MOVING OBJECT DETECTION IN 2D AND 3D SCENES

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

BY

SALİM SIRTKAYA

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

SEPTEMBER 2004

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. Mübeccel Demirekler
Head of Department

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

_____

Assoc. Prof. Dr. Aydın Alatan
Supervisor

Examining Committee Members

Prof. Dr. Mübeccel Demirekler (METU, EE)_____

Assoc. Prof. Dr. Aydın Alatan (METU, EE)_____

Prof. Dr. Kemal Leblebicioğlu (METU, EE)_____

Assoc. Prof. Dr. Gözde Bozdağı Akar (METU, EE)_____

Assoc. Prof. Dr. Yasemin Yardımcı (METU, IS)_____

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.

Salim SIRTKAYA

ABSTRACT

MOVING OBJECT DETECTION IN 2D AND 3D SCENES

Sırtkaya, Salim

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Aydın Alatan

September 2004, 88 Pages

This thesis describes the theoretical bases, development and testing of an integrated moving object detection framework in 2D and 3D scenes. The detection problem is analyzed in stationary and non-stationary camera sequences and different algorithms are developed for each case. Two methods are proposed in stationary camera sequences: background extraction followed by differencing and thresholding, and motion detection using optical flow field calculated by “Kanade-Lucas Feature Tracker”. For non-stationary camera sequences, different algorithms are developed based on the scene structure and camera motion characteristics. In planar scenes where the scene is flat or distant from the camera and/or when camera makes rotations only, a method is proposed that uses 2D parametric registration based on affine parameters of the dominant plane for independently moving object detection. A modified version of the 2D parametric registration approach is used when the scene is not planar but consists of a few number of planes at different depths, and camera makes translational motion. Optical flow field segmentation and sequential registration are the key points for this case. For

3D scenes, where the depth variation within the scene is high, a parallax rigidity based approach is developed for moving object detection.

All these algorithms are integrated to form a unified independently moving object detector that works in stationary and non-stationary camera sequences and with different scene and camera motion structures. Optical flow field estimation and segmentation is used for this purpose.

Keywords: Background Extraction, Optical Flow Field, Structure from Motion, Affine Parameter Estimation, Parallax Rigidity

ÖZ

2 BOYUTLU VE 3 BOYUTLU SAHNELERDE HAREKETLİ NESNE TESPİTİ

Sırtkaya, Salim

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç.Dr. Aydın Alatan

Eylül 2004, 88 Sayfa

Bu tez, 2 boyutlu ve 3 boyutlu sahnelerde tümleşik hareketli hedef tespiti çözüm iskeletinin kuramsal taban, geliştirme ve test etme aşamalarını anlatmaktadır. Tespit problemi, sabit ve hareketli kamera sahnelerinde tahlil edilmiş ve her iki durum için ayrı algoritmalar geliştirilmiştir. Sabit kamera sahneleri için iki ayrı yöntem sunulmuştur: Arkaplan özütlemeyi takip eden çıkarım eşikleme, ve "Kanade-Lucas Öznitelik İzleme" tekniği ile hesaplanan ışıl akış alanından hareket tespiti. Hareketli kamera sahneleri için sahne yapısı ve kamera hareket tipine bağlı olarak değişik algoritmalar geliştirilmiştir. Kameradan uzakta veya yassı olan düzlemsel sahnelerde ve/veya kamera sadece dönüş hareketi yaptığında, bağımsız hareket eden nesne tespiti baskın düzlemin ilgin parametrelerine dayalı 2 boyutlu parametrik çakıştırma yöntemiyle yapılmıştır. Sahnenin bir yerine birkaç düzlemden oluştuğu ve kameranın ötelenme hareketi yaptığı durumlarda, parametrik çakıştırma yönteminin değiştirilmiş bir sürümü kullanılmıştır. Bu durumda, ışıl akış alanı kesimlemesi ve ardışık çakıştırma anahtar noktalardır.

Derinlik değişiminin fazla olduğu 3 boyutlu sahneler için, parallaks sabitliği tabanlı bir yöntem geliştirilmiştir. 2 boyutlu parametrik çakıştırma bu yöntemde de kameranın dönüş hareketi ve optik kaydırmadan kaynaklanan etkileri ortadan kaldırmak için kullanılmaktadır.

Tüm bu algoritmalar, sabit, hareketli kamera ve değişik sahne yapısı ve kamera hareket tipleri ile çalışabilecek, tümleşik bağımsız hareket eden nesne tespiti yapabilmek için birleştirilmişlerdir.

Anahtar Kelimeler: Arkaplan Özütleme, Işıl Akış Alanı, Hareketten Yapı, İlgin Parametre Kestirimi, Parallaks Sabitliği.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Assoc. Prof. Dr. Aydın Alatan for his continued guidance, also patience and support throughout my research. He provided counsel, and assistance that greatly enhanced my studies and know-how.

I am also grateful to Tuba M. Bayık for her support, patience and belief in me. She consistently assisted and motivated me during the whole master's period.

Thanks go to my manager Dr. Murat Eren for his patience and support from beginning to end of my M.Sc. program. I am also grateful to several of my colleagues, especially those who shared my office and thereby my problems. These include Naci Orhan, Onur Güner, Alper Öztürk and Volkan Nalbantoğlu.

And my parents and sisters, I want to thank them for everything.

ASELSAN A.Ş. who supported this work is greatly acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

IMO	Independently Moving Object
SCMO	Stationary Camera Moving Object
MCMO	Moving Camera Moving Object
2D	Two Dimensional
3D	Three Dimensional
IR	Infrared
KLT	Kanade Lucas Tracker
BG	Background
FG	Foreground
FR	Frame
Thr	Threshold
MS	Microsoft
MFC	Microsoft Foundation Class

CHAPTER 1

INTRODUCTION

1.1 Background

Most biological vision systems have the talent to cope with changing world. Computer vision systems have developed in the same way. For a computer vision system, the ability to cope with moving and changing objects, changing illumination, and changing viewpoints is essential to perform several tasks [1].

Independently moving object (IMO) detection is an important motion perception capability of a mobile observatory system. IMO detection is necessary for surveillance applications, for guidance of autonomous vehicles, for efficient video compression, for smart tracking of moving objects, for automatic target recognition (ATR) systems and for many other applications [6][21].

The changes in a scene may be due to the motion of the camera (ego-motion), the motion of objects, illumination changes or changes in the structure, size or shape of an object. The objects are assumed rigid or quasi-rigid; hence, the changes are generally due to camera and/or object motion. Therefore, there are two possibilities for the dynamic nature of the camera and world setup concerning IMO detection:

1. Stationary camera, moving objects (SCMO)

2. Moving camera, moving objects (MCMO)

For analyzing image sequences, different techniques are required in each of the above cases. In dynamic scene analysis, SCMO scenes have received the most attention. A variety of methods to detect moving objects in static scenes has

been proposed [1][3][4][5]. While examining such scenes, the goal is usually to detect motion, to extract masks of moving objects for recognition, and to compute their motion characteristics. Utilization of difference pictures [1][5], background elimination [10][11], optical flow computation [17][18] are typical tools that are constructed for detection of moving objects.

MCMO is the most general and possibly the most difficult case in dynamic scene analysis, but it is also the least developed area of computer vision [1]. The key step in IMO detection in MCMO case is compensating for the camera induced motion. A variety of methods to compensate for the ego-motion has been proposed [6][8][19][21][27].

Irani [6] have proposed a method for the robust recovery of ego-motion. The method is based on detecting two planar surfaces with different depths in the scene and computing their 2D motion in the image plane. Although it is possible to calculate 3D camera motion from the 2D motion of a single plane, the 2D motion difference of two planes are used to make the relations linear and decrease the complexity. After calculating the camera parameters, the remaining task is the registration of the images and segmenting out the residual motion areas belonging to IMOs. This methods works well in controlled environments, such as the cases ıin which the camera is allowed to make certain motions, the scene has enough depth variation, the number and size of IMOs are small and there exists at least two planes in the scene. However, generalization of the algorithm for outdoor scenes is an ill-conditioned problem, and biasing of IMOs biases the ego-motion estimation.

Aggarwall [8] have also proposed a method to remove the ego-motion effects through a multi-scale affine registration process. The method assumes small IMOs, and calculates affine motion parameters of the dominant background in a Laplacian image resolution hierarchy. After registration with calculated affine parameters, areas with residual motion indicate potential object activity. This detection scheme is reliable for remote or planar scenes, but it gives poor results in 3D scenes where depth variation and camera translation creates parallax motion.

Adiv [9] have proposed a method to detect IMOs by partitioning the optical flow field, generated by the camera and/or object motion, into connected segments where each segment is consistent with a rigid motion of a roughly planar surface

and therefore, is likely to be associated with only one rigid object [9]. The segments are grouped under the hypotheses that they are induced by a single rigidly moving object. This scheme makes it possible to deal with IMOs by analyzing the motion of the segmented 3D objects in time. Objects belonging to the static background will have constant motion parameters unlike those of IMOs. This detection scheme is reliable for the 3D scenes where depth variation is significant and camera makes enough translation. Nevertheless, there exists inherent instabilities in recovering 3D motion from noisy flow fields.

Chellappa and Qian [27] have proposed a method to moving object detection with moving sensors based on sequential importance sampling. Their method is based on detecting feature points in the first image of the sequence and tracking these feature points to find an approximate sensor motion model. The algorithm then segments out the feature points belonging to the moving object. Their method works both with 2D and 3D scenes, however, feature selection is inherently problematic and the proposed algorithm has an off-line character.

Michal Irani and P. Anandan [6] have also proposed a unified approach to moving object detection in 2D and 3D images. Their detection scheme is based on a stratification of the moving object detection problem into scenarios that gradually increase in their complexity. In 2D scenes, the camera-induced motion is modeled in terms of a global 2D parametric transformation and the transformation parameters are used for image registration. This approach is robust and reliable only when applied to flat (planar) scenes, distant scenes or when the camera is undergoing only rotations and/or zooms. When the camera is translating and the scene is not planar, a modified version of the 2D parametric registration is applied [6]. In this scheme [6], the camera induced motion is modeled in terms of a few number of layers of 2D parametric transformations and the registration is done sequentially. When the scene contains large depth variations (3D scenes) and camera makes considerable translational motion, the 2D parametric registration approaches fail. Hence, the proposed method [6] selects a plane for registration and applies a parallax based rigidity constraint over the registered image for 3D scenes. Since parallax motion occurs due to depth differences of the points, and have a ‘relative projective structure rigidity’ for the points that belong to the static background, then this knowledge is used to detect IMOs in 3D scenes. The unified

approach bridges the gap between the 2D and 3D approaches, by making the calculations at each complexity level the basis for the next complexity level. Irani leaves the integration of the algorithms into a single algorithm, as a future work. A solution for such an integration algorithm is proposed in this thesis.

1.2 Objectives

In this thesis, two algorithms are proposed for SCMO case. The first algorithm is based on well-known background subtraction and thresholding. The difference between the current image and the extracted background is thresholded for moving object detection. Morphological operations, such as erosion and dilation, are also added to the algorithm, as discussed in [8]. The second algorithm is based on calculation and segmentation of the optical flow field, since the optical flow field in SCMO case will mostly be induced by moving objects.

MCMO case is analyzed for three different schemes (2D, Multilayer, 3D) as suggested in [6]. The 2D parametric model estimation is implemented different from their suggestion. Instead of locking onto a dominant plane, as suggested in [22][23], the optical flow field is calculated and segmented using the affine motion parameters implicitly in the segmentation process. Optical flow field segmentation brings out the advantage of analyzing the scene for algorithm integration.

In real image sequences, it is not possible to predict in advance which situation (stationary camera, moving camera, 2D images, and 3D images) will occur. Moreover, different types of scenarios can occur within the same sequence. Therefore, an integration of the algorithms for different situations into a single algorithm is required for full automation. An optical flow and segmentation based scheme is suggested for such an integration.

The objective of this thesis is to examine and implement ‘independently moving object detection’ with the most reliable and robust algorithms for different scenarios, and finally integrate these algorithms into a single algorithm reliably. The performance of such algorithms for infrared (IR) sequences is also being tested.

1.3 Thesis Outline

Chapter 2 provides the necessary theoretical bases for background extraction, thresholding, morphological operations and optical flow calculation. Using these tools, two different algorithms are examined for moving object detection by using a stationary camera.

Chapter 3 provides the necessary theoretical bases for 2D projective modeling of 3D velocity of a plane in terms of affine parameters, segmentation of the optical flow field using affine parameters, 3D scene analysis, parallax motion and parallax rigidity estimation. Using these models and estimations, different independently moving object detection algorithms are designed for 2D, multilayer and 3D cases where camera is non-stationary. Finally, the unification of these non-stationary and stationary camera algorithms into a single algorithm is also discussed.

Chapter 4 provides simulation results for different independently moving object detection schemes. The algorithms are applied to real and artificial sequences separately, in order to differentiate the estimation noise from sensor noise. The algorithms are tested with day and thermal camera images and with very different cases to analyze the noise compensation, reliability and robustness. The results for different cases are discussed in this section.

Chapter 5 provides the summary for the overall study as well as some concluding remarks and some open points for possible future studies.

CHAPTER 2

MOVING OBJECT DETECTION WITH STATIONARY CAMERA

2.1 Introduction

Moving object detection problem with a stationary camera is the basic step of moving object detection, since the inherent ambiguities of ego-motion (camera motion) is not considered and the scene structure (depth variations etc.) can be discarded at the very beginning. However, still there must be some processing concerning the gradual and sudden illumination changes (such as clouds), camera oscillations, high frequency background objects, such as tree branches or sea waves, biasing of moving objects and the noise coming from the camera (especially in thermal cameras).

Two approaches, concerning the above stated noise issues, are introduced for the detection problem. The first approach is extraction and elimination of the background and differentiation afterwards [5]. In order to eliminate the effects of noise, thresholding and simple image processing operations are added to the differentiation step [1][5][12]. The second approach is calculation of the optical flow vectors, namely the motion vector at each pixel, and thresholding the calculated optical flow field between successive frames.

Both approaches have advantagess and disadvantages concerning their accuracy, robustness, noise resistance and processing time. Background elimination technique is simple and effective, but it can be erroneous on noisy data.

On the other hand, the optical flow approach is more robust whereas requires more processing.

2.2 Background Elimination and Thresholding Approach

In stationary camera case, if the background of the scene is known and can be assumed to stay constant, simple subtraction of the intensities of the current image and the background should be enough to detect the foreground, in this case the moving object. However, generally the background is not predetermined and does change due to illumination and scene conditions. Therefore, extraction and currency of the background and noise removal after differencing become key issues for this approach.

2.2.1 Background Extraction

The simplest way of background extraction is user intervention. Although this approach seems to be primitive, it works in situations, where the background does not change considerably (indoor applications) and processing time is a critical issue. This approach is still very sensitive to the selected threshold, which will be used after subtraction. Another approach can be selection of the previous frame as the background. This approach also depends on the threshold selection and the objects speed and the frame rate.

A robust way of determining background intensities is taking the background as the average of previous n frames or using a moving average in order to minimize the memory requirements [10]. In simple average case, the memory requirement is 'n times the frame size', whereas in moving average, it is just the frame size. The relation below defines the extraction of the background by using moving average:

$$BG_{i+1} = \alpha * FR_i + (1-\alpha) * BG_i \qquad (2\text{-}1)$$

where, α is the learning rate of the algorithm and is typically 0.05, BG_i is the estimated background for the i^{th} frame, and FR_i is the i^{th} frame itself. Moving

average approach can also be used with some selectivity, such that if some pixels are chosen as the foreground (object) they do not contribute to the background in the next step:

$$BG_{i+1}(x,y) = \begin{cases} \alpha * FR_i(x,y) + (1-\alpha) * BG_i(x,y) & 'if\ FR_i(x,y)\ is\ background' \\ BG_i(x,y) & 'if\ FR_i(x,y)\ is\ foreground' \end{cases} \quad (2\text{-}2)$$

2.2.2 Thresholding

After finding the background, the foreground object detection is achieved by subtracting and thresholding the difference image [1][5].

$$\begin{aligned} FR_i(x,y)\ is\ foreground \quad & if\ \left|FR_i(x,y) - BG(x,y)\right| > Thr \\ FR_i(x,y)\ is\ background \quad & if\ \left|FR_i(x,y) - BG(x,y)\right| < Thr \end{aligned} \quad (2\text{-}3)$$

Thresholding is a fundamental method to convert a gray scale image into a binary mask, so that the objects of interest are seperated from the background [1]. In the difference image, the gray levels of pixels belonging to the foreground object should be different from the pixels belonging to the background. Thus, finding an appropriate threshold will solve the localization of the moving object problem. The output of the thresholding operation will be a binary image whose gray level of 0 (black) will indicate a pixel belonging to the background and a gray level of 1 (white) will indicate the object.

Thresholding algorithms can be divided into 6 major groups [15]. These algoritms can be distinguished based on the exploitation of

- Histogram Entropy Information
- Histogram Shape Information
- Image Attribute Information
- Clustering of Gray-Level Information
- Local Characteristics

- Spatial Information

Entropy is an information theoretic measure about probabilistic behaviour of a source. The entropy based methods result in different algorithms which use the entropy of the foreground-background regions or the cross-entropy between the original and binarized image, etc. Assuming that the histogram of an image gives some indication about this probabilistic behaivour, the entropy is tried to be maximized, since the maximization of the entropy of the thresholded image is interpreted as indicative of maximum information transfer [15][16].

Histogram shape based methods analyze the peaks, valleys and curvatures of the image histogram and set the threshold according to these morphological parameters. For example, if the object is clearly distinguishable from the background, the gray level histogram should be bimodal and the threshold can be selected at the bottom point of the valley between two peaks of the bimodal distribution.

Attribute similarity-based methods select the threshold by comparing the original image with its binarized version. The method looks for similarities like edges, curves, number of objects or more complex fuzzy similarities. Iteratively searching for a threshold value that maximizes the matching between the edge map of the gray level and the boundaries of binarized images and penalizing the excess original edges can be a typical example of this approach.

Clustering based algorithms initially divide the gray level data into two segments and apply the analysis afterwards. For example, the gray level distribution is initially modeled as a mixture of two Gaussian distributions representing the background and the foreground and the threshold is refined iteratively such that it maximizes the existence probability of these two Gaussian distributions.

Locally adaptive methods simply determine thresholds for each pixel or a group of pixel, instead of finding a global threshold. The local characteristics of the pixels or pixel groups, such as local mean, variance, surface fitting parameters etc. is used to identify these thresholds.

The spatial methods utilizes the spatial information of the foreground and background pixels, such as context probabilities, correlation functions, co-occurunce probabilities, local linear dependence models of pixels etc.

All these methods are tested on difference images of thermal camera sequences. Experiments showed that, the entropy-based approaches give best results for the tested dataset. Therefore, entropy-based Yen [16] method is examined in detail and implemented. In the Entropy-Yen method, the entropic correlation, TC , is utilized.

$$TC(T) = C_b(T) + C_f(T) = -\log(\sum_{g=0}^{T}\left(\frac{p(g)}{P(T)}\right)^2) - \log(\sum_{g=T+1}^{G}\left(\frac{p(g)}{1-P(T)}\right)^2) \quad (2\text{-}4)$$

The optimal threshold value is determined by maximizing “entropic correlation” equation as

$$T_{opt} = \arg\max_{T}\{TC(T)\} \quad (2\text{-}5)$$

In the equations above, the probability mass function (pmf) of the image is indicated by p(g), g = 0...G, where G is the maximum luminance value in the image, typically 255 if 8-bit quantization is assumed. If the gray value range is not explicitly indicated as [g_{min}, g_{max}], it will be assumed to extend from 0 to G. The cumulative probability function is defined as

$$P(g) = \sum_{i=0}^{g} p(i) \quad (2\text{-}6)$$

It is assumed that the pmf is estimated from the histogram of the image by normalizing to the number of samples at every gray level.

2.2.3 Proposed Background Elimination Method

Background elimination and thresholding are the bases for moving object detection in stationary camera videos [5]. Robust estimation of the background and the threshold value might eliminate most of the noise present in the images, but

still such a method needs noise removal steps for exact localization of the moving object. Simple image processing tools, like size filters and morphological operators [1][8][12], are proposed after background subtraction and thresholding step. Although they are simple filters, they introduce significant improvement in noise removal and accurate localization. The block diagram of the proposed method is given in Figure 2-1.

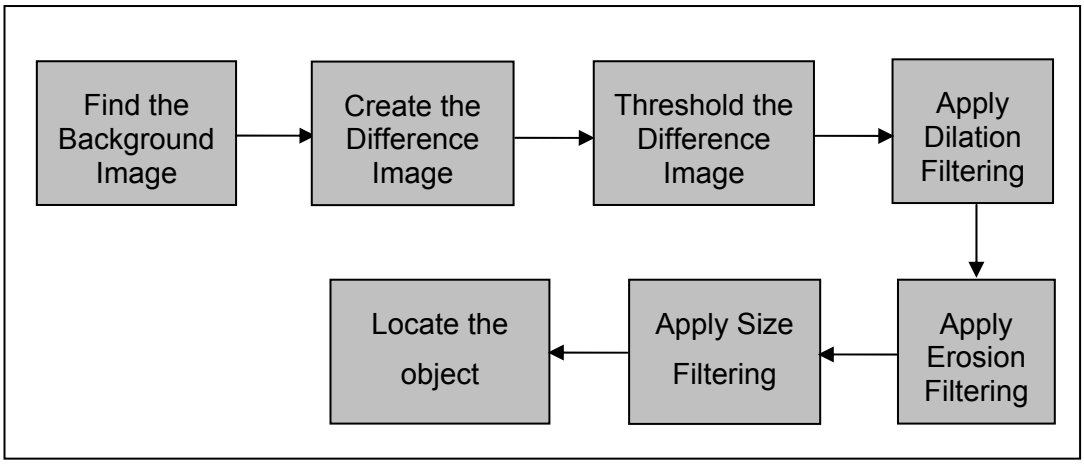


Figure 2-1 Block diagram of IMO detection using background elimination algorithm in stationary camera sequences

Dilation and erosion [1][8] are used consecutively as morphological operators to remove the noise and recover back all the object parts. The pattern of the morphological filters is a 3x3 structure that has a checkerboard pattern as shown in Figure 2-2. This pattern is experimented on thresholded difference images and give satisfactory results.

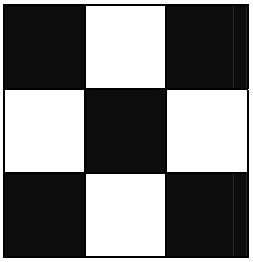


Figure 2-2 The pattern used in dilation and erosion filters

In most cases, there are still some remaining regions after thresholding and morphological operations due to unavoidable noise. Usually, such regions are small and a size filter [1] is used to decrease the false alarm rates due to these noisy regions. In the proposed algorithm, the pixel groups that result in a foreground area that is smaller than 100 pixels are eliminated. The comparative simulation results of this algorithm can be found in Chapter 4.

2.3 Proposed Optical Flow Field-based Method

In this approach, the optical flow field between successive frames is computed and the magnitude field of the motion vectors is thresholded to locate the moving object. A similarity measure filter is added to decrease the false alarm rate.

The algorithm is based on the assumption that only the moving objects will create an optical flow field, since the unexpected changes in the scene illumination which may cause optical flow, is minimized due to the small time difference between successive frames. Moreover, the erroneous flow field caused by the camera noise is eliminated by the optical flow estimation itself.

2.3.1 Optical Flow Field Estimation by Pyramidal Kanade Lucas Tracker

The most common approach for the analysis of visual motion is based on two phases: Computation of an optical flow field and interpretation of this field [9] The term optical flow field refers both to a velocity field composed of vectors

describing the instantaneous velocity of image elements, and a displacement field composed of vectors representing the displacement of image elements from one frame to the next, due to the motion of the camera, the motion of objects in the scene or apparent motion which is a change in the image intensity between frames that mimics object or camera motion. In this section, the computation of the optical flow field and its interpretation to moving object detection in stationary camera sequences are explained.

Optical flow field estimation methods are generally based on the minimization of brightness constraint equation [2][13][18][28].

$$I_x(x,y)*u(x,y)+I_y(x,y)*v(x,y)+I_t(x,y)=0 \qquad (2\text{-}7)$$

where $I_x(x,y)$ is the spatial derivative of the image intensity along x-axis, $I_y(x,y)$ is the spatial derivative of the image intensity along y-axis, $I_t(x,y)$ is the time derivative of the image intensity between consecutive frames, $u(x,y)$ is the horizontal component of the optical flow field and $v(x,y)$ is the vertical component of the optical flow field.

Other constraints, such as smoothness of the neighboring pixels and rigid body motion assumptions [13], are usually introduced to find a solution and also to make the estimation more robust and accurate. Note that the smoothness assumption fails at object boundaries since it is based on the assumption that neighboring pixels will have similar brightness and motion characteristics.

There exist algorithms, which try to match some features such as corners, edges, blocks etc. between two or more frames [1][27] to estimate the optical flow. However, the resulting flow field of such algorithms is not dense enough for detection purposes. Apart from these, there are also some methods making use of fitting affine or perspective parameters to motion vectors in a region and estimating the flow field by taking derivatives [13].

The utilized method in this thesis is called Kanade Lucas Tracker [17][18], and it is based on minimization of the brightness constraint equation within a block assuming that the motion vectors remain unchanged over the whole block. The

goal of tracking is to find the location of in first image at the second one, such that their intensity differences are minimized, i.e. they are similar. Minimization is achieved within a block to overcome the aperture problem. The resulting optical flow vector is the one which minimizes the constraint function, $E(d)$, defined as

$$E(d)=E(\delta x,\delta y,\delta t)=\sum_{x=x_i-w_x}^{x_i+w_x}\sum_{y=y_i-w_y}^{y_i+w_y}(I(x,y,t)-I(x+\delta x,y+\delta y,t+\delta t)) \qquad (2\text{-}8)$$

where $I(x,y,t)$ defines the intensity function, (x_i,y_i) defines the point where optical flow estimation is conducted, (w_x,w_y) defines the neighborhood. In this thesis, the minimization is conducted in 5x5 blocks.

Accuracy and robustness are two important key points of any tracker [18]. While the accuracy component relates to the local subpixel accuracy attached to tracking, the robustness component relates to sensitivity of tracking with respect to changes of lighting, size of image motion, etc. These two components cannot be introduced at the same time, since one needs small blocksize selection, while the other tends to smooth the images by selecting large block sizes. Hence, there is a natural tradeoff between the robustness and acccuracy components, if one uses traditional methods. In order to overcome this tradeoff, a pyramidal and iterative implementation of the Kanade Lucas Tracker [18][28] is introduced. By applying the pyramidal implementation, the robustness component is satisfied by making the calculations in different resolutions, and the iterative implementation solves the accuracy problem.

The pyramidal representation of an image is introduced in a recursive fashion as suggested in [18][28]. The first pyramidal level is the image itself. The second level is computed from the first level by subsampling the first image after applying a lowpass filter to compensate for the anti-aliasing effect [18] ([1/16 1/4 3/8 1/4 1/16] x [1/16 1/4 3/8 1/4 1/16] is used as the low-pass filter in this implementation). For example, for an image, I, of size 320x240, the consecutive pyramidal image levels I^0, I^1, I^2, I^3 are of respective sizes 320x240, 160x120, 80x60 and 40x30.

The entire tracking algorithm can be summarized as follows in form of the following pseudo code [18].

Goal : Let **p**=(x_1,y_1) be a point on the first image I. Find its corresponding location **r**=(x_2,y_2) on the next image J (i.e. J(**r**,t+δt)=I(**p**,t)). I and J are intensity matrices.

Build pyramid representations of I and J : $\{I^L\}_{L=0,,Lm}$ $\{J^L\}_{L=0,,Lm}$

Initialize the pyramidal guess $g^{L_m} = \begin{bmatrix} g_x^{L_m} & g_y^{L_m} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ gLm

for L = Lm **down to** 0 **with step of** -1

Loc. of point p on image I^L: $p^L = \begin{bmatrix} p_x & p_y \end{bmatrix} = p / 2^L$

Derivative of I^L *wrt x*: $I_x(x,y) = \dfrac{I^L(x+1,y) - I^L(x-1,y)}{2}$

Derivative of I^L *wrt y*: $I_y(x,y) = \dfrac{I^L(x,y+1) - I^L(x,y-1)}{2}$

Spatial gradient matrix:

$$G = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y) I_y(x,y) \\ I_x(x,y) I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

Initialization of iterative tracker: $\bar{v}^0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$

for k=1 **to** K **with step of** 1 (or until |n|<threshold)

Image diff: $\delta I_k(x,y) = I^L(x,y) - J^L(x + g_x^L + v_x^{k-1}, y + g_y^L + v_y^{k-1})$

Image mismatch vector: $\bar{b}_k = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} \delta I_k(x,y) I_x(x,y) \\ \delta I_k(x,y) Iy(x,y) \end{bmatrix}$

Optical flow: $\bar{n}^k = G^{-1} \bar{b}_k$

Guess for next iteration: $\bar{v}^k = \bar{v}^{k-1} + \bar{n}^k$

end of for loop on k

Final optical flow at level L: $\bar{d}^L = \bar{v}^K$

Guess for next level L-1: $\bar{g}^{L-1} = \begin{bmatrix} g_x^{L-1} & g_y^{L-1} \end{bmatrix}^T = 2\left(\bar{g}^L + \bar{d}^L\right)$

end of for loop on L

Final optical flow vector : $\bar{d} = \bar{g}^0 + \bar{d}^0$

Location of point on J : $r = p + d$

Solution : The corresponding point is at location **r** on image J

The inner loop calculates the motion vector iteratively, while moving on the trajectory and the outer loop calculates and passes motion vectors between resolutions. Keeping all the computations at sub-pixel accuracy, the intensity values at non-integer locations should be computed. In other words, (x,y) are not necessarily integers. Bilinear interpolation is used to compute the image brightness values for non-integer locations. The formulation is as follows:

$$x = x_0 + \alpha_x \ ; \ y = y_0 + \alpha_y \tag{2-9}$$

$$\begin{aligned} I^L(x,y) = (1-\alpha_x)(1-\alpha_y) I^L(x_0,y_0) + \alpha_x (1-\alpha_y) I^L(x_0+1,y_0) + \\ (1-\alpha_x)\alpha_y I^L(x_0,y_0+1) + \alpha_x \alpha_y I^L(x_0+1,y_0+1) \end{aligned} \tag{2-10}$$

Another issue in optical flow calculation is to decide which pixels deserve further processing and which pixels should be kept out of calculations. This process is called *feature selection* [18]. Once G matrix of a pixel is computed, for further processing it should be invertible. In other words, the minimum eigenvalue of G must be greater than a threshold. This gives us the hint to find the easy-to-track pixels. The feature selection is implemented before the tracking as a pre-processing filter. The feature selection process is as follows

- Compute G matrix and its minimum eigenvalue λ_{min} at every pixel
- Find the maximum of the minimum eigenvalues, λ_{max} over the whole image.
- Retain the image pixels that have λ_{min} value larger than a percentage of λ_{max} .

Actually, this process finds out the smoothest parts of the image where brightness variation is minimum and aperture is unavoidable, thus, prevents the algorithm from diverging.

In this implementation the maximum eigenvalue percentage is selected as 1%. The window size is selected as 5, 3 pyramidal levels are used, and the maximum iteration number of the inner tracking loop is selected as 5.

2.3.2 Detection Algorithm

The optical flow field gives us the motion vector at every pixel. The magnitudes of these vectors depend on the motion characteristics of the moving pixel. If a pixel belongs to an object, the other pixels belonging to the same object will create similar optical flow vectors. Thus, moving objects create optical flow vector clusters of similar motion. The residual optical flow fields must be due to illumination changes, camera or estimation noise, so they should be small in magnitude and have irregularities. As a result, thresholding of the magnitude field and post-processing the thresholded field by size and similarity measure filters will end up with the moving object detected. The block diagram of the algorithm is given in Figure 2-3.

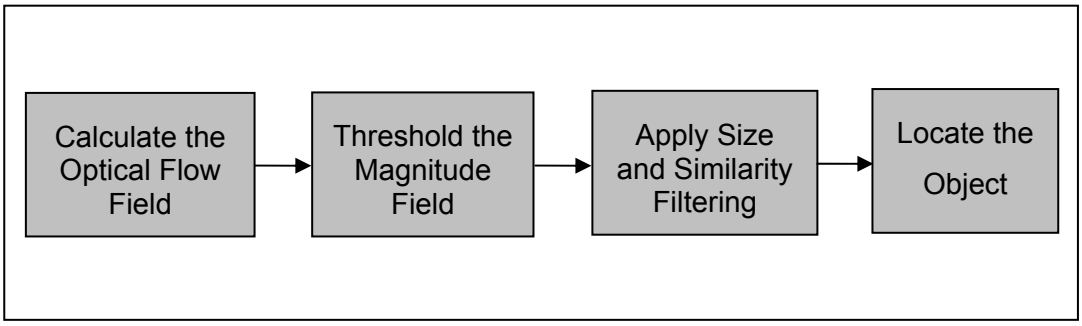


Figure 2-3 Block diagram of IMO detection algorithm using optical flow field in stationary camera sequences

The similarity in a group of vectors can be measured by using statistical properties, such as mean and standard deviation. Let N be the number of vectors in the group. Since the motion vectors have two components, horizontal and vertical, one should have N observations of two variables. The means and standard deviations of two variables over the whole observation space are calculated seperately. For similarity, both standard deviations should be smaller than a threshold which can be related to the mean of the vectors. If one of the components is zero-mean, than either the magnitude of the vectors of that component are close to zero or they are odd distributed. If they are close to zero, their standard deviation will also be close to zero, i.e. they are similar. If they are zero mean but have large standard deviation then the vector component group is said to be non-similar.

For non-zero means, the standard threshold is set to a larger value depending on the mean value. This fact is related to the process noise of the optical flow field for big vectors. The error gets larger when the vector gets bigger.

The following equation defines the similarity measure of a group of optical vectors calculated by KLT method.

$$S = \frac{standard\ \ deviation}{1 + \alpha * \|mean\|} \tag{2-11}$$

This equation states that the maximum similarity is achieved when standard deviation is 0. When this value increases, similarity decreases. When the mean tends to increase, the similarity increases for constant standard deviation, i.e. standard deviation gets a degree of freedom when the mean increases. The rate of increase is biased with +1 and α terms. These values avoid singularities and establish a more convenient change ratio between the standard deviation and the mean. In this implementation, α is set to 0.5. The threshold for similarity, S, is set to 1. Both similarity measures should be smaller than this threshold.

For multiple moving objects, these values should be calculated for each object.

Simulation results for this algorithm are presented in Chapter 4.

CHAPTER 3

MOVING OBJECT DETECTION WITH MOVING CAMERA

3.1 Introduction

When the camera is not stationary, the 2D motion observed in an image sequence is primarily caused by 3D camera motion (the ego-motion) and by 3D motions of independently moving objects. The key step in moving object detection is accounting for the camera-induced image motion. After compensating the camera motion, the remaining motions must be due to the independently moving objects.

The camera induced 2D image motion depends on both the ego-motion parameters (3D rotation and translation) and the depth of each point in the scene. Estimating all of these parameters for compensating the camera motion, is an inherently ambiguous problem and is ill conditioned [6]. When the scene contains large depth variations, the ego-motion parameters may be recovered. However, when the depth variations are not significant, namely, when the scene can be modeled as a 2D plane, the recovery of the ego-motion parameters is not reliable. Thus, direct recovery of egomotion parameters is not an efficient way in moving object detection. Instead, the motion parameters should be integrated implicitly with the object detection algorithm.

In 2D scenes, the image motion can be expressed in terms of a global 2D parametric transformation of a planar surface and the detection problem becomes trivial after this computation. However, this approach is robust and reliable, when

applied to planar scenes or distant scenes or when the camera is making rotations only. Thus, such an approach cannot be applied to 3D scenes, where the depth variation in the scene is significant and/or the camera makes translational motion. In 3D scenes, the detection problem is solved by parallax motion analysis.

Therefore, the moving object problem differs for 2D and 3D scenes, even for the intermediate case, when the depth variation is not significant, but a global parametric transformation is not enough to represent the motion.

3.2 Planar scenes

When the scene viewed from a moving camera is planar (flat) or at such a distance, that it can be approximated by a flat 2D surface or the camera is making rotations, zooms, small translations then the camera-induced motion can be expressed by a global parametric transformation. The 3D motion of a planar surface and its 2D projection shall be examined in detail to parameterize the camera induced image motion.

3.2.1 Modeling 3-D Velocity

Let (X,Y,Z) represent a Cartesian coordinate system which is fixed with respect to the camera and let (x,y) represent a corresponding coordinate of a planar image as shown in Figure 3-1 [23]. The image plane is located at the focal length $Z=f_c$.

The perspective projection of a scene point (X,Y,Z) on the image plane at a point (x,y) is expressed by:

$$x = f_c \frac{X}{Z} \ , \ y = f_c \frac{Y}{Z} \tag{3-1}$$

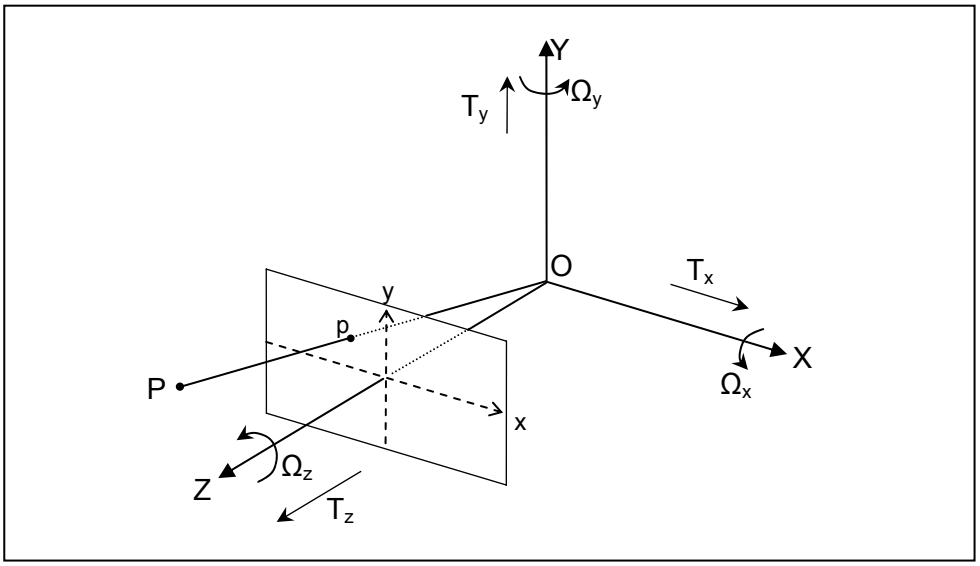


Figure 3-1 Coordinate System

According to classical kinematics the 3D motion of a rigid body can be written as follows [13]:

$$X' = R\,X + T \tag{3-2}$$

where 3x3 matrix R represents the rotation in terms of angular parameters (Ω_x, Ω_y, Ω_z) and 3x1 vector T represents the translation in terms of translational parameters (T_x, T_y, T_z)

In order to find the instantaneous velocity one should find the displacement, as time difference between two instants goes to zero in the limit [13][14]. For small angles, the relation $X' = R\,X + T$ becomes

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & -\Phi & \Psi \\ \Phi & 1 & -\Theta \\ -\Psi & \Phi & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} X'-X \\ Y'-Y \\ Z'-Z \end{bmatrix} = \begin{bmatrix} 0 & -\Phi & \Psi \\ \Phi & 0 & -\Theta \\ -\Psi & \Phi & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{3-3}$$

Dividing both sides by Δt (time difference) and taking limit $\Delta t \to 0$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\Phi} & \dot{\Psi} \\ \dot{\Phi} & 0 & -\dot{\Theta} \\ -\dot{\Psi} & \dot{\Phi} & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \Rightarrow \overline{\dot{X}} = \overline{\Omega} \times \overline{X} + \overline{V} \quad \begin{matrix} \overline{\dot{X}} = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} \end{bmatrix} \\ \overline{\Omega} = \begin{bmatrix} \dot{\Theta} & \dot{\Psi} & \dot{\Phi} \end{bmatrix} \\ \overline{V} = \begin{bmatrix} V_x & V_y & V_z \end{bmatrix} \end{matrix} \quad (3\text{-}4)$$

Note that $\overline{\Omega} = \begin{bmatrix} \dot{\Theta} & \dot{\Psi} & \dot{\Phi} \end{bmatrix} = \begin{bmatrix} \Omega_x & \Omega_y & \Omega_z \end{bmatrix}$

In order to relate the 3D velocity to 2D, projection is used.

$$u = \dot{x} = \frac{dx}{dt} = \frac{d(f\frac{X}{Z})}{dt} = f\frac{Z\dot{X} - X\dot{Z}}{Z^2} = f\frac{\dot{X}}{Z} - x\frac{\dot{Z}}{Z} \quad (3\text{-}5)$$

Similarly,

$$v = \dot{y} = \frac{dy}{dt} = \frac{d(f\frac{Y}{Z})}{dt} = f\frac{Z\dot{Y} - Y\dot{Z}}{Z^2} = f\frac{\dot{Y}}{Z} - x\frac{\dot{Z}}{Z} \quad (3\text{-}6)$$

Substituting the 3D velocity relation in terms of angles, equations become

$$u = -f_c\left(\frac{V_x}{Z} + \Omega_y\right) + x\frac{V_z}{Z} - y\Omega_z + -x^2\frac{\Omega_y}{f_c} + xy\frac{\Omega_x}{f_c} \quad (3\text{-}7)$$

$$v = -f_c\left(\frac{V_y}{Z} + \Omega_x\right) - x\Omega_z + y\frac{V_z}{Z} - xy\frac{\Omega_y}{f_c} + y^2\frac{\Omega_x}{f_c} \quad (3\text{-}8)$$

Assuming (X,Y,Z) lies on a planar surface in the scene, represented by the equation, where (A,B,C) defines the plane:

$$Z = A + BX + CY \quad (3\text{-}9)$$

Dividing both sides of equation 3-9 by A.Z, one may get

$$\frac{1}{Z} = \frac{1}{A} - \frac{B}{A} * \frac{X}{Z} - \frac{C}{A} * \frac{Y}{Z} \quad (3\text{-}10)$$

This can be rewritten as

$$\frac{1}{Z} = \alpha + \beta x + \gamma y \tag{3-11}$$

where: $\alpha = \frac{1}{A}$, $\beta = \frac{-B}{A f_c}$, $\gamma = \frac{-C}{A f_c}$

Hence, the displacement equation becomes

$$u = a + b x + c y + g x^2 + h xy \tag{3-12}$$

$$v = d + e x + f y + g xy + h y^2 \tag{3-13}$$

where

$$a = - f_c \alpha V_x - f_c \Omega_y \tag{3-14}$$

$$b = \alpha V_z - f_c \beta V_x \tag{3-15}$$

$$c = \Omega_z - f_c \gamma V_x \tag{3-16}$$

$$d = - f_c \alpha V_y + f_c \Omega_x \tag{3-17}$$

$$e = - \Omega_z - f_c \beta V_y \tag{3-18}$$

$$f = \alpha V_z - f_c \gamma V_y \tag{3-19}$$

$$g = \frac{-\Omega_y}{f_c} + \beta V_z \tag{3-20}$$

$$h = \frac{\Omega_x}{f_c} + \gamma V_z \tag{3-21}$$

Equations 3-12 and 3-13 describe the 2D motion in the image plane, expressed in terms of eight parameters (a,b,c,d,e,f,g,h) , which correspond to a general 3D motion of a planar surface in the scene, assuming a small field of view and small displacement. When the depth variation of the scene is much smaller

than the average distance of the scene from the camera, the equations 3-12 and 3-13 describe the image motion to sub-pixel accuracy [6].

3.2.2 Detection Algorithm

The 2D image motion vectors u and v can be calculated by making use of any optical flow calculation method. The 'Kanade Lucas Feature Tracker' [18], which is explained in Chapter 2.2, is suitable for calculation of the optical flow vectors for further processing.

Instead of using the 8 parameters, the linear terms a,b,c,d,e,f (6 affine) are used to express the optical flow field and only these 6 affine parameters are estimated for the image motion [19]. The reason for this elimination is to build linear algorithms and discard the computational noise coming from the second order terms g and h. Actually, when we use affine models for analyzing motion, we are proposing that the optical flow in an image can be described as a set of planar patches in velocity space [20].

Given the 'optical flow field', a set of affine motion model parameters can be determined. In this 2D case, there exist only one affine motion model, since all the image motion is assumed to belong to the same 3D planar motion.

The affine parameters are estimated using linear regression techniques [20]. This algorithm can be seen as a plane-fitting algorithm in the velocity space since the affine model is a linear model of motion. The regression is applied separately on each velocity component because the x affine parameters depend only on the x component of velocity and the y parameters depend only on the y component of the velocity.

Let $\begin{bmatrix} a_e & b_e & c_e & d_e & e_e & f_e \end{bmatrix}$ be the hypothesis vector in 6 dimensional affine parameter space with ${u_e}^T = \begin{bmatrix} a_e & b_e & c_e \end{bmatrix}$ and ${v_e}^T = \begin{bmatrix} d_e & e_e & f_e \end{bmatrix}$ corresponding to x and y components, and $\varphi^T = \begin{bmatrix} 1 & x & y \end{bmatrix}$ be the regressor, then the motion field can be written as

$$U_x(x,y) = \varphi^T u_e \tag{3-22}$$

$$U_y(x,y) = \varphi^T v_e \tag{3-23}$$

Finally, a linear least squares estimate for 6 parameter affine model is given as

$$\left[a_e \quad b_e \quad c_e ; d_e \quad e_e \quad f_e\right]^T = \left(\sum_{P_i} \varphi \varphi^T\right)^{-1} \sum_{P_i} \left(\varphi \left[U(x,y) \quad V(x,y)\right]\right) \tag{3-24}$$

Once the affine parameters are estimated, they can be used to warp the second image towards the first one to extract the image regions that does not match to the general image motion, namely the *independently moving objects* (IMO) [6]. All the regions belonging to the static portions of the scene should be aligned perfectly, due to the 2D registration. The detection of moving objects is therefore performed by determining local misalignments after the global 2D parametric registration. In order to make the detection more robust, thresholding, size filtering and morphological operations steps are added to the detection algorithm. The overall process is summarized in Figure 3-2.

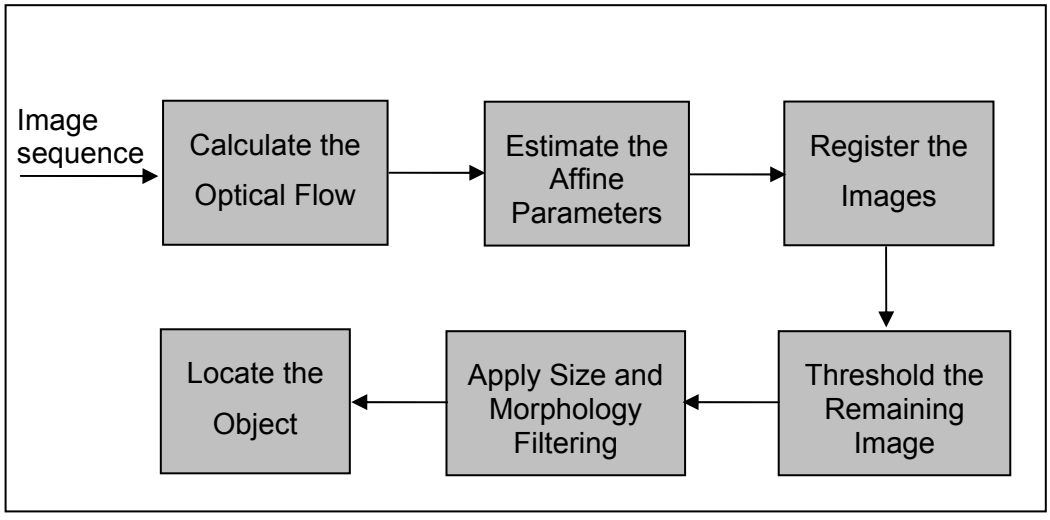


Figure 3-2 The algorithm of IMO detection in 2D scenes with moving camera sequences

3.3 Multi-planar Scenes

When the camera is translating and the scene cannot be modeled as a single plane or a flat surface, it is insufficient to use a single global parametric transformation estimation to describe the image motion [6]. If the depth variation is not significant and the scene can be modeled with a few numbers of planes with different 2D motions, a modified version of the 2D approach can still be used to detect independently moving objects. This approach is based on fitting multiple planar surfaces, multiple 2D layers, to the scene and warping them one after the other [6]. Therefore, key step becomes the segmentation of the optical flow field.

Different from rotational movement, translation of the camera will generate different 2D motion vectors for planes at different depths. Therefore, segmentation of the optical flow field vectors in such a case should end up with the segmentation of the planes at different depths and their corresponding affine motion models.

3.3.1 Optical Flow Field Segmentation

Given the optical flow field, the task of segmentation is to identify the coherent motion regions. When a set of motion models are known, the classification based on motion can directly follow to identify the corresponding regions. However, the corresponding motion models are not known initially. One simple solution is generating a six-dimensional parameter space of the affine transformations where each dimension corresponds to one of the parameters, as suggested in [9]. For computational reasons the parameter space must contain only a finite number of points. Thus, the computation requires a trade off between the computational cost and stability.

The introduced algorithm samples motion data to derive a set of motion hypotheses that are likely to be observed in the image and update these hypotheses in recursive fashion, as suggested in [20]. Thus, the affine motion model estimation and clustering based on motion model estimation are recursively merged. The overall algorithm is given in Figure 3-3 as a block diagram:

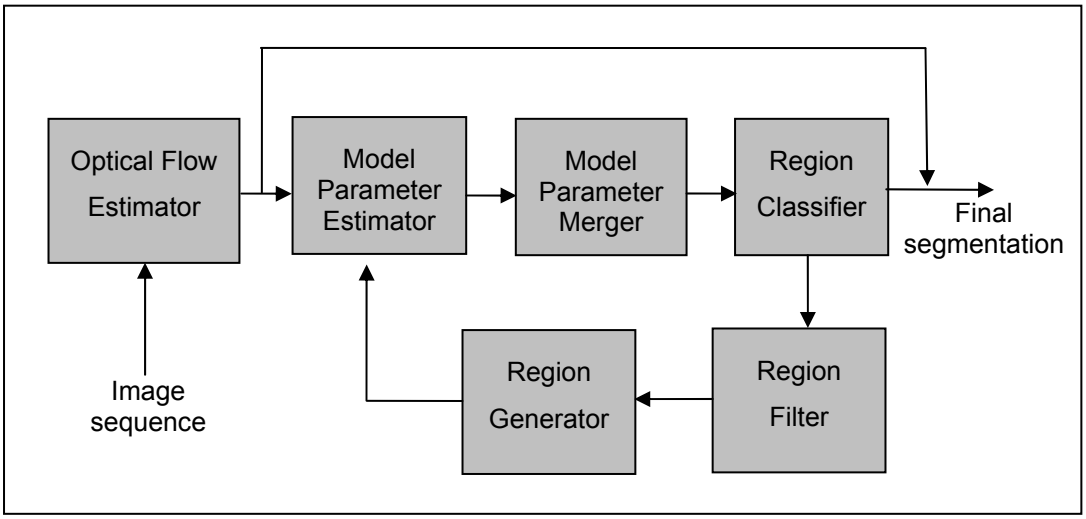


Figure 3-3 Optical Flow Segmentation Algorithm

The region generator initially divides the image into an array of non-overlapping square regions. The size of the initial squares is kept to a minimum (30x30) for localizing the estimation and avoiding estimation of motion across object boundaries. The model parameter estimator estimates the affine motion model parameters within the region using the standard linear regression technique, as discussed in Section 3.2.2 . The initial model estimations will be incorrect regardless of the initial region sizes, since the regions may contain object boundaries. Thus, a reliability measure is used to identify the erroneous estimations [20]. Variance of the model estimation residual error, σ_i^2, can be calculated as follows:

$$\sigma_i^2 = \frac{1}{N_i} \sum (V(x,y) - V_{a_i}(x,y))^2 \qquad (3\text{-}25)$$

where N_i is the number of pixels in the region *i*, $V(x,y)$ is the original optical flow field vector and $V_{a_i}(x,y)$ is the estimated motion vector. The motion models that have a residual greater than a prescribed threshold are eliminated, since they do not provide a good model for the region.

Motion models of the regions belonging to the same layer should have similar parameters. These regions are grouped into the affine parameter space with a *K-means clustering algorithm* as suggested in [20][30]. A scaled distance measure is introduced in order to calculate the separation between the motion models.

$$D_m(a_1, a_2) = (a_1 - a_2)^T \quad M \quad (a_1 - a_2) \tag{3-26}$$

$$M = diag(1 \quad r^2 \quad r^2 \quad 1 \quad r^2 \quad r^2) \tag{3-27}$$

where, r is roughly the dimension of the image.

In K-means algorithm, an initial set of cluster centers, which are separated by a prescribed distance, are selected. Each of the affine hypotheses is assigned to the nearest center. Following the assignment, the centers are updated by calculating the new affine model of the merged region. Iteratively, the centers are updated until the cluster membership is reached or equivalently the cluster centers are unchanged. In these iterations, when the distance between two clusters is smaller than a prescribed distance, they are merged into a single cluster. After K-means algorithm, one should be left with a smaller number of hypotheses describing the optical flow field, which are more precise than the initial parameters.

After representing the optical flow field by a number of affine hypotheses, the region assignment with hypotheses testing stage begins. In this stage, every possible motion vector, calculated using affine hypotheses, is compared to its original value and the best match is assigned to that pixel location. A distortion function is used to derive the hypotheses testing mechanism. The distortion function is defined as:

$$G(i(x,y)) = \sum_{x,y} (V(x,y) - V_{ai}(x,y))^2 \tag{3-28}$$

where $i(x,y)$ indicates the model assigned to (x,y), $V(x,y)$ indicates the motion vector estimated from KLT algorithm and $V_{ai}(x,y)$ indicates motion vector estimated using i^{th} motion hypotheses. It is obvious that the distortion function reaches a minimum, if the motion hypotheses exactly describe the motion in the

image region. Since it is not possible to find an exact description, one should rather try to minimize the distortion function. The minimization is achieved by assigning an hypotheses to each pixel location by minimizing the distortion at that location.

$$i_0(x,y) = \arg\min\left[V(x,y) - V_{ai}(x,y)\right]^2 \tag{3-29}$$

where, $i_0(x,y)$ is the minimum distortion assignment. The assignment is not made at pixel locations, where the error between the estimated and original motion vectors is greater than a prescribed threshold.

After hypotheses testing at every pixel, the resulting motion model parameters of clusters are estimated and the iterative algorithm continues to make hypotheses testing over these new models until a cluster membership or a prescribed iteration number is reached. The resulting clusters are removed from the iteration algorithm, if they have residuals larger than a prescribed threshold.

It is important to note that the iterative analysis is necessary only for the first pair of an image sequence, since motion and shape parameters change slowly. Thus, the clusters of the motion field of the previous pair can be used as an initial step for segmentation. In this way, most of the computational complexity is concentrated in the initial segmentation step, which is required only once per sequence [20].

3.3.2 Detection Algorithm in Multilayer

After clustering the optical flow field and computing the affine parameters for each cluster, the remaining task is warping the clusters sequentially. The first warping is achieved by using motion parameters of the largest cluster, since it will align a large portion of the image. After warping the first plane, the misaligned regions are detected and segmented out for further warping. The detection of the misaligned regions are simply determined by using a thresholding algorithm. The warping continues with the smaller clusters in a decreasing size fashion. The clusters, which have sizes smaller than a prescribed threshold are kept out of computation, since their motion parameters will not be reliable and the probability of being an independently moving object for a small sized cluster is high.

It is obvious that if an independently moving object is large enough, it might be recognized as a layer and will be a part of the warping process [6]. Thus, it will be impossible to detect moving objects. The following cues can be used to distinguish between moving objects and static scene layers:

- Moving objects produce discontinuities in 2D motion everywhere on their boundary, as opposed to static 2D layers. Therefore, if a moving object is detected as a layer, it can be distinguished from real scene layers due to the fact that it should appear as if floating in the air.

- 3D consistency over time of two 2D layers can be examined. If two layers belong to the same static background, their parallax motion will have a consistency over time opposed to independently moving objects. This will be discussed in detail in Section 3.3 .

The thresholding, size filtering and morphological operations steps are similarly added to the end of the algorithm for exact localization of the moving objects. The moving object detection algorithm in multi-planar scenes is summarized in Figure 3-4 as a block diagram,

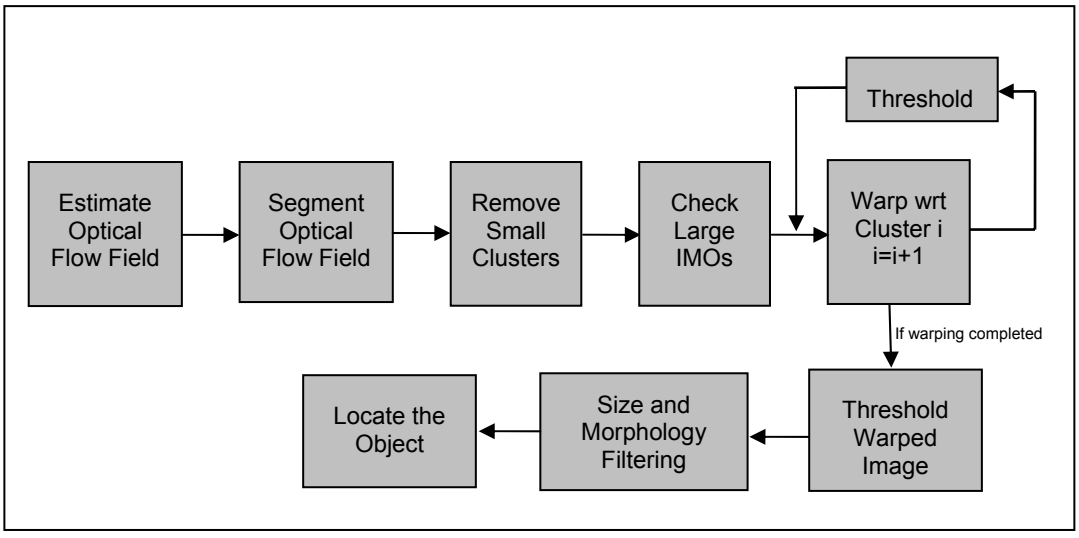


Figure 3-4 IMO detection algorithm in multilayer scenes with moving camera sequences

3.4 Scenes with General 3D Parallax

The differences in 2D motion vectors of projected scene points due to the depth variations of the layers when the camera makes considerable translational motion are called 3D *parallax* [6][14]. Single and multi-layered approaches are adequate to handle large number of situations, where the 3D parallax is not significant. However, they cannot model the parallax in terms of layers. These more complex 3D scenes should be examined using the *plane-parallax decomposition* approach, as suggested in [6].

The starting point of 3D parallax extraction begins with a previously developed approach, 2D parametric registration of a layer. It can be shown that the 2D parametric registration of a layer, removes all the effects of camera rotation, zoom and calibration without explicitly computing them [6][21]. Since the planar motion caused by rotation or zoom does not depend on plane depth (see equations 3-7 and 3-8), all the planes at different depths will have same 2D motions coming from these terms. Therefore registering one plane means registering all other planes in terms of 2D motion vectors due to rotation and zoom of the camera. After registration of one plane, the residual motion is due only to the translational motion of the camera and to the deviations of the scene structure from the planar surface [24]. This fact can easily be seen when the parametric equation of the motion field is examined in detail. Remember the eight affine parameters related to the motion of a plane in equations 3-14 to 3-21. The difference between the 2D projective transformations of two different planes can be parametrically written as

$$a_1 - a_2 = -f_c(\alpha_1 - \alpha_2)V_x \tag{3-30}$$

$$b_1 - b_2 = (\alpha_1 - \alpha_2)V_z - f_c(\beta_1 - \beta_2)V_x \tag{3-31}$$

$$c_1 - c_2 = -f_c(\gamma_1 - \gamma_2)V_x \tag{3-32}$$

$$d_1 - d_2 = -f_c(\alpha_1 - \alpha_2)V_y \tag{3-33}$$

$$e_1 - e_2 = f_c(\beta_1 - \beta_2)V_y \tag{3-34}$$

$$f_1 - f_2 = (\alpha_1 - \alpha_2) V_z - f_c (\gamma_1 - \gamma_2) V_y \quad (3\text{-}35)$$

$$g_1 - g_2 = (\beta_1 - \beta_2) V_z \quad (3\text{-}36)$$

$$h_1 - h_2 = (\gamma_1 - \gamma_2) V_z \quad (3\text{-}37)$$

where as mentioned earlier $\lfloor V_x \quad V_y \quad V_z \rfloor$ are the camera translation parameters α, β, γ are the plane parameters and f_c is the camera focal length. By taking difference, the camera rotation parameters $(\Omega_x, \Omega_y, \Omega_z)$ were removed from the equation, leaving only with the translational parameters and the plane parameters that parametrically depend on the depth of the planes.

The residual flow field after parametric registration of a plane is a radial field centered at Focus of Expansion (FOE) [6][24]. In other words, the motion field is composed of vectors radiating from a common origin, which is also called the epipole.

This observation has led to the so-called “plane+parallax” approach to 3D scene analysis. Figure 3-5 provides a geometric interpretation of the planar parallax and the residual epipolar field.

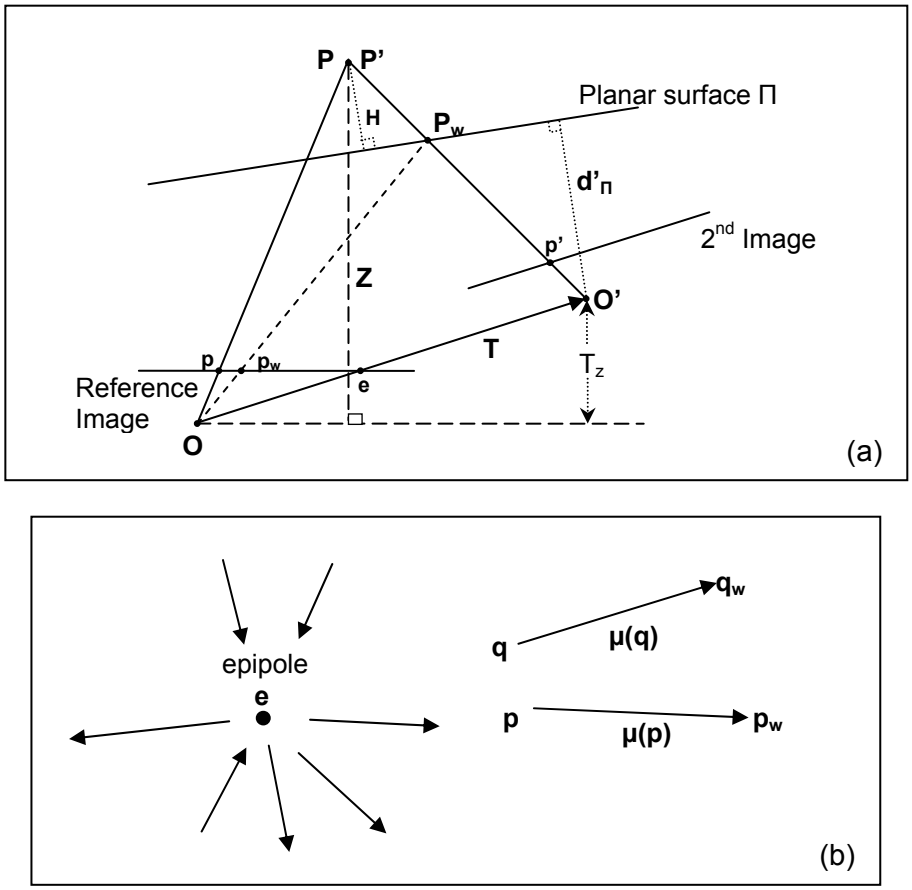


Figure 3-5 **The plane+parallax decomposition**
(a) The geometric interpretation
(b) The epipolar field of the residual parallax displacement

The 2D displacement of point P can be written as

$$\bar{u} = (\bar{p}' - \bar{p}) = \bar{u}_{\Pi} + \bar{\mu} \tag{3-38}$$

$$\bar{u}_{\Pi} = \bar{p}' - \bar{p}_w \quad , \quad \bar{\mu} = \begin{cases} \gamma \dfrac{T_z}{d'_{\pi}} (\bar{e} - \bar{p}_w) & 'if \;\; T_z \neq 0' \\ \dfrac{\gamma}{d'_{\pi}} \bar{t} & 'if \;\; T_z = 0' \end{cases} \tag{3-39}$$

where $\bar{u}_{\Pi}$ denotes the planar part of the motion (homography due to Π) and $\bar{\mu}$ denotes the residual planar parallax 2D motion. $\bar{p}_w$ is an image point in the

first frame that results from warping the corresponding point $\bar{p}'$ in the second image by the 2D quadratic transformation of plane Π. $\bar{e}$ denotes the epipole and $d_{\pi}^{'}$ denotes the distance of the second camera center from the plane. γ is a measure of 3D shape of point P and is the ratio of perpendicular distance of point P to the planar surface Π and the depth with respect to the first camera (γ = H/Z) as shown in Figure 3-5(a). γ is referred as the *projective 3D structure* [6][24] of point P. $\bar{t}$ is the translation vector $(T_x, T_y)^T$. Note that when camera translation along Z-axis is zero, then the epipole point is at infinity.

Derivation of the Plane+Parallax Decomposition:

Let $P = (X, Y, Z)$ and $P' = (X', Y', Z')$ denote the Cartesian coordinates of a scene point with respect to two different camera views, respectively. An arbitrary 3D rigid coordinate transformation between $\overline{P}$ and $\overline{P'}$ is expressed by:

$$\overline{P'} = R\overline{P} + \overline{T'} \tag{3-40}$$

where R represents the rotation between the two camera coordinate systems, $\overline{T'} = (T_x', T_y', T_z')$ denotes the 3D translation between the two views as expressed in the coordinate system of the second camera, and $\overline{T} = -R^{-1}\overline{T'}$ denotes the same quantity in the coordinate system of the of the first camera. Let Π denote an arbitrary 3D planar real or virtual planar surface. Let $\overline{N}$ denote its normal as expressed in the coordinate system of the first camera, and $\overline{N}'$ denote the same quantity in the coordinate system of the second camera. Any point $\overline{P} \in \Pi$ satisfies the equation $\overline{N}^T\overline{P} = d_{\pi}$ (and similarly $\overline{N}'^T\overline{P}' = d_{\pi}'$). For a general scene point $\overline{P}$:

$$\overline{N}^T\overline{P} = d_{\pi} + H \tag{3-41}$$

$$\overline{N}'^T\overline{P}' = d_{\pi}' + H \tag{3-42}$$

where H denotes the perpendicular distance of $\overline{P}$ from the plane Π. Note that H is invariant with respect to the camera coordinate systems (see Figure 3-5(a)). By inverting equation 3-40*, one can obtain*

$$\overline{P} = R^{-1}\overline{P'} - R^{-1}\overline{T'} = R^{-1}\overline{P'} + \overline{T} \qquad (3\text{-}43)$$

From equation 3-42, one can derive

$$\frac{\overline{N'}^{T}\overline{P'} - H}{d'_{\pi}} = 1 \qquad (3\text{-}44)$$

Substituting this in equation 3-43, one can obtain

$$\overline{P} = R^{-1}\overline{P'} + \overline{T}\,\frac{\overline{N'}^{T}\overline{P'} - H}{d'_{\pi}} = \left(R^{-1} + \frac{\overline{T}\,\overline{N'}^{T}}{d'_{\pi}}\right)\overline{P'} - \frac{H}{d'_{\pi}}\overline{T} \qquad (3\text{-}45)$$

Let $\overline{p} = (x, y, 1)^{T} = \frac{1}{Z}K\overline{P}$ *and* $\overline{p'} = (x', y', 1)^{T} = \frac{1}{Z'}K'\overline{P'}$ *denote the images of the point* $\overline{P}$ *in the two camera views as expressed in homogeneous coordinates.* K *and* K' *are 3x3 matrices representing the internal calibration parameters of the two cameras. In general, K has the following form [6]:*

$$K = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix} \qquad (3\text{-}46)$$

Moreover, define $\bar{t} = (t_x, t_y, t_z)^{T} = K\overline{T}$ *. (Note that* $(K\overline{P})_z = Z$ *,* $(K'\overline{P'})_z = Z'$ *, and* $t_z = T_Z$*). Multiplying both sides of equation 3-45 by* $\frac{1}{Z'}K$ *gives:*

$$\frac{Z}{Z'}\overline{p} = K\left(R^{-1}\frac{\overline{TN'}^{T}}{d'_{\pi}}\right)K'^{-1}\overline{p'} - \frac{H}{d'_{\pi}Z'}\bar{t} \qquad (3\text{-}47)$$

Hence,

$$\overline{p} \cong A'\overline{p'} - \frac{H}{d'_{\pi}Z'}\bar{t} \qquad (3\text{-}48)$$

where $\cong$ *denotes the equality up to an arbitrary scale.* $A' = K\left(R^{-1} \frac{\overline{TN}'^{T}}{d'_{\pi}} \right) K'^{-1}$ *is a 3x3 matrix which represents the coordinate transformation of the planar surface* Π *between the two camera views, i.e. the homography between the two views due to plane* Π*. Scaling both sides by the third component (i.e. projection) gives the equality*

$$\begin{aligned} \bar{p} &= \frac{A'\bar{p}' - \frac{H}{d'_{\pi}Z'}\bar{t}}{a'_3\bar{p}' - \frac{HT_Z}{d'_{\pi}Z'}} = \frac{A'\bar{p}'}{a'_3\bar{p}'} - \frac{A'\bar{p}'}{a'_3\bar{p}'} + \frac{A'\bar{p}' - \frac{H}{d'_{\pi}Z'}\bar{t}}{a'_3\bar{p}' - \frac{HT_Z}{d'_{\pi}Z'}} \\ &= \frac{A'\bar{p}'}{a'_3\bar{p}'} + \frac{\frac{HT_Z}{d'_{\pi}Z'}}{a'_3\bar{p}' - \frac{HT_Z}{d'_{\pi}Z'}} \frac{A'\bar{p}'}{a'_3\bar{p}'} - \frac{\frac{H}{d'_{\pi}Z'}}{a'_3\bar{p}' - \frac{HT_Z}{d'_{\pi}Z'}}\bar{t} \end{aligned} \qquad \text{(3-49)}$$

where a'_3 *denotes the third row of the matrix* A'*. Moreover, by considering the third component of the vector in equation 3-47, one can obtain*

$$\frac{Z}{Z'} = a'_3\bar{p}' - \frac{HT_Z}{d'_{\pi}Z'} \qquad \text{(3-50)}$$

substituting this into equation 3-49, one can obtain

$$\bar{p} = \frac{A'\bar{p}'}{a'_3\bar{p}'} + \frac{H}{Z}\frac{T_Z}{d'_{\pi}}\frac{A'\bar{p}'}{a'_3\bar{p}'} - \frac{H}{Zd'_{\pi}}\bar{t} \qquad \text{(3-51)}$$

When $T_Z \neq 0$*, let* $\bar{e} = \frac{1}{T_Z}\bar{t}$ *denote the epipole in the first image. Then,*

$$\bar{p} = \frac{A'\bar{p}'}{a'_3\bar{p}'} + \frac{H}{Z}\frac{T_Z}{d'_{\pi}}\left(\frac{A'\bar{p}'}{a'_3\bar{p}'} - \bar{e} \right) \qquad \text{(3-52)}$$

On the other hand, when $T_Z = 0$*, one can obtain*

$$\bar{p} = \frac{A'\bar{p}'}{a_3'\bar{p}'} - \frac{H}{Zd_\pi'}\bar{t} \qquad (3\text{-}53)$$

The point denoted by the vector $\frac{A'\bar{p}'}{a_3'\bar{p}'}$ *is of special interest, since it represents the location to which the point* $\bar{p}'$ *is transformed due to* A'*. In Figure 3-5(a), this is denoted as point* $\bar{p}_w$*. In addition, one can define* $\gamma = \frac{H}{Z}$*, which is the 3D projective structure of* $\bar{P}$ *with respect to the planar surface* Π*. Substituting these into equations 3-52 and 3-53 yields, when* $T_Z \neq 0$

$$\bar{p} = \bar{p}_w + \gamma \frac{T_Z}{d_\pi'}\left(\bar{p}_w - \bar{e}\right) \qquad (3\text{-}54)$$

and when $T_Z = 0$

$$\bar{p} = \bar{p}_w + \frac{\gamma}{d_\pi'}\bar{t} \qquad (3\text{-}55)$$

Rewriting equation 3-54 in the form of image displacements yields (in homogeneous coordinates)

$$\bar{p}' - \bar{p} = \left(\bar{p}' - \bar{p}_w\right) - \gamma \frac{T_Z}{d_\pi'}\left(\bar{p}_w - \bar{e}\right) \qquad (3\text{-}56)$$

Define $\bar{u} = \bar{p}' - \bar{p} = (u, v, 0)^T$*, where* $(u, v)^T$ *is the measurable 2D image displacement vector of the image point* $\bar{p}$ *between the two frames. Similarly, define* $\bar{u}_\pi = \left(\bar{p}' - \bar{p}_w\right) = \left(u_\pi, v_\pi, 0\right)^T$ *and* $\bar{\mu} = -\gamma \frac{T_Z}{d_\pi'}\left(\bar{p}_w - \bar{e}\right) = \left(\mu_x, \mu_y, 0\right)^T$*. Hence,*

$$\bar{u} = \bar{u}_\pi + \bar{\mu} \qquad (3\text{-}57)$$

$\bar{u}_\pi$ denotes the planar part of the 2D image displacement (i.e., the homography due to Π), and $\bar{\mu}$ denotes the residual parallax 2D displacement. Note that when $T_Z = 0$, then from equation 3-55, $\bar{\mu} = -\frac{\gamma}{d'_\pi}\bar{t}$.

The parallax equation proves the existence of an epipole where all the residual motion vectors expand from or contract to. Therefore, if the epipole is detected, independently moving object detection problem will be identifying the motion vectors that are violating the above-told expansion or contraction rule. It is obvious that the performance of this method critically depends on robust recovery of the epipole. However, generally it is almost impossible to estimate the epipole location accurately due to the biasing of the independently moving objects' motion vectors. Especially if the moving object is dominant, (either in magnitude or in number) the biasing will be significant and the estimation might fail.

Another approach, which does not estimate and use the epipole explicitly, whereas directly compare the parallax motion, is suggested to solve the problem This approach is useful especially when 3D parallax is sparse relative to independent motion information, i.e. independently moving object is big [6].

This approach is based on two observations on the parallax motion field [6].

Observation 1: The parallax-based shape constraint

Let $\bar{\mu}_1$ and $\bar{\mu}_2$ be two motion vectors that are belonging to two points at the static background of the scene. Their *relative 3D projective structure* $\frac{\gamma_2}{\gamma_1}$ is given by:

$$\frac{\gamma_2}{\gamma_1} = \frac{\bar{\mu}_2^T (\Delta\bar{p}_w)_\perp}{\bar{\mu}_1^T (\Delta\bar{p}_w)_\perp} \tag{3-58}$$

where, $\Delta\bar{p}_w = \bar{p}_{w2} - \bar{p}_{w1}$ is the vector connecting the warped locations of the corresponding second frame points as shown in Figure 3-6.and $\bar{v}_\perp$ signifies a vector perpendicular to $\bar{v}$.

Figure 3-6 geometrically illustrates the relative structure constraint

$$\frac{\gamma_2}{\gamma_1} = \frac{\overline{\mu}_2^T (\Delta\overline{p}_w)_\perp}{\overline{\mu}_1^T (\Delta\overline{p}_w)_\perp} = \frac{AB}{AC} \tag{3-59}$$

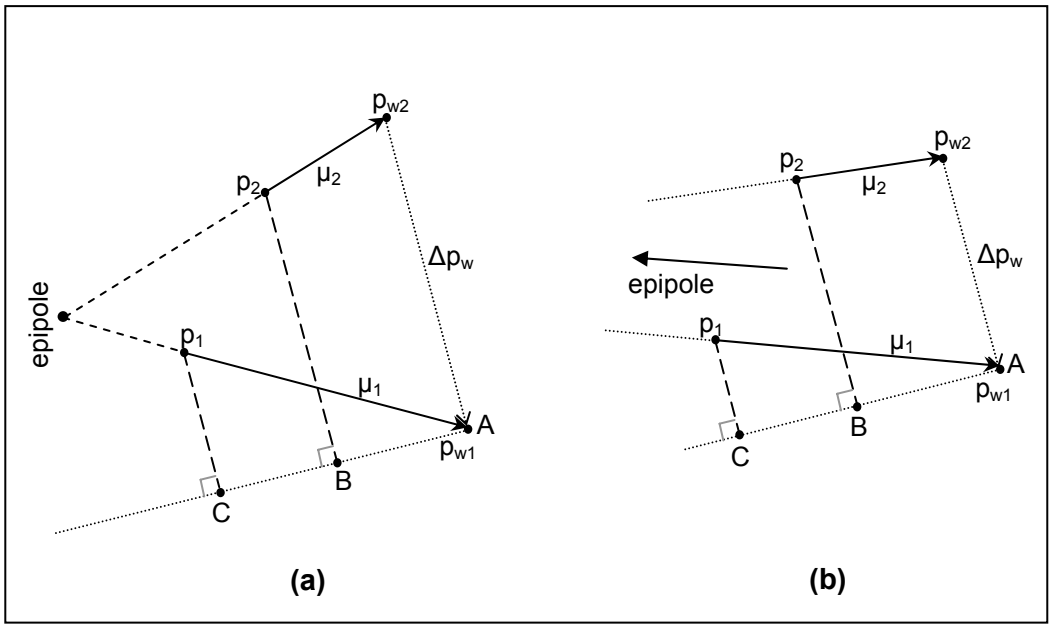


Figure 3-6 The pairwise parallax based shape constraint.
(a) When the epipole recovery is reliable
(b) When the epipole recovery is unreliable

Note that this constraint directly relates the relative projective structure of two points to their parallax displacements alone without computing the camera parameters and the epipole. In Figure 3-6 b, the parallax vectors are nearly parallel and epipole recovery is not reliable. However, the relative structure $\frac{AB}{AC}$ can be reliably computed even in this case.

Proof:

From equation 3-39 *one can derive,*

$$\overline{\mu}_1 = \gamma_1 \frac{T_z}{d_\pi^{'}}(\overline{e} - \overline{p}_{w1}) \quad ; \quad \overline{\mu}_2 = \gamma_2 \frac{T_z}{d_\pi^{'}}(\overline{e} - \overline{p}_{w2}) \tag{3-60}$$

Therefore,

$$\overline{\mu}_1\gamma_2 - \overline{\mu}_2\gamma_1 = \gamma_1\gamma_2 \frac{T_z}{d_\pi^{'}}(\overline{p}_{w2} - \overline{p}_{w1}) \tag{3-61}$$

This last step eliminated the epipole $\overline{e}$*. Equation* 3-61 *implies that the vectors on both sides of the equation are parallel. Since* $\gamma_1\gamma_2 \frac{T_z}{d_\pi^{'}}$ *is a scalar, one gets* $(\overline{\mu}_1\gamma_2 - \overline{\mu}_2\gamma_1) // \Delta\overline{p}_w$*. This leads to the pair wise parallax constraint,*

$$(\overline{\mu}_1\gamma_2 - \overline{\mu}_2\gamma_1)^T (\Delta\overline{p}_w)_\perp = 0 \tag{3-62}$$

When $T_z = 0$, a constraint stronger than equation 3-62 can be derived $(\overline{\mu}_1\gamma_2 - \overline{\mu}_2\gamma_1)^T = 0$. However, equation 3-61 still holds. This is important, since we do not have a priori knowledge of T_z to distinguish between two cases. From equation 3-62, one can easily derive, $\frac{\gamma_2}{\gamma_1} = \frac{\overline{\mu}_2^T(\Delta\overline{p}_w)_\perp}{\overline{\mu}_1^T(\Delta\overline{p}_w)_\perp}$.

Observation 2: The parallax based rigidity constraint

Given the planar-parallax displacement vectors of two points that belong to the background static scene over three frames, the following constraint must be satisfied:

$$\frac{\overline{\mu}_2{}^{jT}(\Delta\overline{p}_w)^j{}_\perp}{\overline{\mu}_1{}^{jT}(\Delta\overline{p}_w)^j{}_\perp} - \frac{\overline{\mu}_2{}^{kT}(\Delta\overline{p}_w)^k{}_\perp}{\overline{\mu}_1{}^{kT}(\Delta\overline{p}_w)^k{}_\perp} = 0 \tag{3-63}$$

where $\overline{\mu}_1{}^j$, $\overline{\mu}_2{}^j$ are the parallax displacement vectors of the two points between the reference frame and j[th] frame, $\overline{\mu}_1{}^k$, $\overline{\mu}_2{}^k$ are the parallax vectors between the

reference frame and k[th] frame, and $(\Delta\overline{p}_w)^j$, $(\Delta\overline{p}_w)^k$ are the corresponding distances between the warped points as in equations 3-62 and 3-63.

Proof:

The key point that enables us to make this observation is the invariance of the projective structure against camera motion. Obviously, the distance H&Z, which define γ, are independent of any frame other than the reference frame.

$$\frac{\gamma_1}{\gamma_2} = \frac{\mu_2^{j^T}(\Delta\overline{p}_w)^{j}{}_{\perp}}{\mu_1^{j^T}(\Delta\overline{p}_w)^{j}{}_{\perp}} = \frac{\mu_2^{k^T}(\Delta\overline{p}_w)^{k}{}_{\perp}}{\mu_1^{k^T}(\Delta\overline{p}_w)^{k}{}_{\perp}} \tag{3-64}$$

Therefore, given two points belonging to the background scene, the points might belong to different layers at different depths. If one takes one of the points as reference and compute the projective structure of the other with respect to the reference, the projective structure will remain unchanged over time and under any camera motion.

These observations led us to extend the 'analysis of planar parallax' to moving object detection, since moving objects will create inconsistencies in projective structures. The advantage of the analysis relies on the fact that, for detection, there is no need to either calculate the camera geometry, camera motion or structure parameters. In addition, the analysis does not rely on parallax information on other image points. A consistency measure is defined as the left-hand side of equation 3-64. The 3D-consistency degrades as this value gets higher.

Figure 3.7(a) graphically displays an example of a configuration, in which estimation of the epipole in presence of biasing of independently moving objects is quite erroneous. Therefore, relying on the epipole computation to detect inconsistencies in 3D motion fails, failing the moving object detection [6].

Instead of using erroneous epipole estimation, the parallax rigidity constraint can be used to detect independently moving object in the same scenario

as shown in Figure 3-7(b). In Figure 3-7, camera is translating to the right. The only static object with pure parallax motion is the tree, while the ball is falling independently.

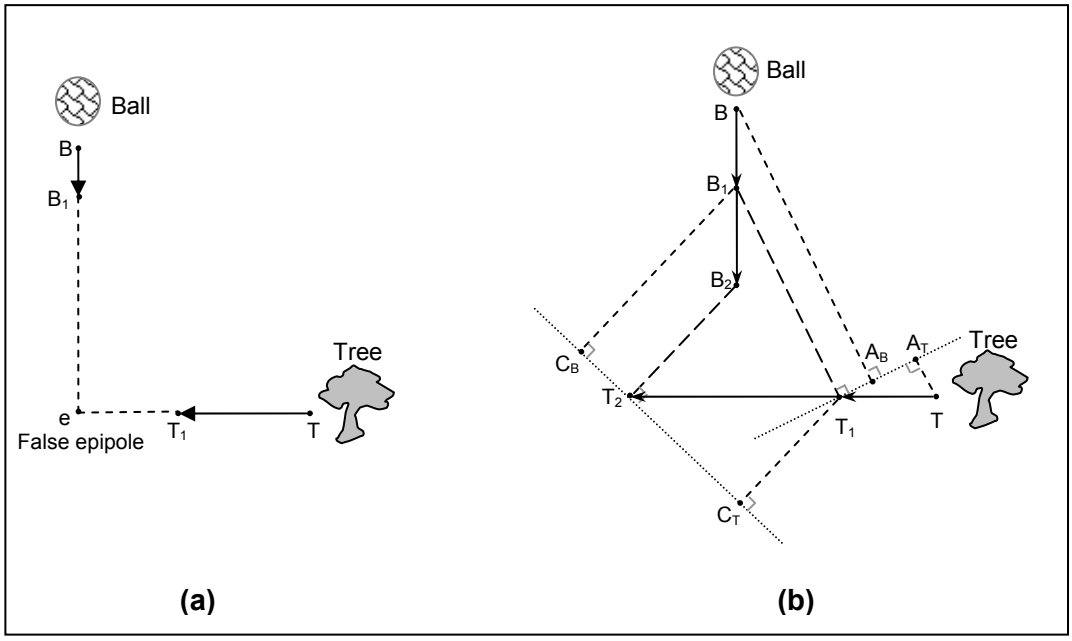


Figure 3-7 **Reliable detection of 3D motion inconsistency with sparse parallax information**
(a) A scenario where the epipole recovery is not reliable
(b) The geometrical interpretation of the rigidity constraint applied to this scenario

The epipole may be incorrectly computed as *e* as shown in Figure 3-7(a) The false epipole seems to be consistent with both motions. However, the rigidity constraint applied to this scenario detects 3D inconsistency over three frames, since $\frac{T_1A_B}{T_1A_T} \neq \frac{T_2C_B}{T_2C_T}$. In this particular case, even the signs do not match.

If the rigidity constraint is applied to all image points, a useful mechanism will be obtained for clustering the parallax vectors (i.e., the residual motion after planar registration) into consistent groups belonging to consistently 3D moving objects, even in the case of erroneous epipoles [6].

In order to apply the parallax rigidity constraint to an image sequence, 2D parametric registration of a plane and selection of a reference point belonging to the background steps should be conducted first. However, these steps are achieved manually in [6].

The largest portion of the clustered optical flow field can be selected as the registration plane since there exist such a plane, in most outdoor scenes.

The reference point is selected as the most intense corner of the image since the optical flow computation will be most accurate at this point. Harris corner detector [25] is selected in this implementation. Harris Corner Detector makes use of the following fact:

- If a windowed image patch is flat, then all the shifts along different axes will result in small changes in terms of intensity differences,
- If the window straddles an edge, then a shift along the edge will result in a small change, but a shift perpendicular to the edge will result in a large change,
- If the windowed patch is a corner or isolated point, then all shifts will result in a large change [25].

The covariance matrix of the image differences at a window, w, is calculated for corner detection. The covariance matrix is given as:

$$C = \begin{bmatrix} \sum_w I_x I_x & \sum_w I_x I_y \\ \sum_w I_y I_x & \sum_w I_y I_y \end{bmatrix} \tag{3-65}$$

The eigenvalues of this matrix, λ_1 and λ_2, give valuable information about the intensity characteristics within the window. If a window contains a corner, the minimum eigenvalue of this matrix should be much greater than 0. Therefore, one should try to find region centers whose eigenvalues are both much greater than zero by the help of the following function

$$\Re = \det(C) - k * trace^2(C) \tag{3-66}$$

where $\det(C) = \lambda_1 \lambda_2$ and $trace(C) = \lambda_1 + \lambda_2$ and k is usually chosen as 0.04 [25]. The reference point is selected as the image point that is at the center of the window which maximizes function $\Re$.

After 2D parametric registration of the largest plane, the parallax rigidity constraint is applied to the whole image by making use of the reference point. The parallax rigidity constraint will extract the image points that belong to independently moving objects. Obviously, this computation should carry some noise due to noise coming from optical flow field estimation and 2D parametric motion estimation of the registration plane. Therefore, the resulting image is not expected to be a perfect binary image, showing the static background and independently moving objects. Thresholding, size and morphological filtering steps, which were previously developed, are added at the end for exact localization of the independently moving objects. The algorithm is summarized in Figure 3-8 as a block diagram.

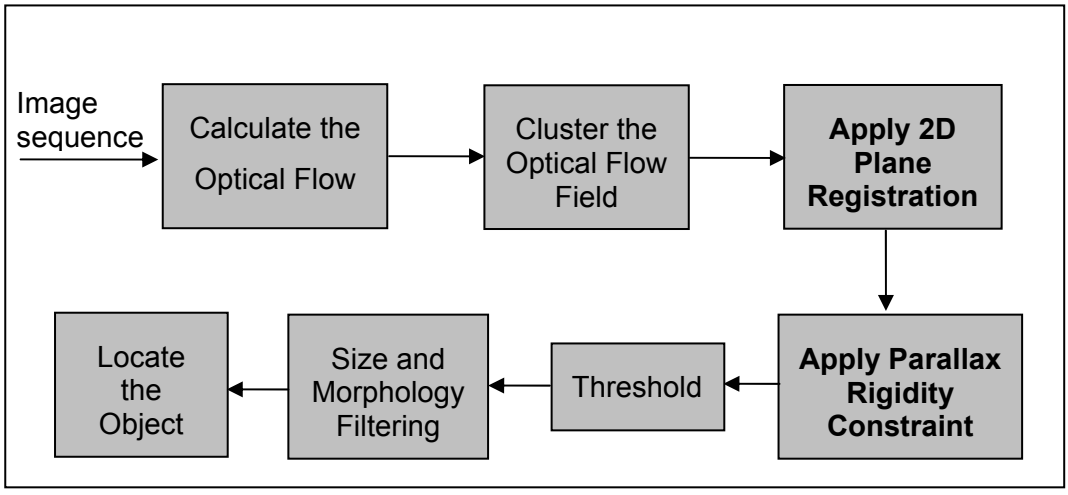


Figure 3-8 The block diagram of IMO detection framework in 3D scenes with moving camera sequences

3.5 Integration of the Algorithms

Up to now, algorithms for different schemes, 2D, the intermediate multilayer and 3D, are introduced. These algorithms progressively increase in their complexity, ranging from the simpler 2D techniques to more complex 3D techniques. Actually, the computations required for the solution at one complexity level become the initial processing step for the solution at the next complexity level. Therefore, there is a natural bridge between the algorithms. However, there is no mechanism to state which algorithm to utilize automatically. In real world, it is possible to have different kind of situations (2D, 3D etc.), and camera motions (stationary, rotation only, translation only, translation+rotation etc.) even in the same sequence. Therefore, a mechanism to detect the kind of the scene or camera motion and select the detection type should be proposed.

The starting point of the mechanism is the optical flow field between successive frames and segmentation of this flow field.

The primary difference between the camera induced 2D motion and 2D projection of the independently moving objects is their intensity. The camera motion will induce 2D motion almost at every image point unlike the independently moving objects. Independently moving objects create 2D motion only at image points where the 2D projective motion is present. Therefore, if the induced optical flow field is dense, there should exist some camera motion and if it is sparse, the camera should be stationary. This statement usually holds at situations where the intensity variation within the scene is large enough and aperture is sparse. This observation lets distinguishing the stationary and non-stationary camera cases, and automates the related algorithms.

Although there can be many other selections, the measure for the density of the optical flow field is selected by using the magnitude field of the motion vectors and the ratio of the pixels that have a magnitude greater than a threshold. The magnitude threshold is selected as 20% of the mean magnitude. Only pixels that have nonzero magnitude are taken into calculations. A ratio of 20% is chosen for density discrimination. If the number of pixels that have magnitude greater than the

threshold is above 20% of the total number of pixels in the image, the camera is assumed non-stationary; otherwise, it is classified as stationary.

If the camera is found to be stationary, one should stop the algorithm selection, since the complexity for this case does not change. However, if the camera is non-stationary, further processing is required to state which algorithm to use depending on the camera motion type and scene structure (i.e. 2D, Multilayer, 3D).

If the optical flow field is dense, the segmentation of the optical field and motion parameters of the segments should give valuable information about the scene structure and camera motion. If the number of reliable segments equal to 1 then one can assume that the scene is distant or flat and/or the camera is making rotations or zooms only. 2D parametric registration should be sufficient for this scheme.

On the other hand, if the number of segments is smaller than 4, the scene can be described by 2D layers at different depths; therefore, multilayer approach is used. Scenes which contain more than 4 segments are considered as dense depth scenes and parallax-rigidity based approaches are utilized for these scenes.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

In this chapter, the efforts to validate the examined and designed architectures are given in detail. The chapter is organized to reveal the complexity levels of the algorithms. Therefore, stationary camera and non-stationary camera results are presented separately. Stationary camera results are presented in two sub-sections according to different algorithms employed. Non-stationary camera results are presented in three different sub-sections according to the complexity levels of the algorithms. 'Algorithm Integration' results are given implicitly in the sub-sections of the chapter. Moreover, the hardware used in video and image capture and software implemented for detection and artificial data generation are also be discussed in brief.

4.2 Experimental Setup

All experiments are carried out offline (not real time) with artificial and real sequences. This in turn, leaded us to develop a system to collect data in real-time and then build an algorithm to process this data off-line.

The real data is acquired using a low-resolution thermal camera and an off-the-shelf day camera. The records contain stationary, controlled non-stationary and uncontrolled non-stationary camera sequences. The records are captured using specialized PC video cards and MPEG capture software.

The algorithms are implemented in C++, compiled using MS Visual C++ 6.0 IDE. The application is realized using MFC 6.0 Library for user interfaces and

DirectX 8.0, GDIPlus Library, and CDirectShowWrapper and CGdiplusWrapper classes of TUBITAK-BILTEN [31] for video and image capturing and viewing.

Some artificial data is also created in MATLAB. The artificial data is used to validate 'affine parameter estimation', 'clustering of the optical flow field', 'parallax shape and rigidity constraint' algorithms and their implementation.

4.3 Stationary Camera Results

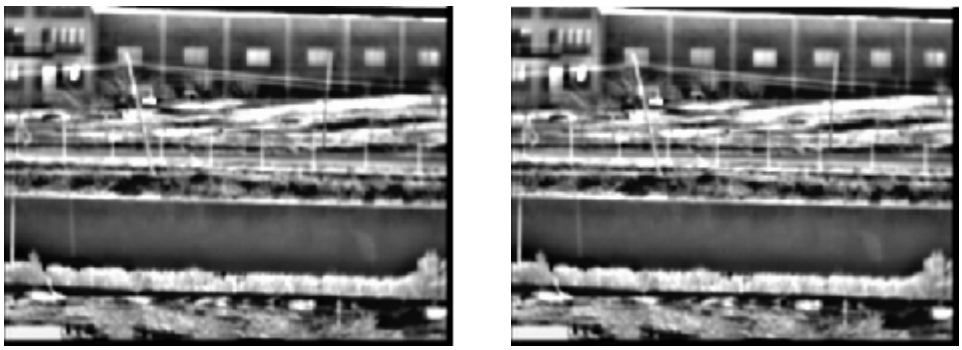
This section presents the results of independently moving object detection in stationary camera scenes, which is mentioned in Chapter 2. Two different methods were designed for IMO detection: 'Background Elimination' and 'Optical Flow Computation'. The results of these different techniques are given separately. The intermediate steps are also presented in order to make more consistent comments on the results.

4.3.1 'Background Elimination Method' Results

In this method, robust recovery of the background and thresholding has the primary effect on the results. The background is recovered in 3 ways: 'moving average of the previous frames', 'previous frame' and 'user intervention'. The threshold is estimated using 'Entropy Yen Method' [15].

The algorithm, which is discussed in Chapter 2.1, is applied to every consecutive frame of an image sequence of 750 frames taken from a 'Low Resolution Thermal Camera' that has an image size of 320x240. The suggested method is able to detect the independently moving objects throughout the sequence with different selections of background extraction methods, except from the 'user intervention' method. If the background is selected as the first frame of the sequence, the algorithm gives two false alarms. Other two background extraction methods give no false alarms throughout the sequence. However, the intermediate steps show that extracting the background using a moving average of the previous frames gives the best results.

The performance of the algorithm when there is no IMO in the sequence is important to show the noise susceptibility and false alarm resistance. Therefore, the first experiment is conducted by using the 'moving average background' and when there is no IMO. Figure 4-1 shows the 85th frame of the sequence and the background calculated up to that frame. It is observed that background can be reliably computed by the algorithm. Figure 4-2 shows the difference and thresholded images. The noise present in the process can easily seen from the Figure 4-2(a). Threshold at that point is 43 and only a small bird flying through the road remained after the thresholding process. This small bird is removed by morphological operations and the result is an empty figure (therefore, it is not illustrated).



(a) **(b)**

Figure 4-1 **a) 85th frame of the sequence**
b) Estimated background up to 85th frame

Figure 4-2 **a) Difference between 85th frame and background**
b) Thresholded difference Image, T=43

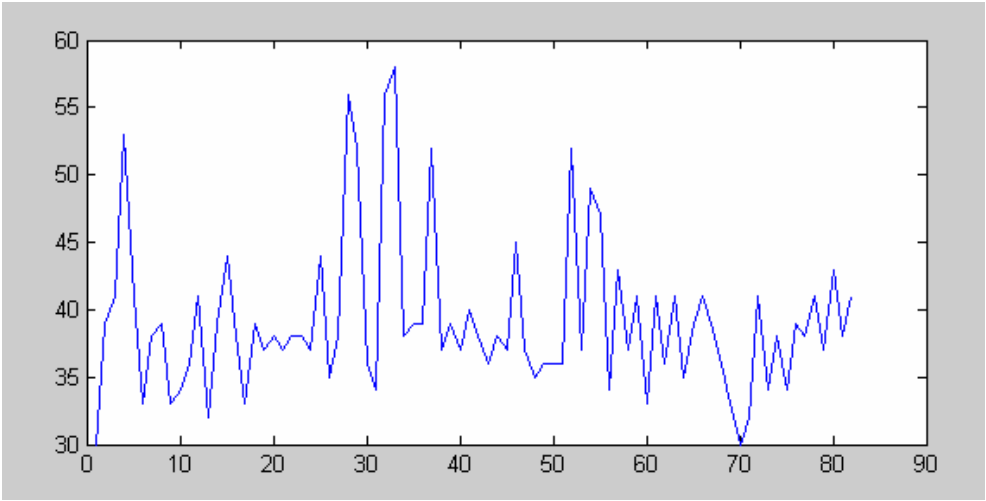


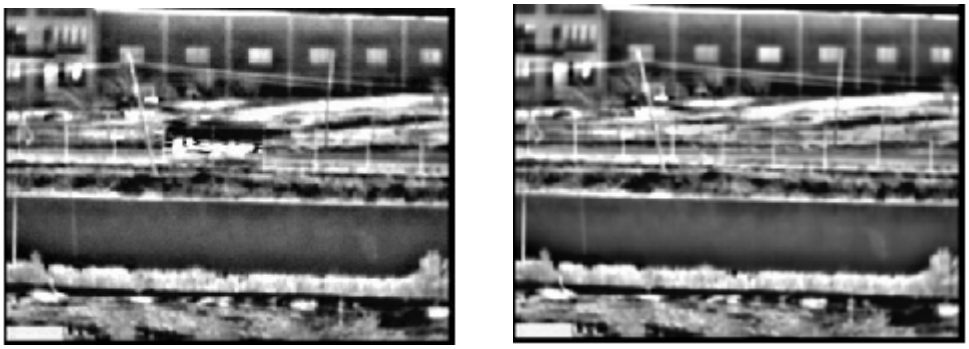
Figure 4-3 **Estimated threshold values up to 85th frame**

Figure 4-3 shows the threshold values , which are automatically calculated by Yen method, up to 85th frame. The variation in the threshold is due to the varying intensity values of the scene due to camera noise and thermal characteristics of the scene points, since the sequence is recorded by a thermal camera.

Figure 4-4, Figure 4-5 and Figure 4-6 show similar results when there is IMO in the scene.

In Figure 4-4(b), it is observed that the averaging filter minimizes biasing of the IMOs to the background extraction.

Note that in Figure 4-5 thresholding removes most of the noise in the difference image, since the algorithm estimates the background reliably. The threshold is determined as 67 at that point. The morphological operations tighten the object and ease the localization problem. In Figure 4-6, it is observed that the threshold tends to increase when there is an independently moving object in the scene. This is due to the existence of higher intensities at the difference image, when there are IMOs in the scene. If a global threshold was selected, it would not respond to these kind of changes.



(a) **(b)**

Figure 4-4 **a) 110th frame of the sequence**
b) Estimated background up to 110th frame

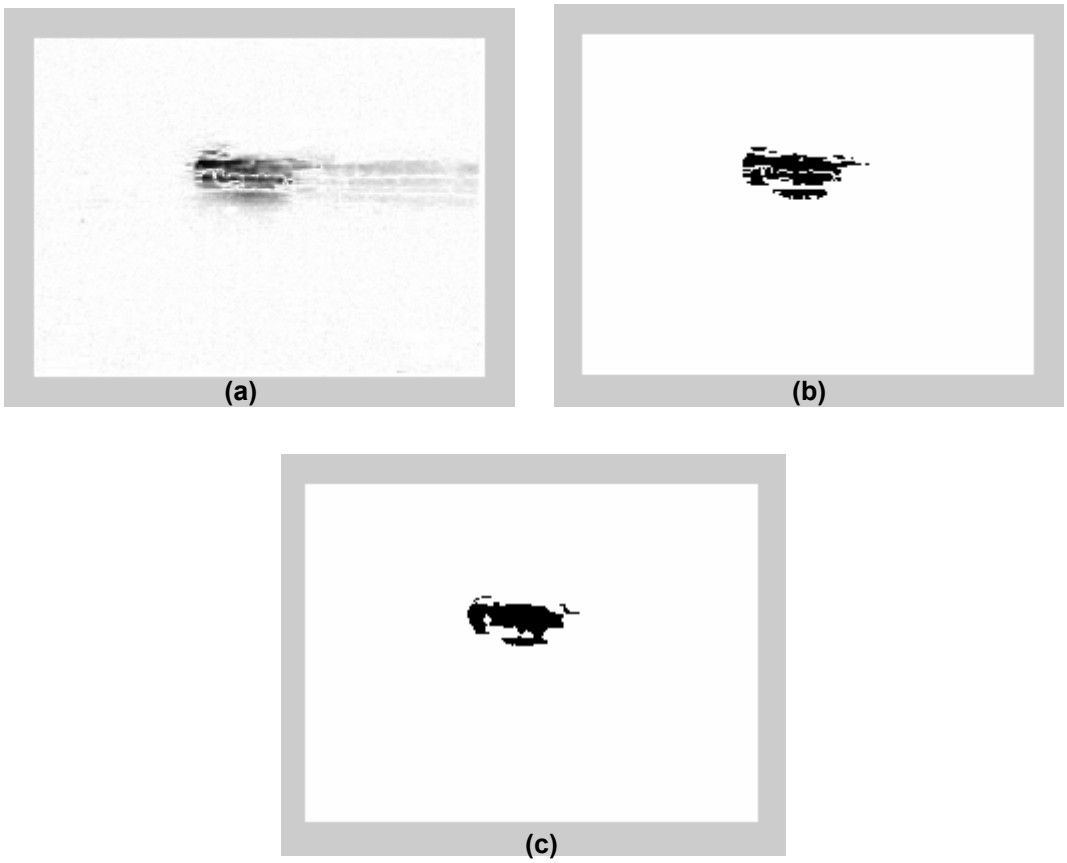


Figure 4-5 a) Difference between 110th frame and background
b) Resulting thresholded difference image, T = 67
c) Resulting image after morphological operations erosion and dilation are applied

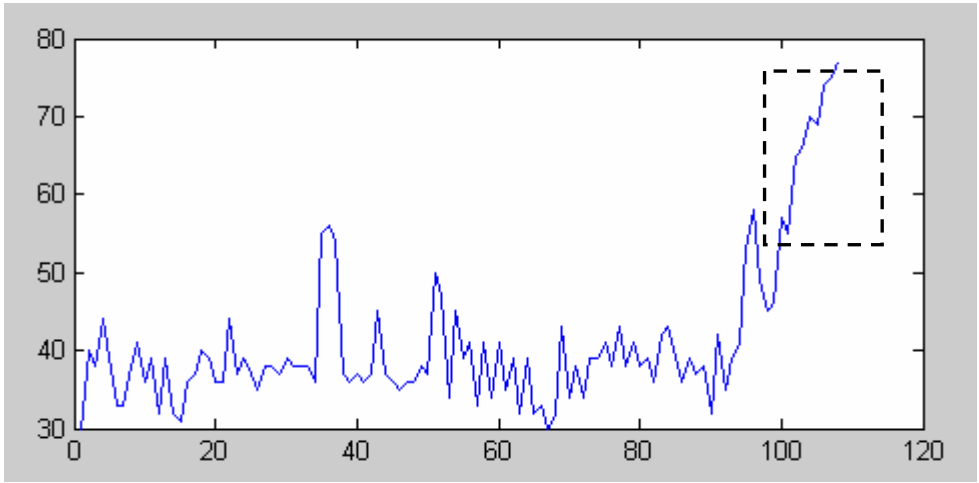


Figure 4-6 Threshold variation up to 110th frame

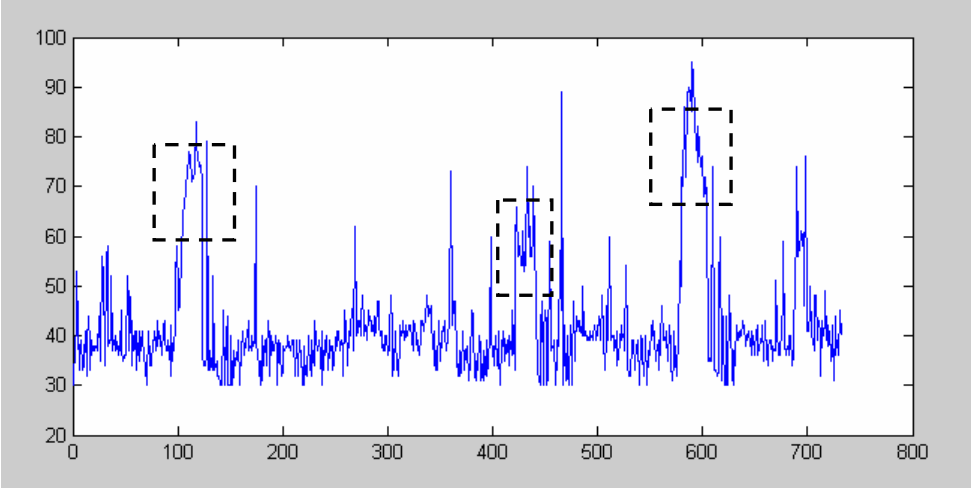


Figure 4-7 Threshold variation throughout the whole sequence, background is selected as the moving average of the previous frames

Figure 4.7 shows the threshold variation throughout the whole sequence. The dotted square regions in the plot indicate the duration where there is IMO. The threshold adaptation is clearly identified from this figure.

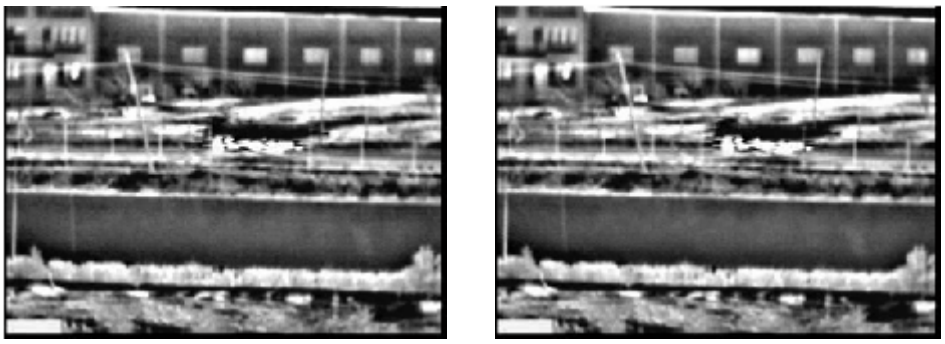
When the previous frame is selected as the background, the background processing power increases, since the background is recent (thus, thermal characteristics of the scene points are recent). However, foreground processing power decreases, since it depends on the IMO motion and thermal characteristics. If the IMO is moving slowly and/or it has a uniform thermal radiation, the difference between two consecutive frames will be poor.

In the processed sequence, the IMOs are cars or trucks. Therefore, their thermal characteristics are textured enough to disable aperture and give satisfactory results. Figure 4-8,Figure 4-9 andFigure 4-10 shows the results of detection algorithm when there is IMO in the scene.

In Figure 4-9(a), the difference between 105^{th} and 104^{th} frame is given. It is observed that the background noise is smaller than the 'moving average background' case, whereas the strength of the IMO decreased in the difference image. Figure 4-9(b) and Figure 4-9(c) gives the processed difference image with

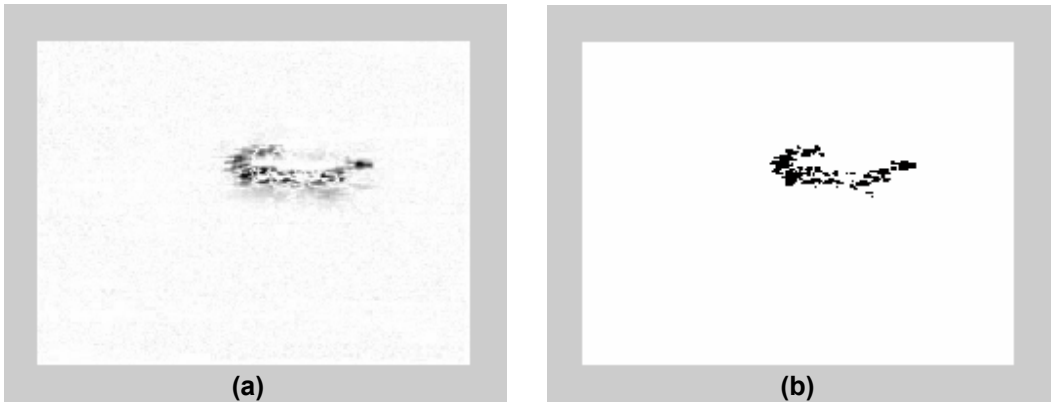
thresholding and morphological operations. The threshold is determined to be 72 at this point.

Figure 4-10 shows the threshold variation up to 105th frame and similar to previous result the threshold tends to increase, when there is IMO in the scene. Figure 4-11 illustrates the threshold variation throughout the whole sequence. The IMO regions can clearly be identified from threshold variation points that are shown with dotted square regions.



(a) **(b)**

Figure 4-8 **a) 105th frame of the sequence**
b) 104th frame of the sequence is selected as the background



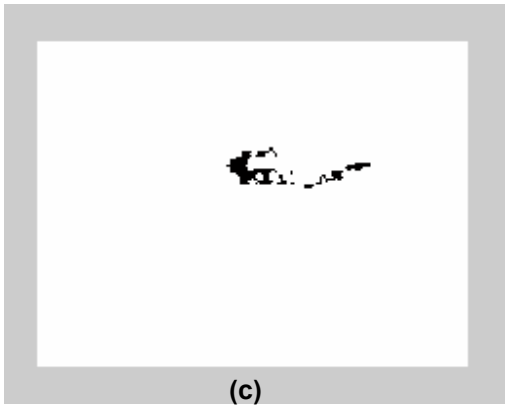


Figure 4-9 a) Difference between 105th frame and background
b) Thresholded difference image, T=72
c) Resulting image after morphological operations erosion and dilation are applied

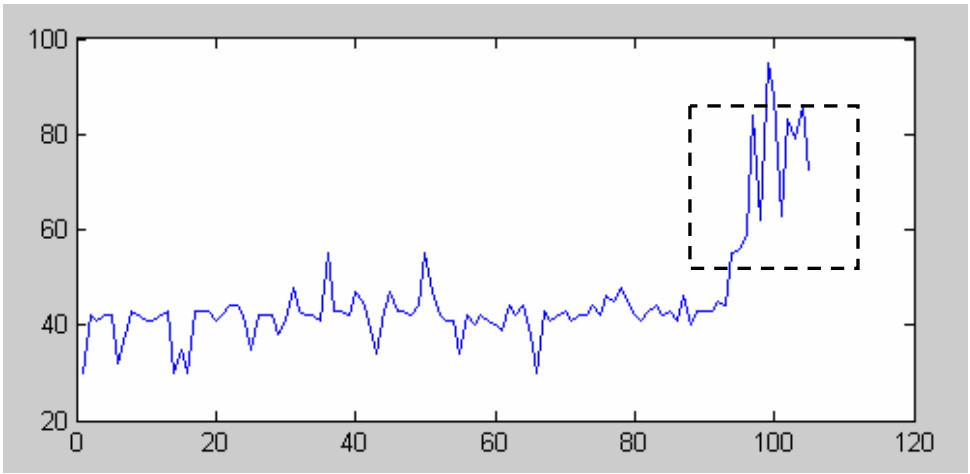


Figure 4-10 Threshold variation up to 105th frame

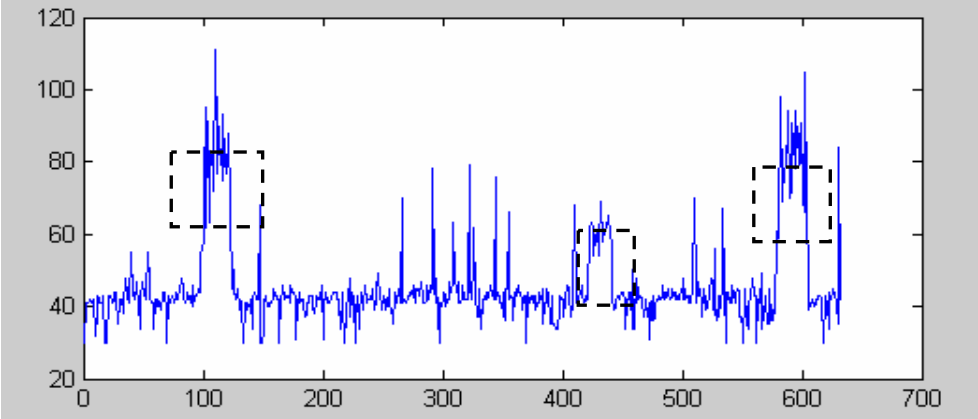


Figure 4-11 Threshold variation throughout the whole sequence, background is selected as previous frame

If one of the initial frames is selected as background, the background processing suffers due to changing thermal conditions through the sequence. This results in false alarms. In the processed sequence for background extracted with user intervention, we have observed two false alarms. Figure 4-12 shows the threshold variation throughout the whole sequence. It is observed that the threshold variation is high due to non-adaptive background selection. The dotted square regions show the IMO regions. This algorithm detects all three IMOs. The dotted circle regions show the false alarm regions.

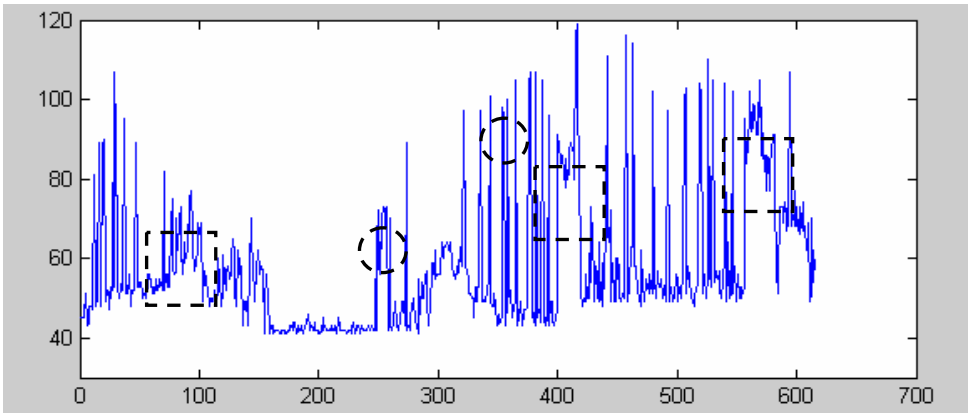


Figure 4-12 Threshold variation throughout the whole sequence, background is selected by user

4.3.2 Optical Flow Method Results

In this method, the reliable computation of the optical flow has the primary effect on the results.

Instead of applying to a whole sequence, the algorithm is only applied to an image pair, since the computation time is up to 2 minutes for each pair on an AMD Athlon 1.7 GHz Computer. This long computation time is due to the fact that the algorithm is applied to every image point 240*320 = 76800 and due to the iterative nature of optical flow calculation of KLT algorithm.

Figure 4-13 shows two consecutive frames of a sequence taken from a low-resolution thermal camera. The optical flow field between these two frames is computed first. Only a small portion of the optical flow field is given in Figure 4-14 in order to present the results better. This portion corresponds to the IMO location. It is observed that the motion field is recovered consistently; i.e. all the flow vectors have same direction and magnitude as expected. The integration algorithm is applied to this optical flow field and the result indicates that this optical flow field belongs to a stationary camera sequence. The related parameters are as follows:

Mean of the non-zero motion vectors = 1.36807
Magnitude threshold of the motion vectors (0.2 x Mean) = 0.2736
Number of total pixels = 66000 (300x220)
Number of pixels above the threshold = 5007
5007< %20 of 66000 => camera is stationary.

The algorithm selection procedure directed us to the stationary camera algorithms. At this point, one might use the previously developed approach (background elimination) for IMO detection but since we have the optical flow field, we go on with “the optical flow field approach”.

The magnitude field of the optical flow field and its thresholded and morphologically processed version is given in Figure 4-15 . The threshold in this scheme is 0.784. This threshold is computed using mean and standard deviation of the magnitude field, $T_{opt} = Mean - 0.2 * Std$, as suggested in [15].

The statistical parameters, mean and standard deviation, of the resulting vector field are computed for similarity measure calculation. The mean of the motion vectors in x-direction is -7.909 and in y-direction it is 0.0879. The standard deviation of the motion vectors in x-direction is 1.618 and in y-direction it is 0.763. the similarity computed for x and y components are:

Similarity of x-component vectors = 0.326
Similarity of y-component vectors = 0.730

Both similarity measures are below the threshold 1. Therefore, the detected objects motion vectors are said to be similar and belong to the same object,

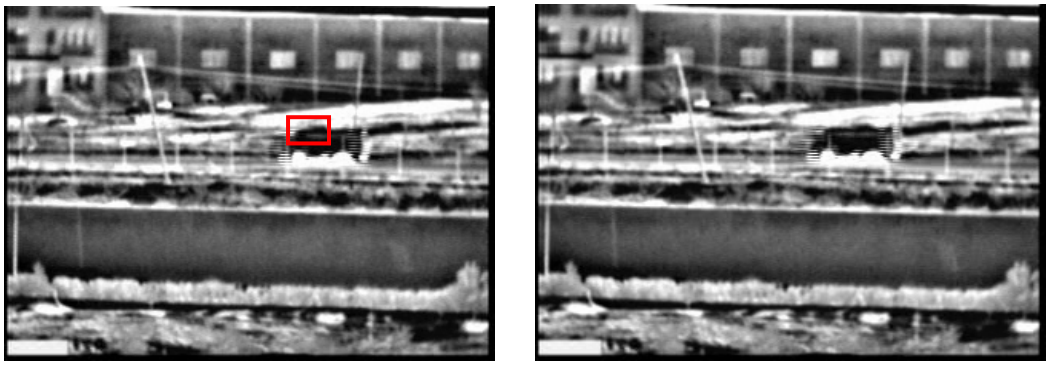


Figure 4-13 Consecutive frames of a sequence taken from a stationary thermal camera

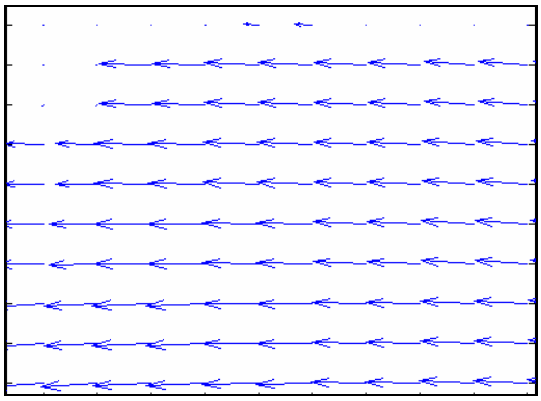


Figure 4-14 A portion of the optical flow field that belong to the IMO

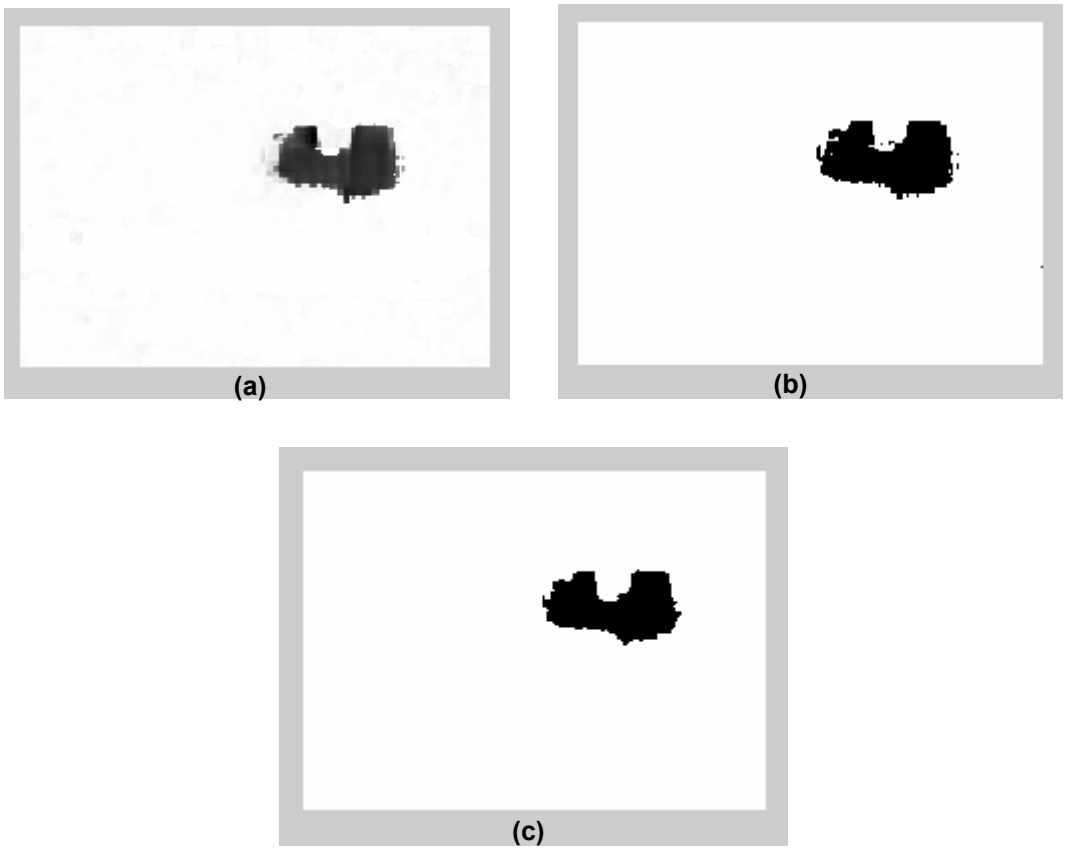


Figure 4-15 **a) The magnitude field of the optical flow**
b) Thresholded magnitude field, T=0.784
c) Resulting image after morphological operations

4.4 Non-Stationary Camera Results

This section presents the results of IMO detection in non-stationary camera scenes which are mentioned in Chapter 3. Three different schemes for IMO detection in non-stationary camera scenes is presented: Planar scenes, multi-planar scenes and 3D scenes. The results of these different schemes are given separately.

The algorithms are applied to artificial and real sequences and the intermediate steps are also presented in order to make more consistent comments on the results. The real data is taken from 'low resolution thermal camera' and

'commercial day camera' sequences and a modified version of the well known 'flower garden sequence' is also used during the experiments.

The algorithms are applied to the image pairs instead of the whole sequences due to long computation time.

4.4.1 Planar Scene Results

Robust recovery of the affine parameters of the dominant plane motion is the most important step in this algorithm.

In order to show the accuracy of the affine estimation algorithm, an artificial data is generated by MATLAB. In this data, there exists two planes, which are far away from the camera, but still close to each other. The camera is making translation and rotation. The camera motion parameters are:

T_x=0, T_y=0.2, T_z=0.1 ; Ω_x=0.05, Ω_y=0.05, Ω_z=0.

The parameters for the first plane are A=18, B=0, C=0, whereas those of the second plane are A=19, B=0, C=0. (Plane equation is $Z = A + B X + CY$). The induced and estimated affine parameters of the planes are given in Table 4-1.

Table 4-1 Induced and estimated affines

	Induced Affine Parameters		**Estimated Affines**
	Plane 1	**Plane 2**	
a	-0.05000	-0.05000	-0.054162
b	0.00556	0.00526	0.005411
c	0.00000	0.00000	-0.000048
d	0.06111	0.06053	0.065080
e	0.00000	0.00000	-0.000396
f	0.00556	0.00526	-0.004895
g	-0.05000	-0.05000	-
h	0.05000	0.05000	-

The algorithm perceived the image as a single plane, since the affine parameters of the planes are very close to each other. The estimated affines are very close to real values , except the 6^{th} parameter, f. Note that, 7^{th} and 8^{th} (g and h) affines are not estimated to make the calculations linear. The results show the effects of linearization on 2D motion field induced by the motion of close planes

The detection algorithm is applied to real image pairs taken from a low-resolution thermal camera. At the first step, optical flow field between the image pairs is computed and clustered. Based on the optical flow field characteristics and cluster numbers the camera is identified as non-stationary and the scene is identified as a planar scene. Thus, single 2D parametric registration of the dominant plane process is applied for IMO detection.

Figure 4-16 shows two images taken from a rotating thermal camera. The camera is rotating in Y-axis and in clockwise direction. The camera motion is controlled by a two-axis tri-pod.

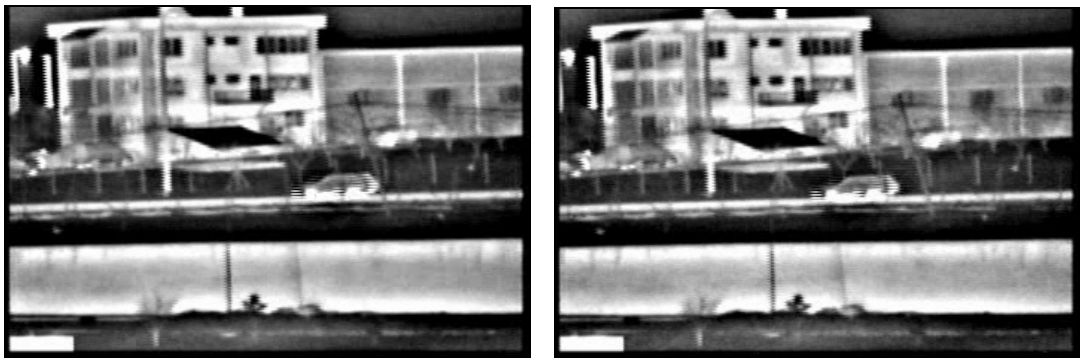


Figure 4-16 An image pair taken from a moving thermal camera

The integration algorithm is applied to this optical flow field and the result stated that this optical flow field belongs to a non-stationary camera sequence. The related parameters are as follows:

Mean of the non-zero motion vectors = 4.585

Magnitude threshold of the motion vectors (0.2 x Mean) = 0.917

Number of total pixels = 73040 (332x220)

Number of pixels above the threshold = 43248

43248 > %20 of 84480 => camera is non-stationary.

Figure 4-17 shows the difference between these two consecutive frames. It is observed that due to the motion of the camera, the difference image is highly cluttered and it is impossible to detect the independently moving car by simple thresholding or by morphological operations.

Figure 4-17 The difference of the image pair given in Figure 4-16

The optical flow field is segmented out and the result is a single plane. The affine parameters of this plane are calculated as follows:

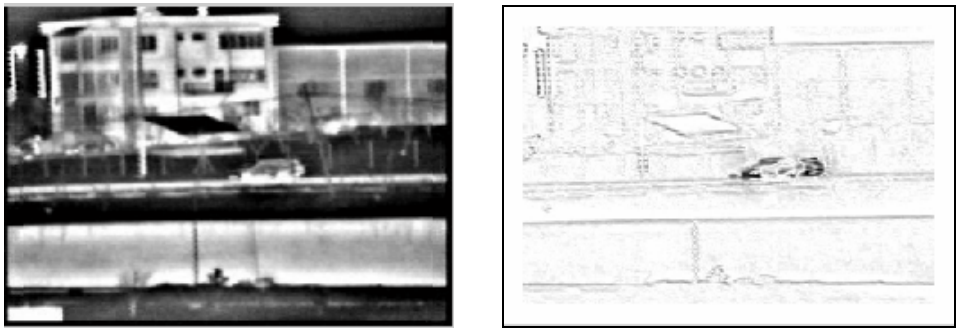
a = -4.191942 b = 0.039537 c = -0.161855

d = 0.019666 e = -0.012608 f = 0.015828

Note that the 1st affine parameter dominates the affine parameter space. This is expected, since $a = -f_c \alpha V_x - f_c \Omega_y$, meaning that rotation along Y axis directly contributes to 1st affine parameter, a. It is observed that the remaining affine parameters are very small, but still nonzero. This may be due to uncontrolled camera motion (i.e. The camera is not leveled, therefore rotation in Y axis may

contribute a small rotation in X axis), measurement noise coming from optical flow field and inherent noise in affine estimation process.

Figure 4-18(a) shows the warped state of the second image of Figure 4-17. The warping is achieved according to the estimated affine parameters. Figure 4-18(b) shows the difference between the warped and the first image. It is seen that most of the background is successfully registered. Figure 4-19 shows the thresholded difference image and the result. The threshold is determined as 56 by the Yen algorithm. It is obvious that the result is able to locate the independently moving object.



(a) **(b)**

Figure 4-18 **a) Warped second image according to the dominant planes' affine parameters**
b) Difference of the warped image and the first image

(a) **(b)**

Figure 4-19 **a) Thresholded difference image, T=56**
b) Difference image after morphological operations (erosion and dilation)

4.4.2 Multi-Planar Scene Results

Reliable clustering of the optical flow field and robust recovery of the affine parameters of each cluster are the most important steps in this scheme, since this method is a modified version of the single parametric registration process.

In order to show the accuracy of the clustering and affine estimation algorithms, an artificial data is generated by MATLAB. In this data, there exists three planes, two of them faraway from the camera (planes 2 and 3) but close to each other and the third one (plane 1) close to the camera and two independently moving objects (a truck and a helicopter). Figure 4-20 presents this artificial scene.

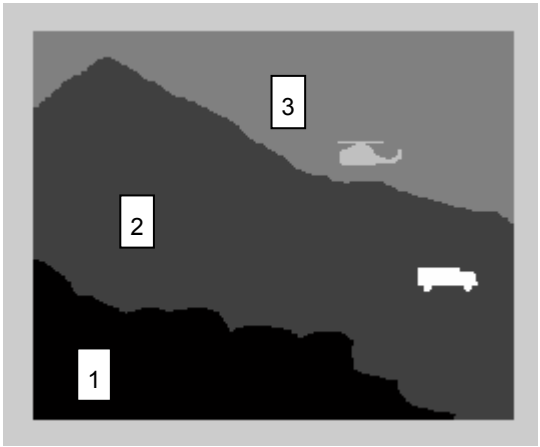


Figure 4-20 Artificial scene created at Matlab for 'MultiLayer Detection Algorithm' verification

The camera is both translating and rotating. The camera translation creates different 2D image motions for the planes due to their depth difference. The camera motion parameters are selected as:

$T_x=0$, $T_y=0.2$, $T_z=0$; $\Omega_x=0$, $\Omega_y=0.05$, $\Omega_z=0$

The parameters for the first plane are A=0.5, B=0, C=0, the second planes' parameters are A=18, B=0, C=0.05 and the third planes parameters are A=18, B=0, C=0.02. (Plane equation is $Z = A + BX + CY$)

The independently moving objects motion vectors are as follows:

Helicopter: u = -0.3 ; v = -0.2 **Truck :** u = 0.2 ; v = -0.5

The clustering algorithm detected the two planes in the scene and clustered the scene accordingly. The independently moving objects are discarded, since they are small. The third plane is not recognized, since induced motion vectors of 2nd and 3rd planes are very close to each other. Figure 4-21 shows the clustered flow field.

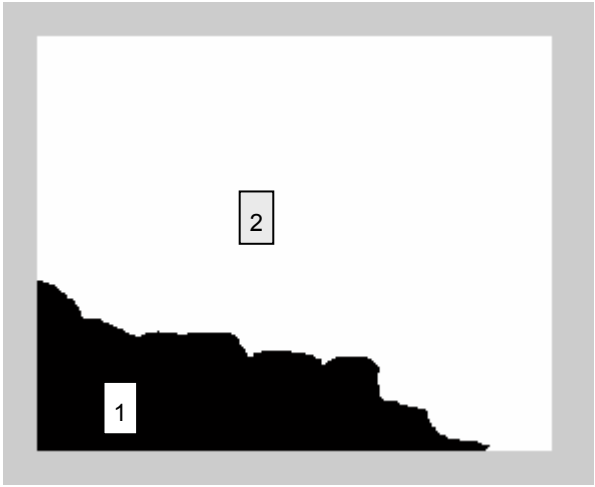


Figure 4-21 Result of Clustering, white and black indicate two different planes

The induced affine parameters of the planes and the estimated affine parameters are given in Table 4-2.

Table 4-2 Induced and estimated affines

	Induced Affine Parameters			Estimated Affines	
	Plane 1	**Plane 2**	**Plane 3**	**Plane 1**	**Plane 2**
a	-0.05000	-0.05000	-0.05000	-0.051232	-0.054254
b	0.00000	0.00000	0.00000	0.000041	-0.000005
c	0.00000	0.00000	0.00000	-0.000002	-0.000002
d	0.40000	0.01111	0.01053	0.405228	0.012443
e	0.00000	0.00000	0.00021	0.000058	-0.000018
f	0.00000	0.00056	0.00000	-0.000047	0.000026
g	-0.05000	-0.05000	-0.05000	-	-
h	0.00000	0.00000	0.00000	-	-

It is observed that the clustering and affine estimation algorithms are performing quite well and give satisfactory results. Ideally, sequential registration of the planes based on these affines will result for the perfect detection of IMOs.

However, in natural sequences clustering of the optical flow field is not perfect, since the estimated optical flow field contains some errors. The exact boundaries of the planes are difficult to detect; in fact the planes may even not have exact boundaries. In addition, the optical flow field is not fully dense. These reasons cause errors in clustering and affine estimation.

This fact is illustrated in a typical image sequence. Figure 4-22 shows two consecutive frames taken from the “Flower Garden Sequence”, where the camera is translating to the right. Figure 4-23 shows a down-sampled version of the optical flow field between these consecutive frames. Down sampling is achieved just for illustrative purposes. The planes at different depths have different motion vectors, since the camera is performing translational motion.

Figure 4-24 shows the segmented optical flow field. The tree, the flower garden and the sky is segmented almost properly. The improperly segmented parts are due to noise coming from optical flow field calculation, the optimization of the inner parameters of segmentation process.

Figure 4-22 Two consecutive frames from “Flower Garden Sequence”

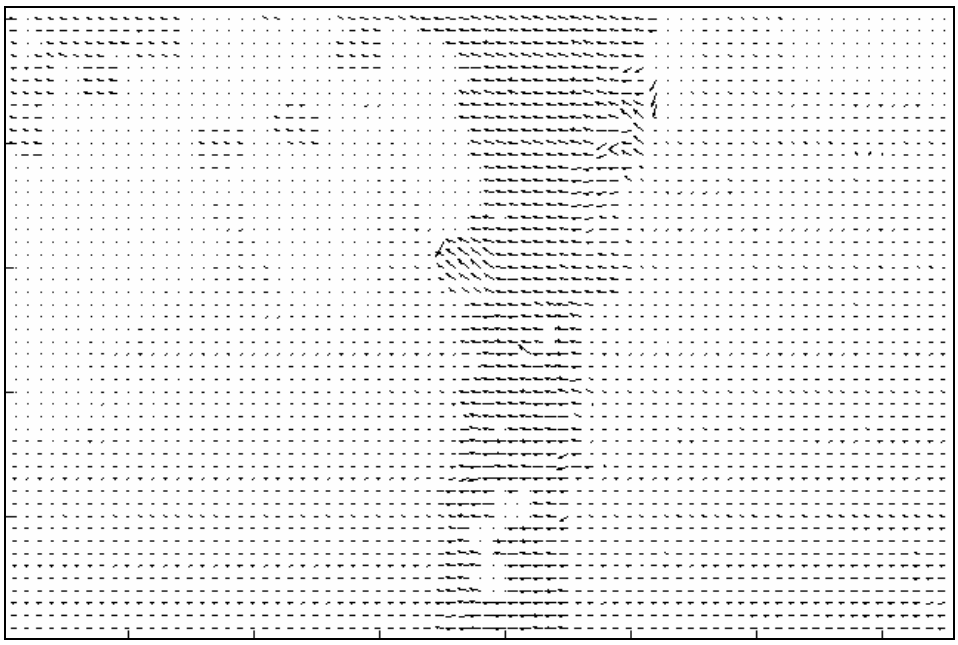


Figure 4-23 Optical Flow Field between the frames given in Figure 4-22

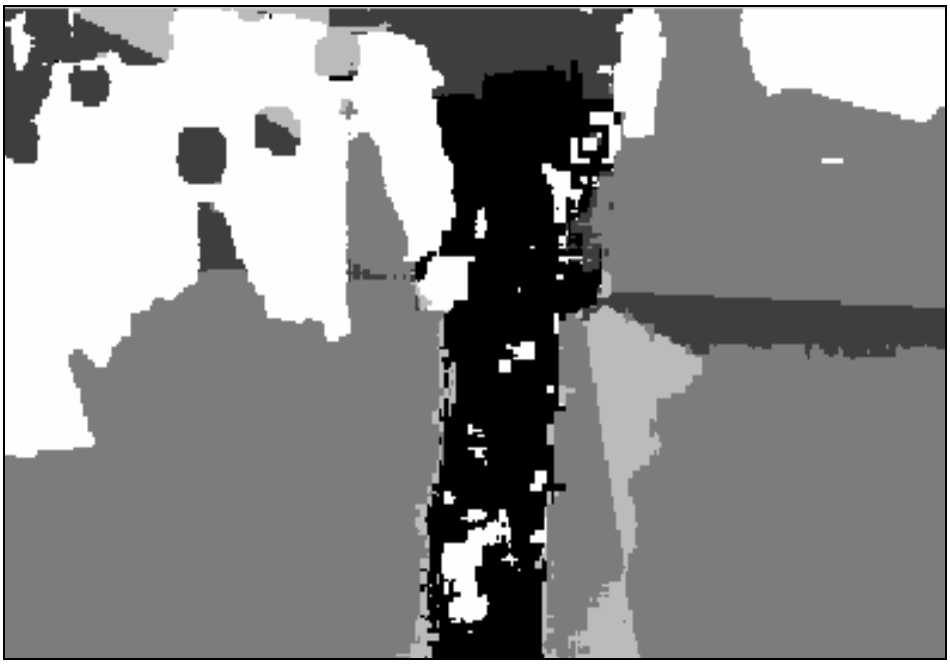


Figure 4-24 Result of segmentation. Number of segments is 5.

The same segmentation framework is applied to a multilayer scene for IMO detection. Figure 4-25 shows two consecutive frames taken from a day camera. The camera is translating to the right.

Figure 4-25 Two consecutive frames taken from a day camera

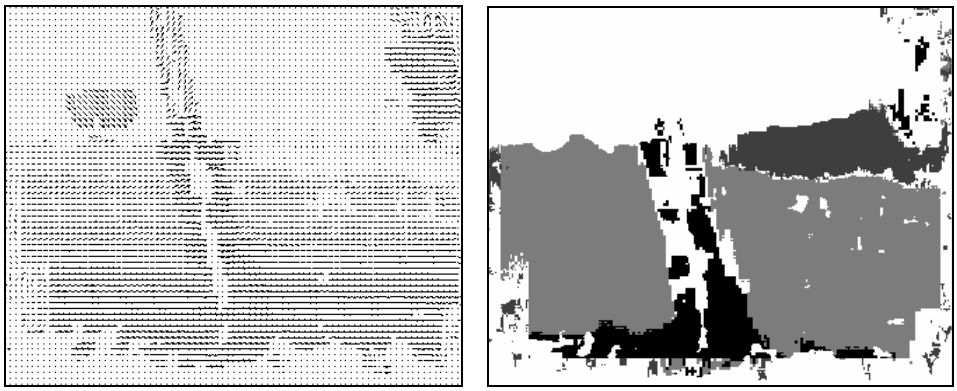


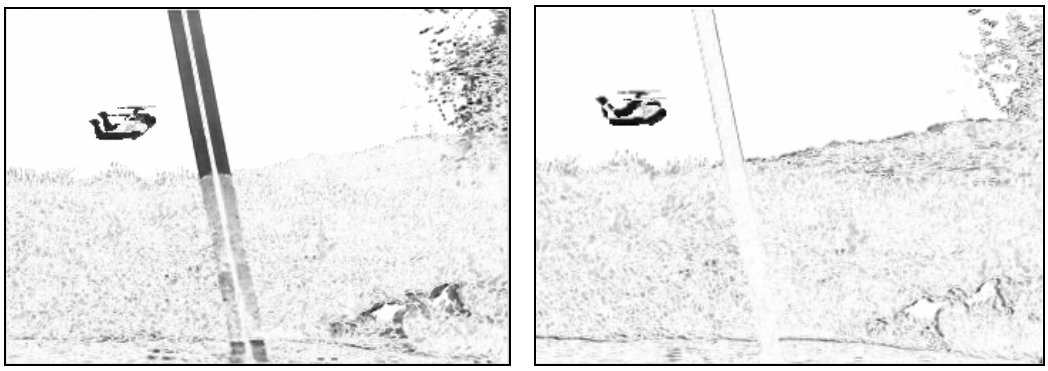
Figure 4-26 a) Optical Flow between the frames
b) Segmented optical flow field. Number of segments is 3

Figure 4-26 shows the resulting estimated optical flow field between these frames and the result of the segmentation for the optical flow field. The number of segments is determined as 4. Some flow vectors are not processed since none of the affines is able to fit within acceptable error limits. For example, most of the flow vectors belonging to the tree branch at the top right are not processed. The zero and non-reliable flow vectors are painted to white in the resulting segmentation image. The helicopter is not segmented out since it is small. The reliable segments are used in consecutive registration for IMO detection. The affine parameters of these planes are given in Table 4-3.

Table 4-3 Affine parameters of the segments (planes)

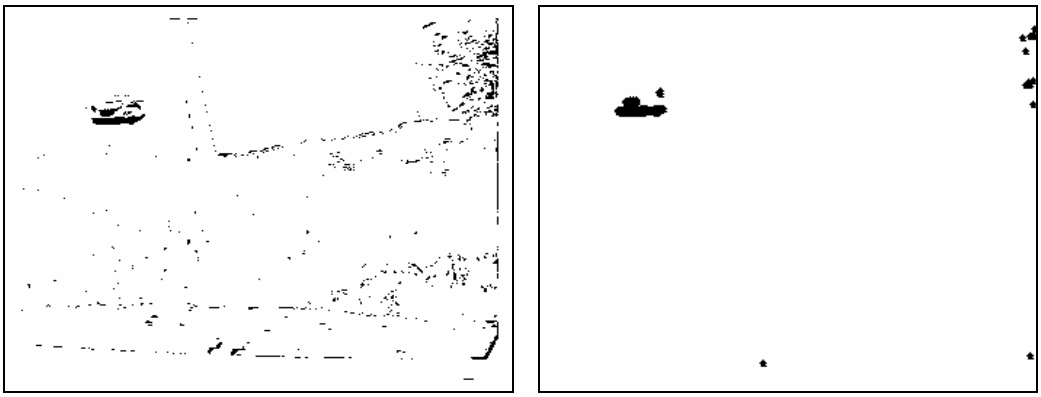
	Estimated Affine Parameters		
	Plane 1 (foreground)	**Plane 2 (background)**	**Plane 3 (mast)**
a	2.990174	0.697773	9.287494
b	-0.000058	-0.009532	0.002393
c	0.044135	-0.001776	0.007251
d	0.053848	0.019301	-0.467732
e	-0.001887	-0.001620	0.000595
f	0.001606	-0.000248	0.006704

The difference in the motion characteristics of the planes can easily be observed from the affine parameters. The mast (i.e. the electricity beam) that is very close to the camera has the highest affine parameter set. The foreground and background layers have smaller affine parameters, respectively. The registration starts with the largest plane and continues with the preceding large plane. Figure 4-27 shows the original difference image, the difference after warping the mast as an intermediate step, and the result of the registration process followed by thresholding and morphology.



(a) (b)

Figure 4-27 a) The difference image between the frames of Figure 4-25
b) The difference image after the warping of the mast



(a) (b)

Figure 4-28 a) The final thresholded difference image after all the planes are warped.
b) The result after morphological operations are applied

4.4.3 3D Scene Results

2D parametric registration of the dominant plane and parallax rigidity constraint utilization enable the detection of independently moving objects in the scenes which have dense or sparse parallax motion.

In order to show the accuracy of the parallax rigidity constraint algorithm, an artificial data is generated by MATLAB. In this data, there exists 8 planes at different depths and with different surface parameters, since this must be a 3D scene. In addition, there exist two IMOs in the same scene. Application of the parallax rigidity constraint requires three consecutive frames, therefore the camera makes two sets of motions to generate the artificial optical flow fields. The camera is making translation and rotation both. The camera motion parameters are

Motion 1 : $T_{x1} = 0$, $T_{y1} = 0.2$, $T_{z1} = 1.5$; $\Omega_{x1} = 0.25$, $\Omega_{y1} = 1.5$, $\Omega_{z1} = 0$

Motion 2 : $T_{x2} = 0$, $T_{y2} = 0.2$, $T_{z2} = 1.5$; $\Omega_{x2} = 0.25$, $\Omega_{y2} = 1.5$, $\Omega_{z2} = 0$

The motion parameters of the independently moving objects are:

Helicopter: $u_1 = -1.1$; $v_1 = -0.2$ $u_2 = -1.1$; $v_2 = -0.2$

Truck : $u_1 = -2$; $v_1 = -0.5$ $u_2 = -2$; $v_2 = -0.5$

For generating the artificial data, the optical flow vectors u and v is computed at every image point using the plane parameters, and camera motion parameters as derived in Chapter 3. The affine parameters for each plane is computed by using the formulas

$$u = a + b\,x + c\,y + g\,x^2 + h\,xy \quad , \quad v = d + e\,x + f\,y + g\,xy + h\,y^2$$

Figure 4-29 shows the 3D scene generated by MATLAB.

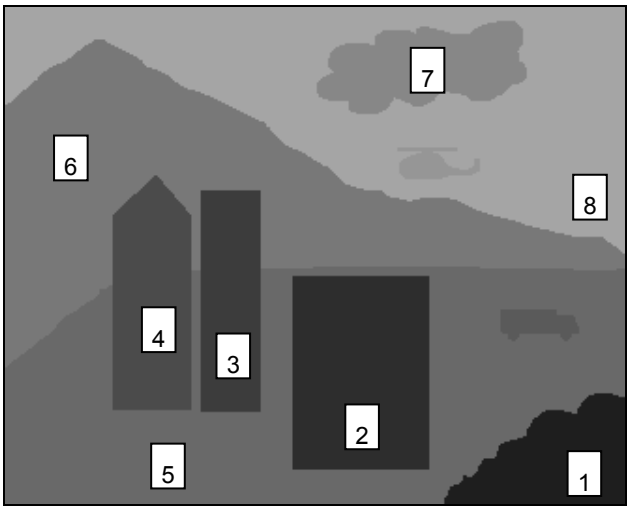


Figure 4-29 3D scene generated by MATLAB

The plane parameters are given in Table 4-4.

Table 4-4 The plane parameters of the 3D scene

	A	B	C
Plane 1	0.8	0	0
Plane 2	1.2	0	0
Plane 3	2	0	0
Plane 4	2.5	0	0
Plane 5	2	0.03	0.05
Plane 6	7	0	0
Plane 7	8	0	0
Plane 8	158	0	0

In order to apply the parallax rigidity constraint to this sequence, one of the planes should be parametrically aligned to remove the effects of rotation. Figure 4-30 shows the result of parallax rigidity constraint applied for IMO detection when the 2D parametric registration step skipped. This result is presented to show the effect of registration process.

Figure 4-30 Parallax rigidity constraint applied to the artificial data without 2D parametric registration.

The reference point is located at bottom of plane 4. This is a proper choice since it belongs to the static background. The helicopter is detected because moves in the opposite direction. However, the truck cannot be detected, since it does not have an extreme motion characteristic. In addition, plane 3 and plane 4 seem to violate the parallax rigidity constraint.

When one registers one of the planes and apply rigidity constraint, the results become much more reliable as shown in Figure 4-31.

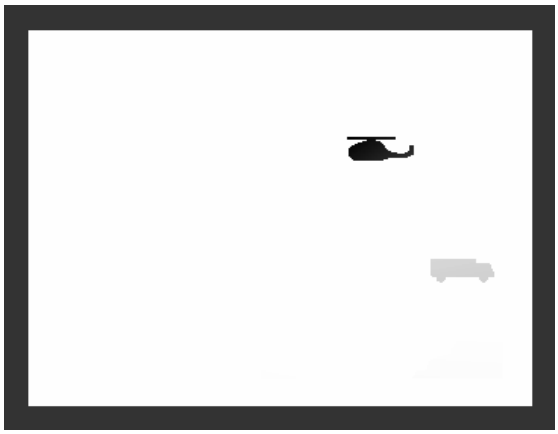


Figure 4-31 Parallax rigidity constraint applied to the artificial data after 2D parametric registration

The registration is conducted according to the parameters of plane 6 (mountain). The same point on plane 4 is selected as reference. Figure 4-31 implies that the parametric registration removes all the effects of rotation and the rigidity constraint can be perfectly applied. The static background is totally eliminated with this algorithm, although different planes belonging to the static background have different 2D image motions.

The same algorithm is applied to a three-frame sequence taken from a day camera. The algorithm makes the reference point selection and plane registration automatically. Figure 4-32 shows these 3 frames. The reference point is selected using the Harris corner detector. Figure 4-33 shows the corner map of the first image. The highest density corner is on top of the right hand-side tree as marked. In this sequence camera only makes Z-translational motion. Therefore, the 2D parametric registration process is useless and is not conducted. Figure 4-34(a) shows the difference image of the first two frames. It is observed that moving objects cannot be identified from the difference image. Figure 4-34(b) shows the parallax rigidity constraint applied to this scenario. The algorithm reliably detected the independently moving objects (cars), since they are violating the rigidity constraint.

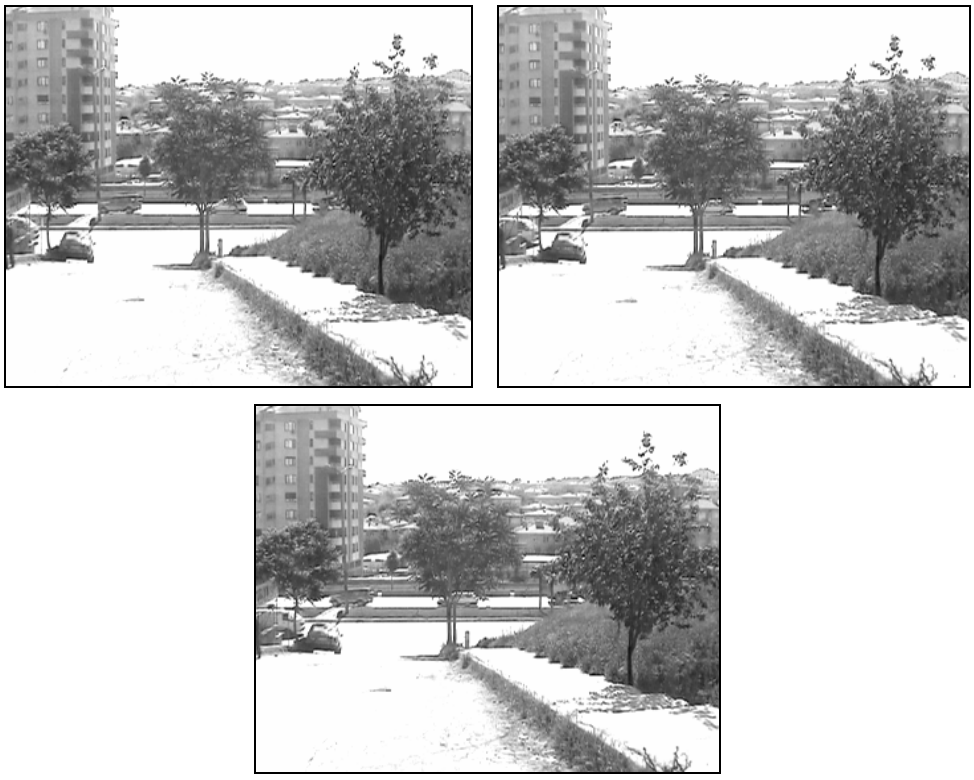


Figure 4-32 Three consecutive frames taken from a translating camera

Figure 4-33 Corner map of the first frame of Figure 4-32. The highest density corner point is at the middle of the circle.

(a) (b)

Figure 4-34 a) The difference image of the first and second frame.. b) The result of parallax rigidity constraint without 2D parametric registration

In order to show the effect of registration, the algorithm is applied to a sequence where the camera is performing both translation and rotation. The sequence is again taken from a day camera. Figure 4-35 shows three frames taken from this sequence.

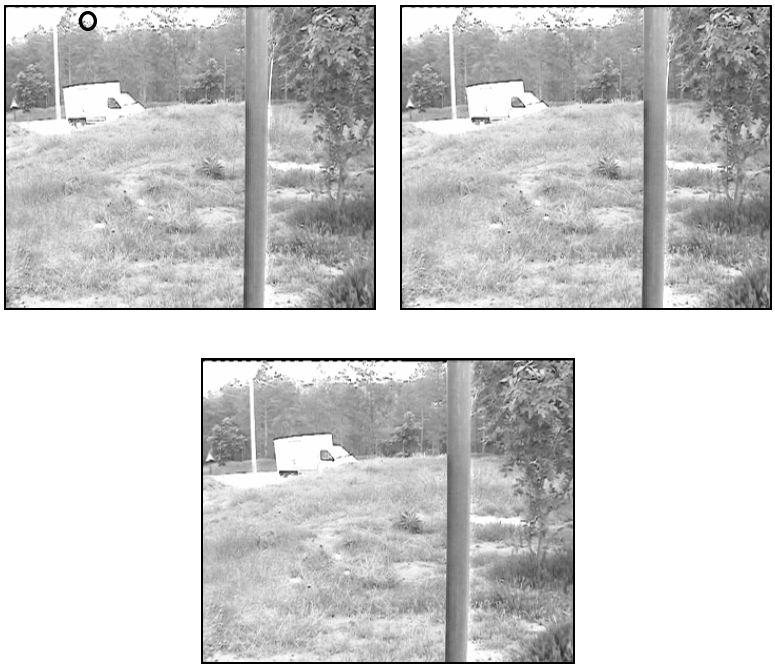


Figure 4-35 Three consecutive frames taken from a rotating-translating day camera

The optical flow field between these frames is segmented out and 7 different planes are detected. Figure 4-36 shows the result of segmentation. The largest segment (plane) is registered for parallax rigidity constraint application. The reference point is determined on top of the background plane composed of trees as marked in the first image.

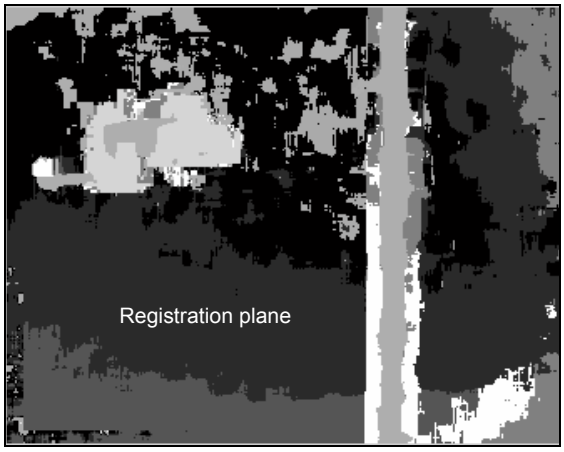


Figure 4-36 Result of segmentation. Number of segments is 7.

Figure 4-37 a) Result of Parallax Rigidity before registration
b) Result of Parallax Rigidity after registration

Figure 4-37 shows results of “Parallax Rigidity Constraint” applied to the sequence before and after registration. The registration process removed the effect of camera rotation and the result became more visible.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Automatic independently moving object detection is an important goal in video surveillance applications, for guidance of autonomous vehicles, for efficient video compression, for smart tracking, for automatic target recognition (ATR) systems, for terminal phase missile guidance, and for many other applications.

In this thesis, independently moving object detection problem is analyzed for different camera motions and scene structures. The detection methodology may vary for different applications. However, the solution for one case can be a pre-processing step for the others. At the first glance, the problem should be separated into two classes: Moving Object Detection in Stationary Camera Sequences and in Non-Stationary Camera Sequences and examined separately.

In stationary camera sequences, two solutions were proposed: Background Elimination and Optical Flow Based Method. The background elimination method gives promising results in the controlled environments, when the camera is strictly fixed. The reliability of the background extraction process turns out to be the most important step in this method. The best results can be obtained, when “moving average selectivity method” is used for background extraction. Thresholding and morphological operations steps are also very important for binarization and localization of the detection results. Yen’s method based on histogram entropy is utilized for automatic threshold selection. The experiments show that adaptive selection of the threshold for each image pair gives more promising results than a global threshold selection. The simulations also show that the morphological operations, erosion and dilation, improve results in object localization, although they are very simple. An important property of the tested method beside its

reliability, is its simplicity and low computational cost. In a controlled environment, the method can be used for moving object detection even with low-cost hardware.

The second method for moving object detection for stationary camera sequences makes use of the optical flow field induced between consecutive frames. Ideally, the optical flow field and the motion field are identical. The optical flow field is determined by a hierarchical version of Kanade Lucas Feature Tracker. The optical flow computation is applied to various kind of frames containing day camera frames, thermal camera frames and artificial frames and the results are quite satisfactory. In stationary camera sequences, the non-zero optical flow field points the moving object, since ideally the static background will not cause any change in scene intensity character. The experiments show that, the magnitude of the optical flow field between two consecutive frames taken from a stationary camera can be used for moving object detection after simple thresholding and morphological operations. The results of this detection scheme are reliable and satisfactory while the computation time is quite high compared to background elimination method. Therefore, using this method in stationary camera sequences can be seen unreasonable at the first glance. However, the optical flow computation is the base step for moving object detection in non-stationary camera sequences as well. Moreover, if the detection is conducted in an uncontrolled environment, where the camera makes various kinds of motions and the scene structure changes frequently within the sequence, the optical flow field should be calculated in any case. This calculation is also necessary for post-processing steps and for model discrimination. Therefore, using the optical flow field for stationary camera sequences should be a part of the algorithm integration process.

In non-stationary camera sequences, different algorithms are proposed for different camera motion characteristics and scene structures. For rotating cameras and/or planar scenes, 2D parametric registration of the dominant plane is used for moving object detection. Segmentation of the optical flow field is introduced in this step to distinguish between the multi-layer and single layer scenes. An affine parameters based method was used for optical flow field segmentation. The experiments on artificial and real data indicate that the affine parameter estimation is reliable, if the scene can be segmented properly. Proper segmentation is achieved if the internal parameters of the algorithm are optimized. It is observed

that this optimized values vary for different scenes, therefore, it is difficult to find global parameters for these values. However, manual optimization gives satisfactory results in moving object detection framework. In the planar scenes, segmentation process generally results in single planar motion, thus the registration is conducted with the affine parameters of this plane. The experiments indicate that 2D parametric registration is enough for most of the planar scenes to detect moving objects, and for all rotating-only camera scenes.

A modified version of the 2D parametric registration process is used for multi planar scenes. When the scene cannot be modeled by a single plane and the camera makes translational motion, the 2D parametric registration of the dominant plane is not sufficient. In this kind of scenes, the segmentation process results in two or three planes. Moving object detection is achieved by sequential registration of these planes. However, if the moving object is large, then it can also be detected as a plane. This scenario is not tested but some methods are proposed in the literature to solve this case.

When the depth variation within the scene is high and the scene cannot be modeled by layers, a parallax based approach is proposed for moving object detection. This method is based on the observation that when camera makes translational motion, the planes at different depths induce different 2D motions according to their depth and plane parameters. In addition, the static background move in an organized fashion, named parallax rigidity. The independently moving objects do not obey this parallax rigidity rule; therefore, parallax rigidity constraint is used for independently moving object detection. Before applying parallax rigidity, 2D parametric registration is required to remove the effects of rotation and create a translation-based framework. The experiments on artificial and real data show that the algorithm gives satisfactory results, when 2D registration is properly applied. Moreover, this parallax-based approach is also experimented in sequences where the parallax information is sparse with respect to the independent motion, and the results are still satisfactory. This property makes this algorithm more reliable compared to the classical plane+parallax based approaches where the epipole is used for independent motion detection. The reference point and registration plane selection is achieved automatically, as a difference from the literature.

In real image sequences it is possible to face with stationary, non-stationary, rotating-only, translating-only, rotating and translating cameras and planar, multi-planar and dense depth scenes. Therefore, none of the algorithms can be applied to the whole sequence. In order to make the detection problem more useful and applicable, an integration of the algorithms is strictly required. An optical flow and segmentation based unification is proposed in this thesis. The density and variance of the optical flow field is used to discriminate two cases: stationary camera and non-stationary camera. The statement is based on the fact that if the camera is not moving the optical flow field will be zero everywhere other than the moving object areas and if the camera is moving the optical flow field will be non-zero almost at every point. The experiments on real sequences indicate that this is a reliable reasoning and can be used for stationary/non-stationary discrimination. On the other hand, the model discrimination within non-stationary camera sequences is more complex. The optical flow field is segmented into regions and the number of reliable segments give valuable information about the scene and camera motion structure. If the number of reliable segments is 1, then 2D parametric registration approach will be sufficient for moving object detection. If the number of segments is between 2 and 4, then the scene is modeled by plane patches and sequential registration is applied. Finally, if the number of segments is greater than 4, parallax rigidity based approach is applied. This procedure is experimented with real sequences with a limited data set and gave satisfactory results. However, the segmentation process is very critical in this approach and it may require manual interaction for optimization. Therefore, more consistent unification approaches, which include temporal analysis, is required and is left as a future work.

The implicit usage of the ego-motion (camera motion) parameters in IMO detection gives us valuable information about navigation characteristics (3D velocity/acceleration and rotation angles) of the platform, where the camera is mounted. Therefore, the approaches mentioned in this thesis can be integrated to commonly-used navigation systems. For example, the integration of this work to ‘Inertial Navigation Systems’. via Kalman Filter, named ‘Image Aided Inertial Navigation via Kalman Filtering’ [29], will be an attractive issue to work on. This integration will be useful for autonomous navigation of mobile robots, long-term

navigation and terminal guidance of cruise missiles, sight lining of helicopter or other moving-platform weapon systems.

Moreover, the parallax based approach is considered to be an attractive framework for 3D reconstruction and scene analysis.

REFERENCES

[1] Jain, R. , Kasturi, R. , G.Schunk, B. , “Machine Vision”, McGRAW-HILL International Editions, 1995.

[2] Klaus, B. , Horn, P. , “Robot Vision”, MIT Press, 1986

[3] Tzannes, A.P. , Brooks, D.H. , “Point Target Detection in IR Image Sequence: A Hypothesis-Testing Approach Based on Target and Clutter Temporal Profile Modeling”, Opt. Eng., Vol 39, No. 8, pp. 2270-2278, August 2000

[4] Soni, T. , Zeidler, J.R. , Walter, H.K. , “Detection of Point Objects in Spatially Correlated Clutter Using Two Dimensional Adaptive Prediction Filtering”, http://citeseer.ist.psu.edu/21704.html, 1992

[5] Rosin, P.L. , Ellis, T. , “Image Difference Threshold Strategies and Shadow Detection”, British Machine Vision Conf., pp. 347-356, 1995

[6] Irani, M. , Anandan, P. , “A Unified Approach to Moving Object Detection in 2D and 3D Scenes”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 6, June 1998

[7] Irani, M. , Rousso, B. , Peleg, S. , “Robust Recovery of Egomotion” , Proc. of CAIP, pp. 371-378, 1993

[8] Strehl, A. , Aggarwal, J.K. , “MODEEP: a motion-based object detection and pose estimation method for airborne FLIR sequences”, Machine Vision and Applications, Springer-Verlag, 2000

[9] Adiv, G. , “Determining three-dimensional motion and structure from optical flow generated by several moving objects”, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. PAMI-7, No. 4, July 1985

[10] Piccardi, M. , “Background Subtraction Techniques: a review”, University of Technology Sydney, 2004

[11] Stauffer, C. , Grimson, W.E.L. , “Adaptive Background Mixture Models for Real-Time Tracking”, Proc. Of CVPR, pp. 246-252, 1999

[12] Borghys, D. , Verlinde, P., Perneel, C. , Acheroy, M. , “Multi-level Data Fusion for the Detection of Targets using Multi-Spectral Image Sequences”, Opt. Eng., Vol. 37, No. 2, pp. 477-484, February 1998

[13] Alatan, A. , “Lecture Notes of Robot Vision Class, EE701”, Electrical&Electronics Engineering Department, METU, 2002

[14] Malvika, R. , “Motion Analysis, Course notes for CS558”, School of Computer Science, McGill University

[15] Sankur, B. , Sezgin, M. , “A survey over Image Thresholding Techniques and Quantitative Performance Evaluation”, Journal of Electronic Imaging, Vol. 13, pp. 146-165, January 2004

[16] Kapur, J.N. , Sahoo, P.K. , Wong, A.K.C. , “A new method for Gray-Level Picture Thresholding Using the Entropy of the Histogram”, Computer Vision, Graphics and Image Processing 29, 273-285, 1985

[17] Lucas, B. , Kanade, T. , “An iterative image registration technique with an application to stereo vision”, Proc. Image Understanding Workshop, 1981

[18] Bouget, J.Y. , “Pyramidal Implementation of the Lucas Kanade Feature Tracker, Description of the Algorithm”, Intel Corporation, 1999

[19] Irani, M. , Rousso, B. , Peleg,S. , “Computing Occluding and Transparent Motions”, Int’l J. of Computer Vision, Vol. 12, p. 5-16, Feb 1994

[20] Wang, J.Y.A. , Adelson, E.H. , “Representing Moving Images with Layers”, IEEE Transactions on Image Processing Special Issue: Image Sequence Compression, Vol. 3, No. 5, p. 625-638, September 1994

[21] Lourakis, M.I.A. , Argyros, A.A., Orphanoudakis, S.C. , “Independent 3D Motion Detection Using Residual Parallax Normal Flow Fields”, ICCV, pp. 1012-1017, 1998

[22] Ben-Ezra, M. , Peleg, S. , Rousso, B. , “Motion Segmentation Using Convergence Properties”, ARPA Image Understanding Workshop, November 1994

[23] Irani, M. , Rousso, B. , Peleg, S. , “Recovery of Ego-Motion Using Region Alignment”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 3, p. 268-272, March 1997

[24] Irani, M. , Anandan, P. , Cohen, M. , “Direct Recovery of Planar-Parallax from Multiple Frames”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, November 2002

[25] Harris, C., Stephens, M. , “A combined Corner and Edge Detector”, Plesey Research Roke Manor, UK, The Plessey Coompany plc. , 1988

[26] Borshukov, G.D. , Bozdagi, G. , Altunbasak, Y. , Tekalp, M. , “Motion Segmentation by Multistage Affine Classification”, IEEE Transactions on Image Processing, Vol. 6, No. 11, November 1997

[27] Qian, G. , Chellappa, R. , “Moving Targets Detection using Sequential Importance Sampling”, Proc. of Int’l Conf. on Computer Vision, Vol 2, pp. 614-621, July 2001

[28] Irani, M. , Anandan, P. , “All About Direct Methods”, Vision Algorithms:Theory&Practice, Springer-Verlag, 1999

[29] Giebner, M. G. , “Tightly-Coupled Image-Aided Inertial Navigation System via a Kalman Filter”, AIR FORCE Inst. Of Tech. Wright-Patterson School of Enginerring and Management, Master’s Thesis, March 2003

[30] Duda, R.O. , Hart, P.E. , “Pattern Classification and Scene Analysis”, John Wiley, 1973

[31] http://mmrg.eee.metu.edu.tr/imageutil
http://mmrg.eee.metu.edu.tr/videoutil