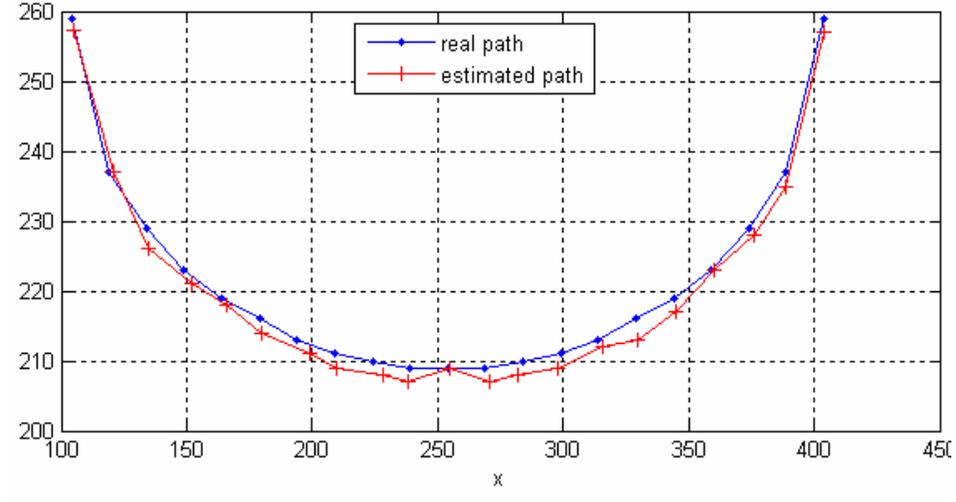


Figure 70 Estimated Path for Elliptic Motion using Kalman
Average Estimation Error: 1.123

- 4) Suppose that the target traverse a polynomial path with the equation given below:

$$y = a_1(x - r_1)(x - r_2)(x - r_3) + a_2 \quad (4.26)$$

Let the target start to motion from the point: (r_1, a_2)

$$\text{State: } \begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} w_{1k} \\ w_{2k} \end{bmatrix} \quad (4.27)$$

$$\text{Observation: } \begin{bmatrix} ZX_k \\ ZY_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} + \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix} \quad (4.28)$$

$$\begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix} = \begin{bmatrix} r_1 + k \\ a_1 \cdot k \cdot (x - r_2)(x - r_3) + a_2 \end{bmatrix} \quad (4.29)$$

$k = 1, 2, \dots, n_s$ and n_s is the total number of frames.

The equation used for the simulation is given below:

$$y = 5.10^{-5} \cdot (x - 50)(x - 200)(x - 450) + 400 \quad (4.30)$$

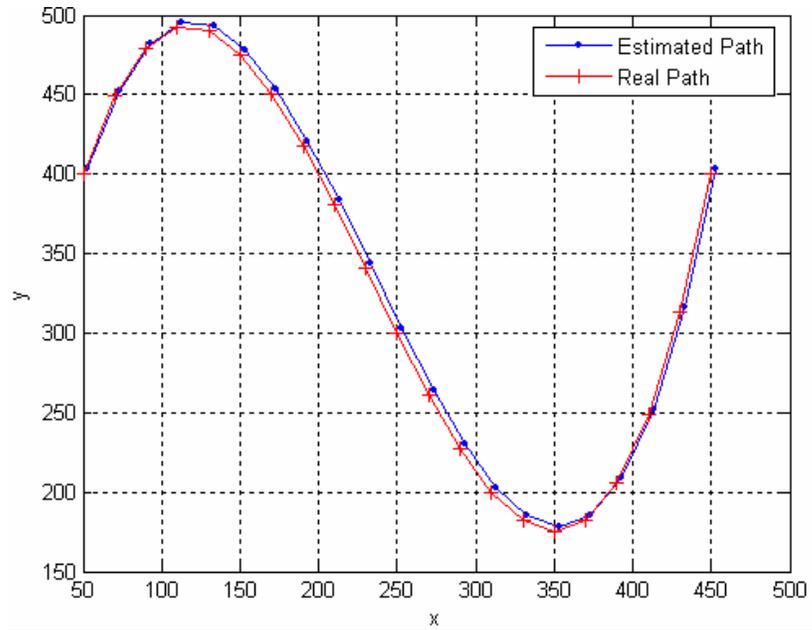


Figure 71 Estimated Path for Changing Maneuver Motion using ODSA
Average Estimation Error: 1.934

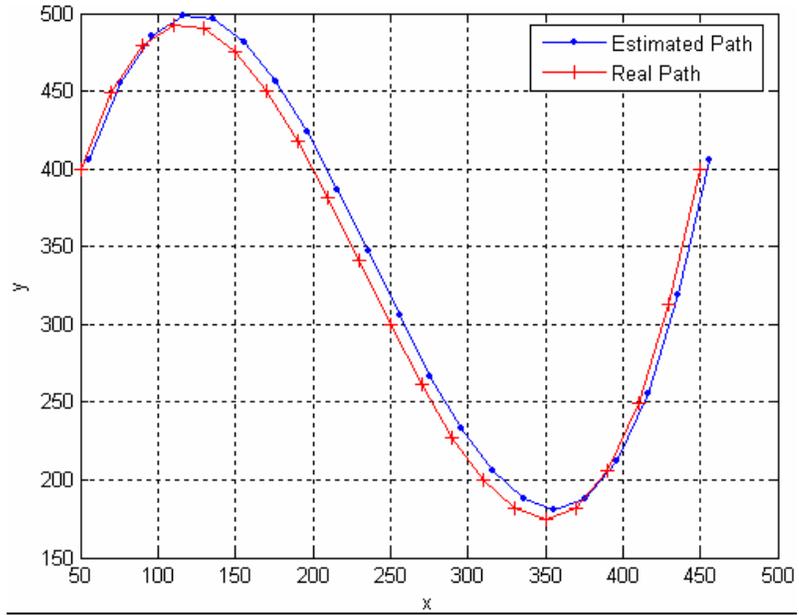


Figure 72 Estimated Path for Changing Maneuver Motion using Kalman
Average Estimation Error: 2.865

4.5.1 Maximum Tolerable Noise

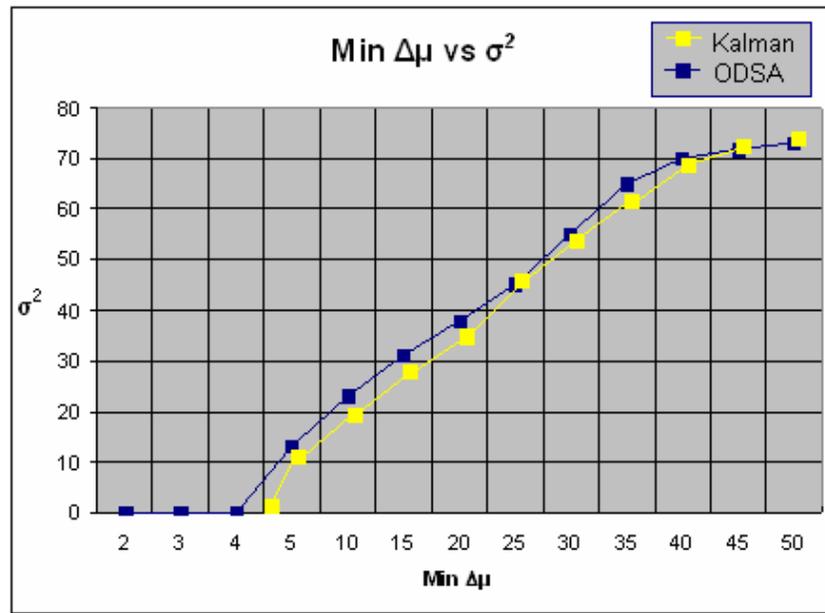


Figure 73 Plot of Minimum Tolerable Difference of Target and Environment Mean Intensity vs. Noise Variance for ODSA and Kalman

Comment:

From Figure 60 and Figure 61 , it can be observed that average estimation errors for linear motion of the target when ODSA and Kalman Filter are used are very close. However, in the circular motion case, and, Kalman Filter performance degrades as compared with ODSA from

Figure 62 and Figure 63.

If the angular velocity of the circular motion is subject to change in Figure 64, the estimated path of the target can be observed from Figure 65 and Figure 66. Again performance of ODSA is better when compared to Kalman Filter.

From Figure 67 and Figure 68 is related parabolic motion. Figure 69 and Figure 70 shows the elliptic path of the target. Polynomial, i.e. changing maneuver is implemented in Figure 71 and Figure 72. In all of these different motion dynamics, the average estimation error is less for ODSA when compared to Kalman Filter.

Furthermore, the effect of the noise variance is investigated for both of the estimation algorithms. ODSA endures more noise variance for the same target and background mean difference when compared to Kalman Filter as can be seen from Figure 73.

Finally, it is important to mention that ODSA gives estimation data after more number of observation samples as compared to Kalman. The estimation error difference comes from this more computational load. Nevertheless, Kalman is more useful for real time applications which require fast computation. The prediction concept is not applicable for ODSA also. On the other hand, Kalman Filter can be used in the area of next step estimation applications.

CHAPTER 5

CONCLUSIONS

In this study, a visual dim single target tracking algorithm is developed for low observable and dense environments with a stationary camera looking at a fixed region of interest. Methodologies of the *Dynamic Vision* and *Optimum Decoding Based Smoothing Algorithms* are explained and some simulations are performed displaying the performances of the algorithm.

Dynamic Vision Algorithm consists of 8 main parts:

1. *Change Detection by Likelihood Ratio Testing*
2. *Labeling and Destruction of Difference Matrix to Subgroups*
3. *Subgroup Centroid Mapping*
4. *Maximization of Distribution Density Functions for Centroids*
5. *Size Filtering by Maximum Histogram Method*
6. *Range and Maneuver Filtering*
7. *Target Vector Calibration with respect to the focus window*
8. *Consecutive Label Decision*

Change Detection by Likelihood Ratio Testing is the statistical comparison of the two consecutive image frames to detect changing regions. The superpixel, a group of pixels which has the sufficient statistics for comparison, is introduced. The threshold value used for comparison is adaptive for the elimination of false alarms.

Labeling and Destruction of Difference Matrix to Subgroups is the grouping of changing regions in a 4-connected geometrical manner. The label that is to be analyzed is destructed to subgroups for the reduction of effect of noise on candidate target movement vectors in the next frame.

Subgroup Centroid Mapping is the matching of each subgroup superpixel element to the candidate centroid points of the labels in the next frame. For matching, LRT is used in a reverse manner of change detection.

Maximization of Distribution Density Functions for Centroids is the decision of maximum probable target movement for a subgroup superpixel element by using two dimensional image filters.

Size Filtering by Maximum Histogram Method is the target extraction method from a given centroid point distribution which is the output of *Maximization of Distribution Density Functions for Centroids* for all subgroups in the analyzed label.

Range and Maneuver Filtering is the filtering of impossible target movements according to the last movement.

Target Vector Calibration with respect to the focus window is the mapping of decided target vector to the main frame coordinates from the focus window coordinates.

Consecutive Label Decision is the assignment of next iteration label according to the last observed target centroid.

The output of the *Dynamic Vision Algorithm*, observation centroids of the tracked target, is used as the input for ODSA which is based on Viterbi decoding algorithm. By reducing the target motion to a finite state model which uses the quantized state vector, a trellis diagram is obtained; and then, the state vector and target centroid points are estimated by finding the most probable path in the trellis diagram.

The aim of the simulations is to investigate the effects of the observation, target, risk and ODSA parameters. When the simulations performed for the overall algorithm are considered, the results can be summarized as below:

- The algorithm can tolerate target SNR values down to ± 0.8 dB and target size of 10 pixel^2 ,
- The algorithm can track targets which has different types of motion characteristics with maximum RMS error of 2.5 pixels approximately,
- Minimum detectable target velocity is around 10 pixels,
- Maximum tolerable illumination rate is approximately 10 pixel intensity,
- The risk parameters affect the probability of detection. As the risk gets higher, the computation time decreases with the tradeoff decrease in the probability of detection,
- Image Size, superpixel size, gate size affects algorithm complexity significantly. As the gate size becomes smaller, the estimation performance increases,
- The quantization number of the initial state vector is effective for initial times,
- The quantization number of the disturbance noise vector affects the estimation performance significantly,
- Increasing the disturbance noise variance degrades the performance significantly,

- The initial state variance affects only the performance for small frame numbers
- Increasing the observation noise variance degrades the estimation performance,
- The maximum number of possible target points can be limited without degrading the estimation performance.

REFERENCES

- [1] *“Target tracking in the presence of interference”* K. Demirbaş, Ph.D. Thesis, University of California, Los Angeles, 1981.
- [2] *“Machine Vision”*(pages 406-413) R. Jain, Univ. of California, San Diego, R. Kasturi, Pennsylvania State Univ., B. G. Schunck, Univ. of Michigan (1995)
- [3] *“Target Detection from image sequences using pixel-based decision criterion”* P.Weï, Univ. of California/San Diego; J. R. Zeidler, Univ. of California/San Diego and Naval Ocean Systems Ctr.; W. H. Ku, Univ. of California/San Diego.
- [4] *“Single-look tracking noise for intensity and binary centroiding algorithms for resolved targets and performance comparison between shared- and separate-aperture active tracking”* L.W. Bandermann, R.Kwong, Lockheed Missiles & Space Co., Inc.
- [5] *“Use Of Measurements From An Imaging Sensor For Precision Target Tracking”*, Y. Bar-Shalom, Univ. of Connecticut, H.M. Shertukde, Univ. of Hartford, K.R. Pattipati, Univ. of Connecticut
- [6] *“An improved Tracking Algorithm for Target Detection”* K.Anil Kumar, G.Raghava Rao and N.L.M.Murukutla R & T IJnit for Navigational Electronics, Univ. of Osmania
- [7] *“Detection of Dim Targets In High Cluttered Background Using High Order Correlation Network”* Ren-Jean Liou, Colorado State University, Mahmood R. Azimi-Zadjadi, Colorado State University , Roy Dent, IBM Corporation
- [8] *“Dim Target Detection Using High Order Correlation Method”* Ren-Jean Liou, Colorado State University, Mahmood R. Azimi-Zadjadi, Colorado State University
- [9] *“Efficient Target Tracking Using Dynamic Programming”* James Arnold, Scott Shaw, Henry Pasternack, Members of IEEE
- [10] *“Maximum Likelihood Track- Before-Detect With Fluctuating Target Amplitude”* S. M. TONISSEN, Member, IEEE, Y. BAR-SHALOM, Univ.of Connecticut
- [11] *“Optimal Multiframe Detection and Tracking in Digital Image Sequences”* Marcelo G. S. Bruno, Univ. of Sao Paulo, Jose M. F. Moura. Massachusetts Institute of Technology
- [12] *“Detection, Estimation, and Modulation Theory”*, Harry L. Van Trees, Massachusetts Institute of Technology

- [13] “*Principles of Digital Communication and Coding*” A. J. Viterbi and J. K. Omura, McGraw-Hill, New York, 1979
- [14] “*Estimation and tracking: principles, techniques and software*”, Y. Bar-Shalom, Xiao- Rong Li, 1993
- [15] “*Matlab Program Version 7.0 Help – Routine: bwlabel(.)*”
- [16] “*Visual Tracking With Group Motion Approach*”, Mübeccel Demirekler, Ali Erkin Arslan, Middle East Technical University
- [17] “*Robust Adaptive Target State Estimation for Missile Guidance using the Interacting Multiple Model Kalman Filter*”, Constantino Rago, Raman K. Mehra, Scientific Systems Company, Inc.
- [18] “*A New Method For Passive Location Estimation From Image Sequence Using Adaptive Extended Kalman Filter*”, Yao Jianchao, DSO National Laboratories
- [19] “*On total least squares fitting spatiotemporal derivatives to smooth opticalflow fields*” Shang Wang, Vishal Markandey, Anthony Reid, Texas Instruments
- [20] “*Target Tracking with Input Estimation*” Ersen Gazioğlu, Aselsan Inc.