Monocular and Binocular Visual Tracking

Gouda Ismail Salama

Dissertation Submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial Fulfillment of the Requirements of the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Amos L. Abbott, Chairman
Hugh F. VanLandingham
John S. Bay
John W. Roach
Aicha Elshabini

December 9, 1999
Blacksburg, Virginia

Keywords: Monocular Tracking, Binocular Tracking, Active Vision, Image Matching,
Low-level Vision, Moment Invariants, Adaptive Window Selection.

Monocular and Binocular Visual Tracking

Gouda Ishmail Salama : Abstract

Visual tracking is one of the most important applications of computer vision. Several tracking systems have been developed which either focus mainly on the tracking of targets moving on a plane, or attempt to reduce the 3-dimensional tracking problem to the tracking of a set of characteristic points of the target. These approaches are seriously handicapped in complex visual situations, particularly those involving significant perspective, textures, repeating patterns, or occlusion.

This dissertation describes a new approach to visual tracking for monocular and binocular image sequences, and for both passive and active cameras. The method combines Kalman-type prediction with steepest-descent search for correspondences, using 2-dimensional affine mappings between images. This approach differs significantly from many recent tracking systems, which emphasize the recovery of 3-dimensional motion and/or structure of objects in the scene. We argue that 2-dimensional area-based matching is sufficient in many situations of interest, and we present experimental results with real image sequences to illustrate the efficacy of this approach.

Image matching between two images is a simple one to one mapping, if there is no occlusion. In the presence of occlusion wrong matching is inevitable. Few approaches have been developed to address this issue. This dissertation considers the effect of occlusion on tracking a moving object for both monocular and binocular image sequences. The visual tracking system described here attempts to detect occlusion based on the residual error computed by the matching method. If the residual matching error exceeds a user-defined threshold, this means that the tracked object may be occluded by another object. When occlusion is detected, tracking continues with the predicted locations based on Kalman filtering. This serves as a predictor of the target position until it reemerges from the occlusion again. Although the method uses a constant image velocity Kalman filtering, it has been shown to function reasonably well in a non-constant velocity situation. Experimental results show that tracking can be maintained during periods of substantial occlusion.

The area-based approach to image matching often involves correlation-based comparisons between images, and this requires the specification of a size for the correlation windows. Accordingly, a new approach based on moment invariants was developed to select window size adaptively. This approach is based on the sudden increasing or decreasing in the first Maitra moment invariant. We applied a robust regression model to smooth the first Maitra moment invariant to make the method robust against noise.

This dissertation also considers the effect of spatial quantization on several moment invariants. Of particular interest are the affine moment invariants, which have emerged, in recent years as a useful tool for image reconstruction, image registration, and recognition of deformed objects. Traditional analysis assumes moments and moment invariants for images that are defined in the *continuous* domain. Quantization of the image plane is necessary, because otherwise the image cannot be processed digitally. Image acquisition by a digital system imposes spatial and intensity quantization that, in turn, introduce errors into moment and invariant computations. This dissertation also derives expressions for quantization-induced error in several important cases. Although it considers spatial quantization only, this represents an important extension of work by other researchers.

A mathematical theory for a visual tracking approach of a moving object is presented in this dissertation. This approach can track a moving object in an image sequence where the camera is passive, and when the camera is actively controlled. The algorithm used here is computationally cheap and suitable for real-time implementation. We implemented the proposed method on an active vision system, and carried out experiments of monocular and binocular tracking for various kinds of objects in different environments. These experiments demonstrated that very good performance using real images for fairly complicated situations.

Acknowledgments

I would like to thank my advisor, Prof. A. L. Abbott, for his continuous guidance and support throughout this work despite his busy schedule. His good spirits and encouragement had a great impact on me.

I would like to thank my committee members Prof. J. S. Bay, Prof. H. VanLandingham, Prof. J. W. Roach, and Prof. Aicha Elshabini for their help, comments, and availability.

Thanks also go to my family members and my friends. Finally thanks also go to my kids and my wife who always stood shoulder to shoulder with me in my difficult time.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The ability to perform visual tracking within a dynamic world is of fundamental importance for many higher-level tasks. Tracking can be considered a primitive capability that facilitates such applications as surveillance, active control of fixating cameras, and autonomous navigation through complex scenes.

The difficulty in visual tracking is the potential variability in the images of an object over time. This variability stems from three principal sources: variation in target deformations, variation in illumination, and partial or full occlusion of the target. When ignored, any one of these three sources of variability is enough to cause a tracking algorithm to lose its target. There are two principal challenges for visual tracking: to develop good models of image variability, and to design effective and efficient tracking algorithms that use these models.

There are two main approaches to object tracking. The first approach derives an optical flow field, or dense motion field, for the sequence, and then analyzes the structure of the flow field to infer structure, motion, or both, for the objects in the image. The second approach is based on the correspondence of discrete features of an object in one image with those features in a subsequent image. This approach typically uses template matching or other search techniques to determine the locations of features in an image, and then infers object motion from these correspondences. There are two main advantages to correspondence-based feature tracking over the image flow approach. The computational burden is reduced significantly since the velocity of only a few points in the image is measured. Also, the correspondence-based feature tracking makes it possible to choose salient features thus increasing the probability of proper feature velocity measurement.

This dissertation describes a new tracking methodology that emphasizes both prediction and measurement. The approach has been tested through the use of a region-based, steepest-descent matching technique that has been used successfully for monocular tracking [Shi94]. Early experiments with this technique showed the need for a good initial estimate in order to locate the correct correspondence. To provide this estimate, we have developed a system that combines Kalman-type prediction with this matching approach, and we have extended it to accommodate binocular image sequences. The system operates using only 2-dimensional (2D) information. It does not require a priori information about the imaging geometry, nor about the scene or object being tracked, and does not attempt to recover 3-dimensional (3D) motion or structure information. The resulting system has been successfully tested with several real image sequences.

Tracking requires establishing correspondences of features or markings between successive image frames. This process is referred to as the correspondence problem. Unfortunately, the problem of locating corresponding points for a given target is very difficult in general. Most existing methods utilize region-based similarity measures to locate candidate matches between images. These methods work well in some cases, but do not perform well for complex scenes, particularly those involving significant perspective or occlusion. The correspondence search is also difficult in low-contrast (textureless) regions, and when repeating patterns are present.

Occlusion detection is an important problem in 3D-computer vision, especially when multiple views are used, such as stereo vision. Occlusion occurs where there is a depth discontinuity, and part of object is not visible to both cameras. This causes binocular rivalry in stereo images.

Without the occurrence of occluded regions in the images, image matching is a simple one to one mapping of the two images. Constraints such as uniqueness, smoothness or ordering of the disparity, which are utilized to simplify the matching process, are invalid assumptions in occluded regions. One way to avoid correspondence errors in occluded areas is a bi-directional or dual matching process [Geig92, Hoff89, Jone92, and Luo88]. The visual tracking method described here is improved to detect the occlusion based on the residual error computed by the matching approach. If the residual

matching error exceeded a user-defined threshold, this means that the tracked object may be occluded with another object. When occlusion is detected, tracking continues with the predicted locations based on Kalman filtering. This serves as a predictor of the target location until it reemerges from the occlusion again. Although the method uses a constant image velocity Kalman filtering, it has been shown to work well in a non-constant velocity situation.

With a few exceptions [Okut92, Abbo95], most systems utilize image windows of a fixed, predetermined size for similarity estimation. We found that the search technique used here is very sensitive to the choice of window size, and we have developed a new approach based on the moment invariants to select that size. Intuitively, a window that is too small will tend be distracted by texture primitives, or (at the other extreme) it may enclose a featureless region which is unsuited for area-based comparisons. For a window that is too large, the similarity measurements may give incorrect matches due to repeating patterns, occlusion, and perspective differences. This dissertation presents a novel approach to the automatic selection of window size, based on image invariants.

Moments and moment invariants are often used for image reconstruction, image registration, and recognition. Traditional analysis assumes moment invariants that are defined in the continuous domain. Moment "invariants" that are computed from discretized images are only approximations to the true invariants because of errors that result from spatial quantization. This dissertation also presents an analysis of quantization-induced error on (2-dimensional) Hu moment invariants and affine moment invariants, and on invariants derived from (1-dimensional) contour moment invariants [Sala98a]. Error bounds are given in several cases. We consider an extension of the Hu moment invariants, as introduced by Maitra [Mait79]. In addition to the invariance properties for Hu moment invariants, these are additionally invariant to scale changes in image intensity.

There are many applications for the visual tracking system presented in this dissertation. Its ability to keep the moving object near the center of an image frame is beneficial several fields. The tracking system using active camera has advantages in indoor or outdoor surveillance, in automatic video recording and video teleconferencing, and in manufacturing environment.

1.2 Computer Vision Paradigms

Three different vision paradigms [Blak92] are usually distinguished in the area of computer vision. The first paradigm is passive vision and is limited to the study of visual information. It corresponds to either one view (monocular scene analysis), two or three views at different locations in space (stereo), or two or three views at different times (motion analysis). The second paradigm is dynamic vision, which corresponds to the study of visual information in an unbounded sequence of views. The third paradigm, active vision, is the control of the optics and the mechanical structure of cameras to simplify the processing for computer vision.

Active fixation is an important area of research. In the monocular case, fixation means controlling the camera movements to keep the target in the central field of view. In the binocular case, fixation means controlling two cameras so that their optical axes intersect at a desired point in a three-dimensional scene

Figure 1.1 illustrates the top view of an active stereo fixation system [Lin96], where S represents the horizontal distance between two cameras, α and β are rotation angles of the left and right cameras respectively, C is the fixation point of the two cameras in a three-dimensional scene, and γ is the angle of vergence. The depth d of the point C can be computed using

$$d = \frac{S}{\tan\alpha + \tan\beta} \tag{1.1}$$

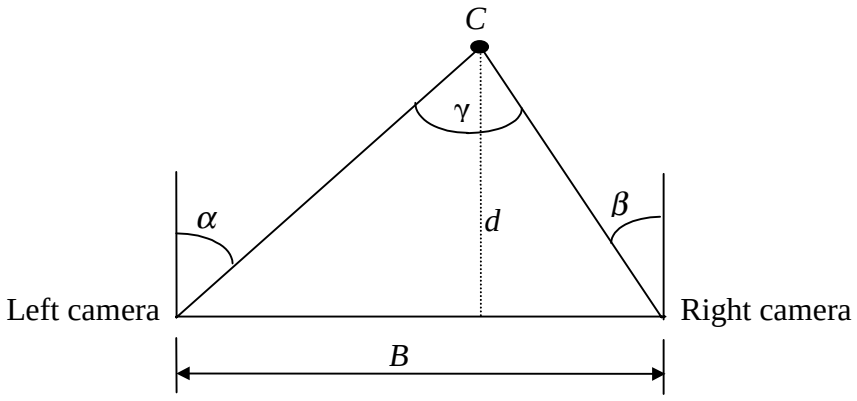


Figure 1.1. Fixating camera system. The result of fixating a target is to keep the cameras fixed on the desired target.

Zheng [Zhen94] has discussed the importance of stereo fixation, which is the first step in a tracking system. Acquiring a correct initial target point before tracking is actually very important for stereo tracking systems.

1.3 Monocular Tracking Definition

Tracking is concerned with the visual pursuit of an object of interest that moves randomly. As objects move relative to a camera, the patterns of image intensities change in a complex way. In general, $I(x,y,t)$ can represent an image sequence, where x and y are spatial coordinates in the image and t is the time variable. Then using an assumption given by [Shi94], image intensity I at time $t+\tau$ approximately satisfy the following model:

$$I(x, y, t+\tau) \approx I(x+u(x, y, t, \tau), y+v(x, y, t, \tau), t) \tag{1.2}$$

where $u(x, y, t, \tau)$ and $v(x, y, t, \tau)$ are the displacements of the point (x, y) in the x and y directions, respectively, between time t and $t+\tau$. Equation (1.2) means that for an image sequence of a moving object, a pixel at (x, y) in one image can be found at $(x+u, y+v)$ in the subsequent image if the displacements u and v can be determined properly.

Given two successive frames in an image sequence $I(x,y,t)$ and a window in the first frame at time t, *monocular tracking* using a passive camera in this dissertation means repeatedly detecting the position of a desired target based on a matching process.

The monocular tracking vision system using an active camera is shown in Figure 1.2. At every sampling instant, the moving camera acquires a new image, the new target position is computed using the extracted affine motion parameters and the control commands are generated to drive the pan-tilt unit to keep the target near the center of an image frame.

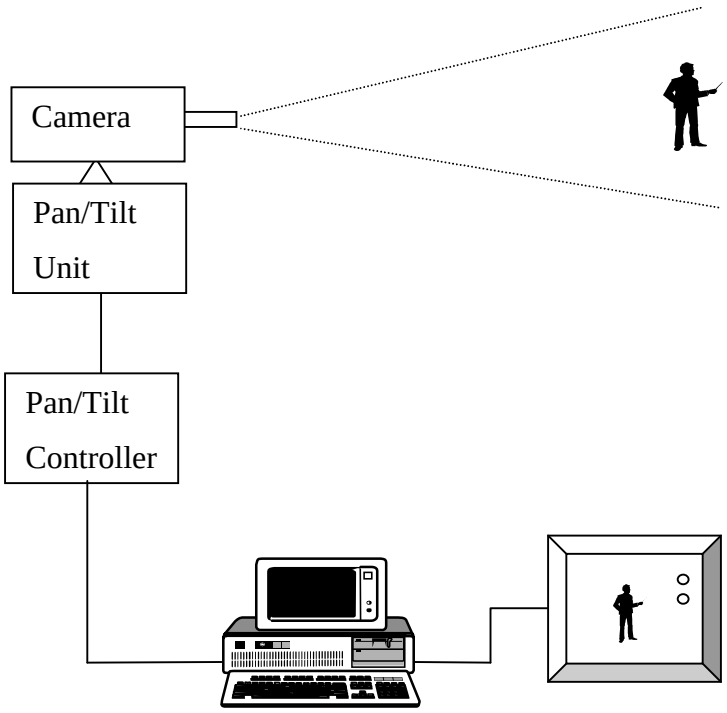


Figure 1.2 Monocular tracking system

1.4 Binocular Tracking Definition

Given two stereo image pairs $(I_L,\ I_R)$ and $(J_L,\ J_R)$ in a motion sequence and a window in image I_L. Binocular tracking using passive cameras in this dissertation means repeatedly following the desired object based on the extracted affine motion parameters that result from two matching – one between the first and the current frame from the dominant (left) camera and a second match between the first frame from the left camera and the current frame in the right camera.

The binocular tracking system using active cameras is shown in Figure 1.3. At every sampling instant, the stereo cameras acquire a new stereo image pair for the moving object. Two matching steps are performed – one between the first and the current frame from the dominant (left) camera and a second match between the first frame from the left camera and the current frame in the right camera. The output of matching is used to estimate the new target positions in the next stereo pair. Then the control commands

are generated to drive the pan-tilt unit of the two cameras so that their optical axes intersect at the target.

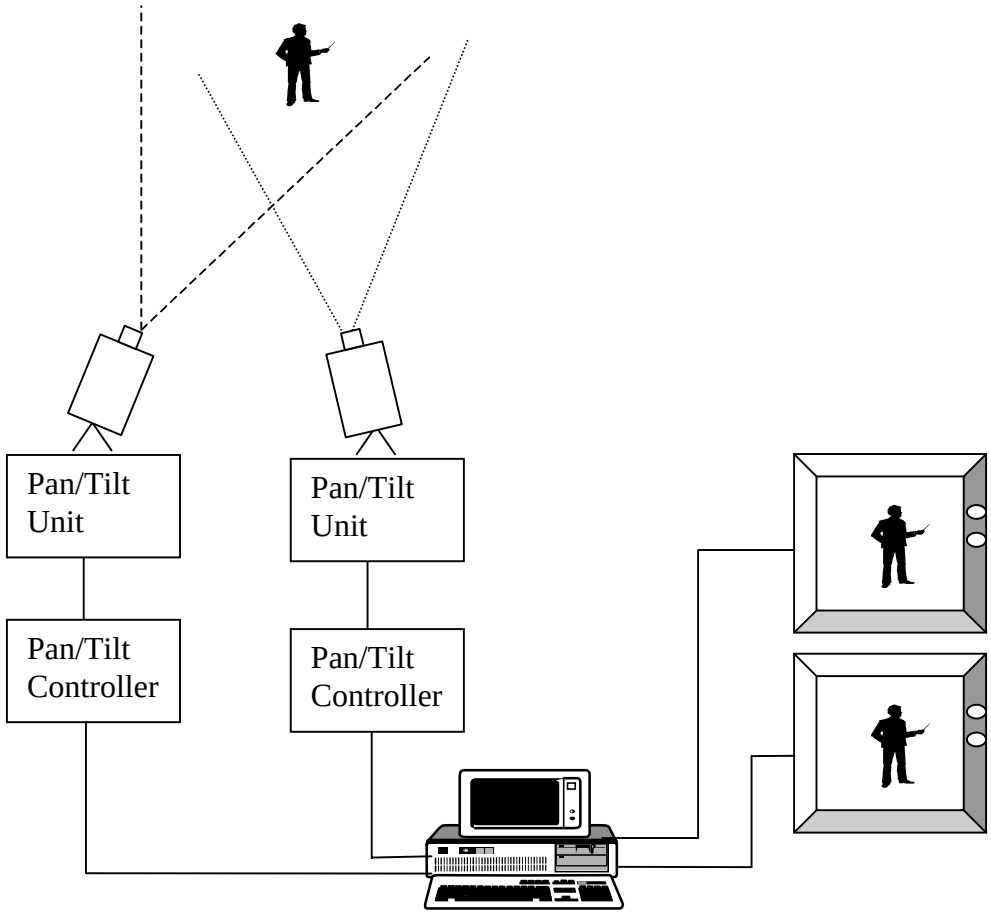


Figure 1.3 Binocular tracking system

1.5 Problem Statement

This dissertation addresses the problem of visual tracking for monocular and binocular images, and for both passive and active cameras.

In the monocular case, for a passive camera, the problem is to take an image sequence $I(x,y,t)$, along with an initial target location (x,y) and to determine new target locations $(x+u,y+v)$ over time. In the binocular case, for passive cameras, the problem is to determine the affine motion parameters between two successive stereo image pair.

When active (motorized) camera is used for the either case, an additional problem is to determine new tilt/pan steps, in an attempt to keep the target centered in the image(s).

A part of the problem in area-based image matching applications is the selection of the window size to achieve a good matching and to achieve a good tracking performance when occlusion occurs.

1.6 Contributions of this Work

In 1994, Shi and Tomasi [Shi94] published a technique for seeking area-based image correspondences using a Newton-Raphson style search for 2D affine deformation parameters. They demonstrated its effectiveness for monocular tracking, in which only small interframe motion is present. One contribution of this work is that we have adapted the Shi-Tomasi search technique so that 1) it works for both monocular and binocular image sequences, and for both passive and active cameras and 2) it is not limited to small frame-to-frame movements. The search method requires a good initial estimate of the match location, and our technique provides that by using a Kalman filter. The system therefore represents a novel combination of Kalman-type prediction with steepest-descent search for image correspondences. The fact that 2D parameters are found means that difficult 3D recovery is not required, and the fact that 2D affine parameters are found means that the system can yield good performance in the presence of changes in perspective.

A major problem for area-based tracking is object occlusion. In general a moving object may be occluded for any number of reasons. Other objects, moving or not, may obscure the desired object, or the desired object exits the field of view of the camera. When the desired object is occluded, the template will not match any area of the image very well and the matching residue is expected to be large. The visual tracking system proposed here was extended to detect the occlusion based on the residual error computed by the matching method. If the residual matching error exceeded a user-defined threshold, this means that the tracked object may be occluded with another object. When occlusion is detected, tracking continues with the predicted locations based on Kalman

filtering. This serves as a predictor of the target location until it reemerges from the occlusion again.

This dissertation also describes a new approach to automatic window-size selection based on moment invariants of the images being considered for area-based image matching applications. This is particularly important for such tasks as visual tracking and stereo matching, both of which require a search for corresponding points within different images. The area-based approach to image matching often involves correlation-based comparisons between images, and this in turn requires that a finite region of interest be specified for the correlation windows. Typically, window sizes are determined based on the sudden increasing or decreasing in the first Maitra moment invariant. We applied a robust regression model to smooth the first Maitra moment invariant to make the method robust against the noise.

Another contribution is a theoretical analysis of quantization-induced error for several important cases of moment invariants. Analytical expressions and experimental results have been computed for rectangular images undergoing changes in scale, rotation, and skew. As expected, the error due to quantization decreases as the image size increases relative to pixel size. The quantization error of affine moment invariants (AMIs) tends to be much smaller than Hu moment invariants [Hu62]. In general, quantization errors are not necessarily periodic with respect to rotation or skew changes. This work is the first to consider quantization error for affine moment invariants, and represents a significant extension of previous work for the case of Hu moment invariants.

1.7 Dissertation Outline

The main objective of Chapter 1 is to introduce and explain the problem of visual tracking.

Chapter 2 reviews the related literature, beginning with the previous work on motion detection and visual tracking in both monocular and binocular image sequences. After briefly reviewing the previous work related to stereo matching in binocular vision, this chapter also gives an overview of the active vision paradigm and some of its applications.

Chapter 3 describes the hardware configuration of the proposed active tracking system for monocular and binocular image sequences.

Chapter 4 contains the principal novel work in this dissertation, giving details of the new approach to visual tracking for monocular image sequences. The method describes a steepest-descent approach for finding corresponding points between two images. The system seeks 2D affine transformations between two images. Because this search requires a good starting point, it has been augmented with Kalman-type prediction. Also, this chapter discusses a method of handling the occlusion of the moving object in the monocular image sequence.

Chapter 5 describes the details of the new approach to binocular visual tracking system for image sequences. The monocular tracking algorithm is extended to work for binocular image sequences. Also, this chapter discusses a method of handling the occlusion of the moving object in the binocular image sequence.

In Chapter 6, a novel approach has been developed that can automatically select a window size for a given target in a scene based on image moment invariants. Also, an analysis of the quantization-induced error on (two-dimensional) Hu moment invariants and affine moment invariants, and on invariants derived from (one-dimensional) contour moments have been presented. Error bounds are given in several cases.

In Chapter 7, experimental results show the effect of window size selection on the tracking for both monocular and binocular image sequences. Also, the performance of the proposed tracking system is illustrated by presenting results in several situations.

Chapter 8 contains concluding remarks about the work in this dissertation and explains several areas of future work that would enable the system to become more fully autonomous.

This dissertation includes Appendices A and B. Appendix A discusses the 2D affine transformation. Appendix B describes the main equations of the Kalman filter.

Chapter 2

Background and Literature Review

2.1 Introduction

This chapter reviews previous work involving monocular tracking methods and binocular tracking methods and briefly comments on the differences between our current tracking system and other different types of systems implemented elsewhere. Also, it reviews the image matching approaches involving feature-based and area-based algorithms with a comparative analysis. Furthermore, the revision of previous work involving the methods used to select the window size to aid in matching. Finally, this chapter presents the active vision paradigm and some of its applications.

2.2 Overview of Motion Detection and Visual Tracking

Motion detection and tracking are becoming increasingly recognized as important capabilities in any vision system designed to operate in an uncontrolled environment. One application of motion analysis [Miti94] includes video compression, which attempts to exploit temporal redundancy by reducing transmission rates while maintaining image quality. Another application is mobile robotics, which deals with the challenges of robot positioning, obstacle detection and avoidance, tracking of moving objects, etc. Satellite imagery is a third example in which image motion analysis is used to measure cloud movements and establish wind maps. Still another application of image motion is in the biomedical field. Echography, numeric radiography, and anciangiograms all have been the source of tasks involving image motion processing. Finally, surveillance applications include monitoring of urban and road traffic, and protection of sites from intrusion.

Military applications include target tracking and autonomous navigation of various objects such as vehicles. Tracking of moving objects, for measuring motion-parameters and obtaining a visual record of the object in various stages of motion, is an important in many fields spanning both military and industrial applications. However,

traditional methods of tracking, such as radar tracking, do not provide any photographic record of the target being tracked and are also known to interfere with operating environment due to transmission of high power electromagnetic waves. Increased availability of high speed image-processing hardware and efficient algorithms have now made it possible to attempt designing a real-time video tracking system capable of tracking fast moving objects at close to medium ranges [Wang98].

In general, there are two approaches to tracking and they are fundamentally different. One of them recognition-based tracking [Aloi91, Bray90, Genn82, Wilc87, and Wang95] and the other is motion-based tracking [Cai95, Mae96, Lee95, Reid96, and Okad96]. In recognition-based tracking, the object is recognized in successive images and its position extracted. One advantage of this tracking method is that it can be achieved in three-dimensions. Also, the translation and rotation of the object can be estimated. The obvious disadvantage is that only a recognizable object can be tracked. Object recognition is a high-level operation that can be computationally very costly to perform. Thus, the efficiency of the recognition method, as well as the types of recognizable objects limits the performance of the tracking system. Motion-based tracking systems rely entirely on motion detection to detect the moving object. They have the advantage of being able to track any moving object regardless of size or shape, and so are more suited for our systems.

The problem of extracting motion information from a sequence of images has received much attention. Several techniques have been used to obtain this information. The Kalman filter is applied to estimate position and motion parameters from a sequence of moving object images. Roach and Aggarwal [Roac80] used the modified Levenberg-Marquardt finite difference algorithm to solve the simultaneous non-linear equations that arise from the description of body movements. In Roach and Aggarwal [Roac79], tracking of a rigid convex polyhedral is considered. For each new frame the objects are detected by segmentation. The centroid of each object is computed and from its displacement an estimate of the translation velocity of the object is derived. If the object is partially occluded or uncertainty in the scene prevents segmentation, then individual features such as corners are tracked.

Two main approaches for motion field estimation can be identified, that is, the gradient-based and the matching-based approach. The former is based on the solution of partial differential equations [Horn81] and leads to the estimation of an approximated 2D-motion field called optical flow field. The estimation of motion fields by means of the gradient-based approach is affected errors which are due to: (i) the differential model itself (e.g., image brightness discontinuities due to noise, too crisp patterns, etc.) and (ii) the lack of stable correspondences between the vectors of the motion field and 3D points of the observed 3D moving object. The matching-based approach is based on establishing correspondences of features or markings between successive image frames. By tracking these features, an inter-frame correspondence is searched to estimate the motion (displacement) of selected features on the image plane. The estimation of a motion field by means of the matching-based approach of image patterns is affected by the accumulation error (mainly due to discretization), and it is very sensitive to noise [Li93].

In this dissertation, a new method for tracking moving objects is presented. It belongs to the class of matching-based algorithms.

2.3 Monocular Tracking Approaches

Various methods have been used for monocular tracking. Some are based on the intensity differences between frames [Yach81, Cai95, Murr94, and Jang97]. For example, Jang et al. [Jang97] described a real-time monocular tracking system. This system detects an object entering the field of view of a camera and executes tracking of the detected object by controlling a servo device in such a way that a target always lies at the center of an image frame. Kalman filter state parameters are used to reduce search areas for model matching and to control the servo device. Cai et al. [Cai95] presented an approach to track human motion in a sequence of monocular images. The process consists of detecting motion, segmenting moving objects by recovering the background and finally tracking the objects of interest. The main assumptions are small image motion, fixed viewing system, and constant velocity. Murray and Basu [Murr94] described a method of tracking a moving object in real time using active camera mounted on a pan/tilt unit. The motion detection module computes moving objects independently with a low frame rate using knowledge about the camera's motion.

The intensity-difference-based method cannot be applied to cases in which the camera moves because the background changes. A correlation-based method [Inou92, Mont94, Dell97, Frau90, Papa93, and Hube95] works when camera moves. In the correlation-based tracking method, the image path corresponding to the object being tracked is searched for in each subsequent image using a correlation technique. The position of best correlation is chosen as the position to which the camera is to be driven. Papanikolopoulos et al. [Papa93] used the simplest method of searching for the original template image in each subsequent image and then drove the camera to fixate the position of highest correlation. Huber and Kortencamp [Hube95] proposed a method to avoid the problems caused by rotation by centering the new image template on the centroid of matched features rather than on the position given by the largest correlation value.

Success of correlation-based tracking depends largely on choosing suitable window sizes and thus transferring the proper reference image to the next frame. In correlation-based tracking, very few studies based on estimating an image window size are often reported in the technical literature. To adapt the reference area automatically, Montera et al. [Mont94] determined an object region through expanding the inner point of an object to the outer point in the image. To yield the boundary of the object, they searched for the areas where pixel values vary from above the threshold to below the threshold. This method is difficult to apply to the tracking of non-homogeneous cluttered background and large objects with internal edges. Consequently, Sung et al. [Sung97] presented an image tracking architecture employing the centroid compensation to enhance the stability of correlation-based tracking in cluttered surroundings. Centroid compensation within a window provides the tracker with the capability of correcting the correlation-based tracking method to the real center of an object when weak clutter components exist.

As shown above, correlation-based object tracking methods can be difficult if the appearance of the object changes. An optical flow based method [Mae96, Yama95, Lee95, Okad96, and Broi90] can be applied to such a case. For example, Mae et al. [Mae96] described a real-time monocular object tracking method that extracts optical flows from a sequence of images. This method assumes that pixels corresponding to the same object have similar flow vectors. Broida et al. [Broi90] introduced a recursive

method of estimating the 3D kinematics and structure of rigid moving images. The recursive estimation was done using an iterated extended Kalman filter method. Lee et al. [Lee95] developed a new monocular tracking algorithm to track a target moving in 3D. The algorithm estimates position, linear and angular velocity, and orientation of the viewed surface in real-time using monocular visual sensory feedback. Tracking results were quite stables using a Kalman filter, even in the presence of system and sensor noise.

The optical flow based method cannot discriminate objects with similar flow. A method to track a moving object using optical flow and depth is proposed [Okad96] to solve this problem. The velocity and the depth of the target object are estimated from histograms of the velocity and of the disparity. The target region is estimated as the region that has velocity similar to the predicted target velocity in the predicted target region. Occlusion of the target is detected from an abrupt disparity change in the target region. The method proposed by Okada et al. [Okad96] did not work well when the target had regions with little contrast because neither optical flow nor depth could be obtained.

Features often used for tracking may be classified as region-based, contour-based, or point-based [Reid93]. A region-based method [Basc95, Bade97, and Fuh93] is useful where optical flow and depth cannot be obtained. Region-based motion estimation of 2-D image sequence motion using correlation has several advantages when compared with other approaches such as optical flow [Teka95, Dufa95]. It tends to be more robust in the presence of noise. If the region is selected carefully this method can also enable aperture type problems to be overcome. A region-based approach also gives a direct and a concise description of the global motion field using relatively few elements. This is especially beneficial for any post-processing such as segmentation or tracking. However, the region-based approaches cannot discriminate the target from other objects with similar brightness. A method using optical flow and uniform brightness regions was proposed [Tsuy98] to solve this problem.

Point-based features such as corners are simple to compute, but can be quite difficult to detect reliably. To overcome this, clusters of intensity corners are tracked in [Reid93], whereas multiple hypotheses are formulated and processed in [Cox96]. The first corner-based tracking method for an active platform was presented by Reid and Murray [Reid93, Reid96] which uses a constant image velocity Kalman filter to establish

the points of correspondence across frames. Wang et al. [Wang95] described an improved approach in the implementation of a monocular object tracking system in real time. Visual processing is performed in a smaller field of view extracted from an image. The visual processing method makes use of a corner detector to identify corners of the object appearing in the fovea window.

A contour-based object tracker approach [Denz97, Hutt94, and Mae96] allows contour extraction and tracking within the image frame rate on general purpose computer architectures. Contour-based features include line segments and image curves. These features have been used in several tracking systems, often with emphasis on the recovery of 3D structure. For example, Mae et al. [Mae96] described a monocular tracker for multiple moving objects based on contour. The contour is determined by using optical flow and edges in a long sequence of images. This method can determine the occlusion relation of two overlapping objects by checking if edges exist on the predicted contour. Denzler and Niemann [Denz97] desired a complete system for data-driven monocular tracking of moving objects in natural scenes. The tracking of the object is based on contour extraction. Also, Huttenlocher and Jaquith [Hutt94] presented a method for detecting moving objects in a monocular image sequence that was obtained using a moving camera. The method was not based on a local computation of image change such as optical flow or the estimation of point motions along edge contours. The motion of the edge contours was based on the spatial proximity defined by a Voroni diagram distance transform of the edge.

A tracking algorithm based on 3D affine structure [Tsui97, Mank97, Reid93, and Wang95] significantly improves the performance over just Kalman filter based tracking with little additional computational overhead. For example, Manku et al. [Mank97] presented a tracking algorithm based on affine structure. They used point correspondences obtained using Reid et al. [Reid93, Reid96] to compute the 3D affine structure of the fixation point and the affine camera projection equations in each frame. Manku et al. [Mank97] then used these structures and projection equations to localize the fixation point in each frame.

2.4 Binocular Tracking Approaches

Binocular tracking involves a search for correspondences temporally, as well as between stereo image pairs. Area-based and feature-based techniques have been used to detect stereo correspondences using optical flow [Pan94, Barr95] or intensity corners [Bhat97, Reid93, Reid96, Fair95] in the matching process. These techniques estimate 3D motion parameters of an object relative to the cameras. For example, Hung et al. [Hung95] presented a 3D predictive visual tracker for multiple moving rigid objects even when the stereo cameras are moving. The 3D features are computed from a sequence of stereo images by combining two 2D temporal matching modules and one stereo correspondence. Falkenhagen [Falk95] presented an approach to estimate the depth map for a 3D object from stereoscopic image sequences. This approach calculates a current depth map for each stereoscopic image-pair by averaging two independently estimated depth maps, the temporally independent depth map and the motion compensated depth map. Motion of the objects is estimated from the current image pair simultaneously using a gradient-based approach.

Lee and Kay [Lee91] presented a Kalman filter approach for accurately estimating 3D orientation of a moving object from a sequence of stereo images. The basic assumption of this approach is that stereo and temporal matches had been done in advance. No experimental results were presented. Shieh et al. [Shie92] reported an approach to recover motion parameters from a stereo image sequence. Zhang and Faugeras [Zhan92] proposed an approach using 3D line segments obtained from stereo images. The motion estimation problem was then formulated as an extended Kalman filtering problem. Tracking was performed using prediction-matching update looping. Using this approach multiple matches could be handled.

Barron and Eagleson [Barr95] used the left and right monocular motion and structural parameters of two stereo image sequences (direction of translation, relative depth, observer rotation and rotational acceleration) to compute absolute depth, absolute translation and absolute translation acceleration for each pair of left and right images. No features or image velocities have to be matched between left and right frames, and there is no need for a priori surface structure model. Yi et al. [Yi95] described an approach to determine 3D relative motion and position of separate features for obstacle avoidance and

3D tracking of a mobile robot. A Kalman filter was used to reduce image measurement noise and to provide optimal 3D motion and position when a sequence of stereo images was used.

Homainejad and Shortis [Homa95] presented a stereo vision system for tracking a dynamic object. The size, shape, and behavior of the object in the scene, the precise position of CCD cameras and the performance of the system were fundamental parameters to achieving appropriate precision and reliability. Pan et al. [Pan94] proposed a Kalman filter based algorithm for 3D-motion estimation from a stereo image sequence using a unified temporal-spatial optical flow field. The motion estimation problem was then formulated as an extended as a Kalman filtering problem. Determination of covariance matrices for system and measurement noise was briefly addressed.

2.5 Comparison with other Tracking Systems

In this section, we will briefly comment on the differences between our current tracking system and other different types of systems discussed in sections 2.3 and 2.4. Several tracking systems have been developed which either focus mainly on the tracking of targets moving on a plane, or attempt to reduce the 3-dimensional tracking problem to the tracking of a set of characteristic points of the target. These approaches are seriously handicapped in complex visual situations, particularly those involving significant perspective, textures, repeating patterns, or occlusion.

Our monocular and binocular visual tracking method combines Kalman-type prediction with steepest-descent search for correspondences, using 2-dimensional affine mappings between images. This approach differs significantly from many recent tracking systems, which emphasize the recovery of 3-dimensional motion and/or structure of objects in the scene. We argue that 2-dimensional area-based matching is sufficient in many situations of interest, and we present experimental results with real image sequences to illustrate the efficacy of this approach, including its potential for real-time implementation. Also, our tracking system is able to handle many situations that involve occlusions, significant perspective, textures, or repeating patterns. The resulting system has been successfully tested with several real monocular and binocular image sequences in different situations.

2.6 Image Matching

2.6.1 Introduction

Solutions for image matching were first suggested in the late 1950s by Hobrough [Hobr59]. Since then, interest in image matching has increased steadily. The problem remains current because of the information content of the most elementary primitive in the input data set, the pixel. Since, the value of the pixel can change due to noise, and can be confused with adjacent pixels. Ambiguous solutions may occur if image matching is based on local information only. Because of these problems, image matching on the basis of single pixels is certainly impossible, but windows that contain enough textures can be used.

The key problem in stereo vision systems is to find the corresponding points in the left and right images. A survey of stereo vision techniques is given in [Koos93]. Several factors make the correspondence problem difficult. These factors are occlusions, (i.e., points in one image with no corresponding point in the other), perspective distortions, and repeating patterns.

Stereo-matching techniques may be divided into two techniques: intensity-based matching techniques [Abbo95, Fusi97, Kana94, Rema94, Lott94, Mena97] and feature-based matching techniques [Bhatt97, Mank97, Reid93, Fair95, Xu87, Lee93, Wang95].

2.6.2 Intensity-based approaches

In intensity-based techniques, matching is generally based on maximizing a similarity measure such as sum of squared difference (SSD) between the corresponding areas in left and right images. Intensity-based matching can be done by at least three methods: optical flow, Fourier transformation, and correlation. The use of correlation as a similarity measure between two signals is a well-known technique. It is commonly used in stereo vision for visual correspondence problems [Rema94, Lott94, and Mena97].

The main problem with intensity-based matching [Kana94, Lott94, Fusi97] is that the window size must be large enough to include enough intensity variation for matching but small enough to avoid the effects of projective distortion. If the window size is too small and does not cover enough intensity variation, it gives a poor disparity estimate because the signal to noise ratio is low. If, on the other hand, the window is too large and

covers a region in which the depth of scene points varies, then the position of maximum correlation or minimum SSD may not represent correct matching. This is due to different projective distortions in the left and right images. For this reason, a window size must be selectively adapted depending on local variations of intensity and disparity.

Various methods have been used to overcome some of the above problems. For example, Kanade and Okutomi [Kana94] presented an iterative stereo matching algorithm using an adaptive window. This algorithm selects the size and shape of the matching window adaptively for each pixel on the basis of a local evaluation of the variation in both the intensity and disparity. In this algorithm, Kanade and Okutomi proposed starting out with an initial disparity estimate that was computed over the image using a window with an invariant shape and size. They then refined this estimate iteratively using an adaptive window. As observed by Kanade and Okutomi [Kana94], when the correlation window covered a region with non-constant disparity, area-based matching was likely to fail, and the error in the depth grew with the window size. To overcome such difficulties, Kanade and Okutomi proposed a statistically sound, adaptive technique which selected for each pixel the window size that minimized the uncertainty in the disparity estimates.

In another adaptive approach, Lotti and Giraudon [Lott93] presented a correlation-based algorithm with an adaptive window-size constrained by an edge map extracted from stereo aerial images. In this algorithm, the authors showed the effect of contour-constrained windows on correlation computation. They also presented the use of disparity limits as an initial map of Kanade correction for sub-pixel precision. Fusiello et al. [Fusi97] presented a new robust area-based algorithm, addressing all problems (i)-(iii) previously listed by exploiting symmetry in matching and multiple windows. For this reason it was called the Symmetric Multi-Window (SMW) algorithm. The SMW algorithm assumed that conjugate pairs are along raster lines and the image intensity of a 3D point is the same on the two images. In the SMW algorithm, occlusions are detected by checking the left-right consistency, and suppressing unfeasible matches accordingly. Sung et al. [Sung97] proposed an adaptive window algorithm with four direction sizing factors (AWA-FSF). This algorithm defined eight districts: four side districts and four corner districts, and determined four window sizing directions and four sizing factors

using the extracted mutual information obtained from the relation of a corner district and a side district.

In another adaptive approach, Menard and Kropatsch [Mena97] proposed a new method to detect the optimal scale to determine the corresponding region for each location in a given stereo pair. For each region in the stereo pair, the scale parameter of the correlation scale-space could change continuously and adaptively depending on the gray level, disparity information, and size of the search window. Furthermore the shape of the search window was changed from rectangular to circular.

Most classical correlation methods fail near the disparity discontinuity, which occurs at the boundaries of objects. Lan and Mohr [Lan95, Lan97] proposed a partial correlation technique for solving this problem. They used a robust statistical tool to find the best part to be considered for applying correlation. Schmid and Mohr [Schm95] presented a matching method, which was based on invariant of the luminance function.

2.6.3 Feature-based approaches

There are several approaches for solving the feature correspondence problem. For example, Pilu et al. [Pilu97] proposed a new algorithm for performing feature-based stereo correspondence detection based on singular value decomposition. Wang and Ohnishi [Wang95] proposed a stereo matching algorithm based on a grouping of intensity segments. The intensity segments on a scan line were defined as intensity functions on the corresponding intervals. A polynomial fitting technique was used to approximate the intensity segments.

Bhattacharya and Sinha [Bhat97] described a method for addressing the correspondence problem by modeling the approximate 2D affine transformation between the stereo images with the help of complex moments. In this method, corners were chosen as features and around them an intensity kernel was defined and complex moments were calculated. Waxman and Duncan [Waxm86] used the correlation between binocular difference flow and disparity to drive the correspondence between left and right images.

In general, intensity-based techniques have the disadvantage of being sensitive to photometry variations during image acquisition and to distortions as a result of a changed viewing position. Meanwhile, the feature-based techniques have the advantage of being

less sensitive to photometry variations and being faster than area-based techniques. The main difference between intensity-based and feature-based techniques is the direct use of intensity values instead of features in the matching process.

Recently, some algorithms combining the intensity-based and feature-based techniques have taken advantages of the reliable primitives of each technique. Weng et al. [Weng92] used some primitives simultaneously (i.e. intensity value, edgeness, and cornerness) to determine the correct disparity taking into account possible structural discontinuities and occlusions. Cochron and Medioni [Coch92] first performed intensity based techniques which use the local variance of intensities pattern, then obtained accurate disparities using edge information as a feature-based primitive from the blurred disparity map. This method applies a set of constraints to identify and remove low-confidence matches, then performs surface interpolation to obtain a full resolution disparity map. The performance of a stereo vision system based on the above methods depends mainly on the extraction of the optimal features, high-level or low-level primitives (, which is insensitive to image translation and noise,) and optimal fusion of these features.

2.7 Active Vision Paradigm for Early Vision Problems

2.7.1 Overview

Traditional computer vision methodology regarded the visual system as a passive observer whose goal was the recovery of a complete description of the world. This approach led to systems that were unable to interact in a fast and stable way with a dynamically changing environment. To overcome the efficiency and stability requirements of traditional computer vision systems, several variations of a new paradigm appearing under the names active, attentive, purposive, behavior-based, animate, and qualitative vision were introduced in the last decade. A common principle of the new theories is the behavior-dependent selectivity in the way that visual data are acquired and processed.

An image of a scene at particular instant time is called a frame. A sequence of image frames taken from a changing world, is the input to a dynamic scene analysis system. The variations in a scene may be due to the motion of the camera, the motion of

the objects, variation in illumination, occlusion, or object deformation. There are four possibilities for the dynamic nature of the camera and world setup [Jain95]:

1. Stationary camera, stationary objects (SCSO)
2. Stationary camera, moving objects (SCMO)
3. Moving camera, stationary objects (MCSO)
4. Moving camera, moving objects (MCMO)

SCMO scenes have received the most attention in dynamic-scene analysis. MCSO and MCMO are very important in navigation applications. MCMO is the most general and possibly the most difficult situation in dynamic scene analysis, but it is also the least developed area of computer vision.

Detection of moving objects by a stationary camera is easy [Jang97, Cai95], as the difference between two frames in an image sequence will show the direction and magnitude of the object's motion. A more interesting and difficult problem is the detection of moving objects by a moving camera [Papa93], in most cases an electronically controlled camera. Being able to determine how much an object has moved with respect to the stationary environment, rather than with respect to the observer, is the key to determining whether the object is in motion while the observer is moving. A main interest in recent research is to identify which portions of the motion field correspond to moving objects, and which portions are caused by the moving observer, with or without knowing the ego-motion parameters [Kam93].

2.7.2 Advantages of the Active Vision Approach

Active vision systems in a two-camera system have mechanisms that can actively control camera parameters such as position, orientation, focus, zoom, aperture and vergence in response to the requirements of the task. These systems may also have anthropomorphic features such as spatially variant sensors. Furthermore, active vision approaches encompasses attention, selectively sensing in space, resolution and time, whether it is achieved by modifying the physical camera parameters or the way in which the data is processed after leaving the camera [Swai91].

The computations of early vision could be dramatically simplified by the active vision systems through enabling areas of interest to be examined at the desired resolution,

simplifying segmentation of an object of interest from its background, examining hidden areas of the scene, and simplifying the transition from image to world coordinates.

2.7.3 Applications of Active Vision

In computer vision, active vision approach comes as one of the important area of research. This approach has already proved its usefulness in solving difficult problems in computer vision and demonstrated its power in complex real-time applications. As it is still undergoing full development, more results are expected. Active vision research is likely to enable new applications that are presently neither cost-effective nor technologically feasible without intelligent control of the data acquisition process.

Some applications of active vision [Ibra93] include unmanned ground, air, and underwater vehicles; aids for the handicapped; tracking applications; and household service robots. Other applications that will benefit from active vision approaches include terminal homing to target smart weapons, detecting and tracking intruders, determining a computer user's intentions by tracking their eyes and hands, rummaging through garbage to find items to recycle, defensive driving monitors for automobiles, fruit and vegetable harvesting, assessment of toxic waste sites, and airport runway surveillance. In most of the applications considered, computer vision has already proved useful.

Chapter 3

Overview of a Visual Tracking System

3.1 Introduction

This chapter describes the hardware configuration of an active tracking system. The relationship between camera frame positions and pixel locations at different pan-tilt orientations is investigated. This chapter also describes how a camera captures an image using Datacube (MaxVideo 200) and stores it in the memory of a host computer. Furthermore, this chapter presents the binocular camera model of the tracking system, introduces the pan-tilt unit, and describes some of its characteristics and functions. Finally, this chapter discusses how to control the pan-tilt unit (PTU) from the host computer through the serial port.

3.2 Hardware Configuration

Figure 3.1 shows the block diagram of the active tracking system. The system consists of four components: two gray scale video cameras, two robotic pan-tilt units, a frame grabber (Datacube MaxVideo 200), and a Sun SPARCstation 20. When the system starts, the frame grabber receives an image from the camera(s) and imports it into the SPARCstation 20 which in turn sends the necessary command values to the pan/tilt unit(s) to move the target into the center of the image. This cycle repeats over time.

3.3 Pan Tilt Unit

This system makes use of a computer operated pan-tilt unit (PTU), model PTU-46-17.5 (see Figure 3.2), acquired from Directed Perception, Inc. The pan-tilt-tracking mount provides fast, low-cost, and repeatable movement of the camera attached to it. The PTU controller is the key to communications with the host computer, a Sun Sparc station 20 machine, via RS-232 serial interface. The PTU controller is responsible for driving the PTU, as shown in Figure 3.3.

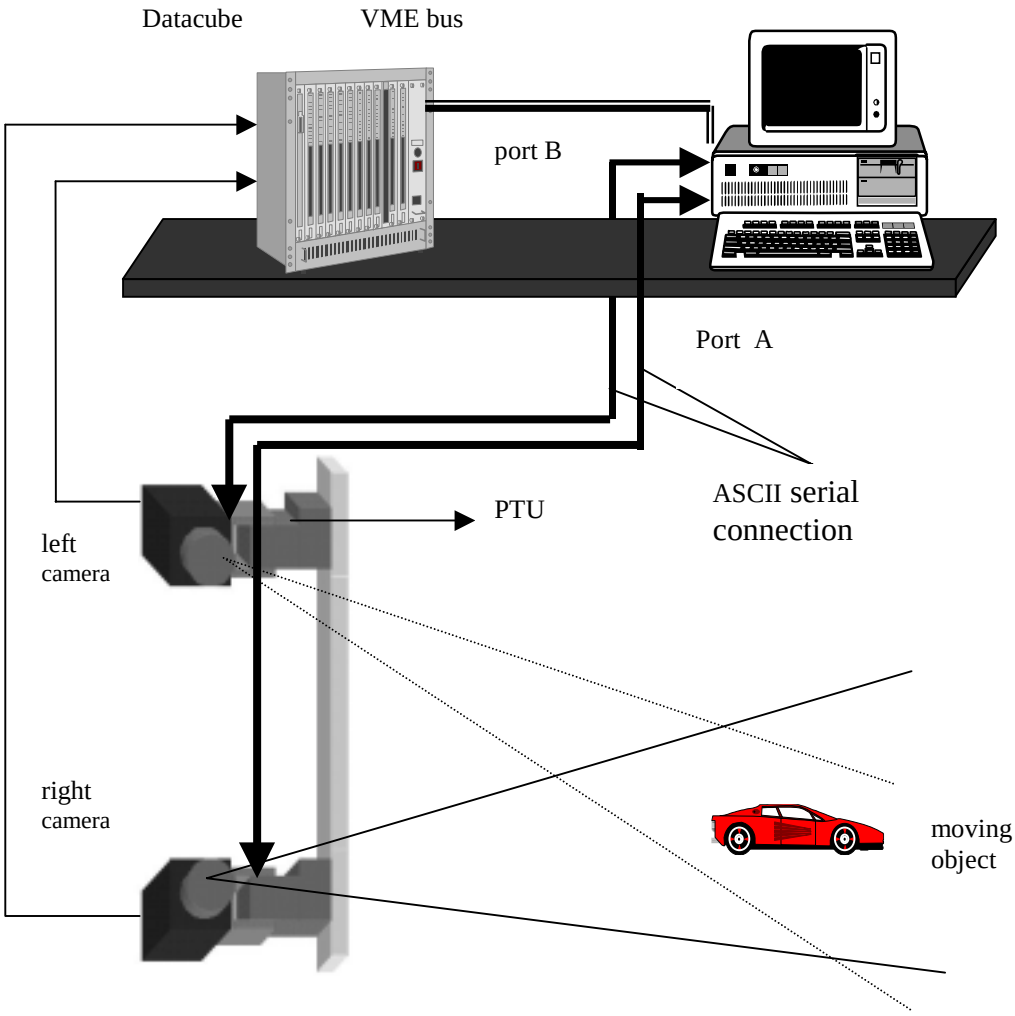


Figure 3.1 An Active Tracking System Configuration

Figure 3.2 Pan-tilt Unit, model PTU-46-17.5 [Perc99]

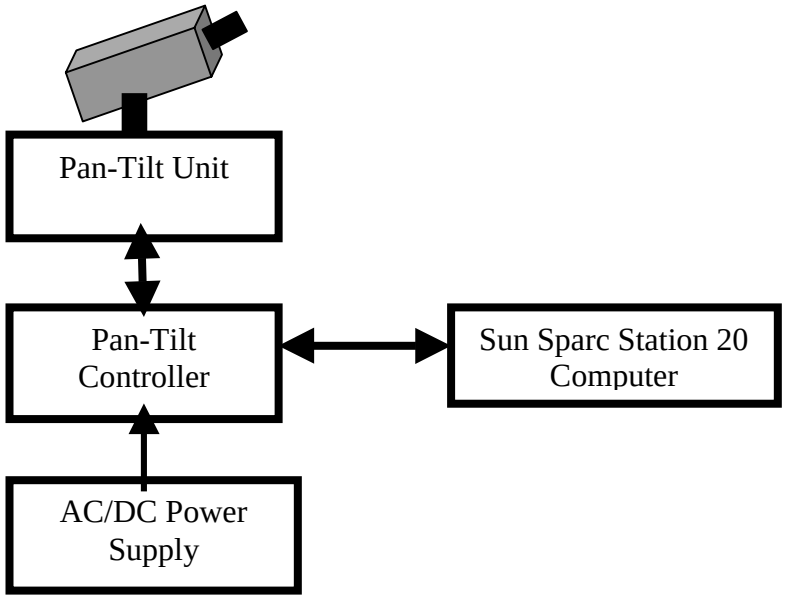


Figure 3.3 System Architecture of the Pan-Tilt Unit

The manual [Perc95] offers a brief overview of the basic features and performance characteristics of the computer controlled PTU. The PTU communicates with the host computer via the serial interface to which it is connected with use of either the interactive ASCII or the encoded binary command sets [Perc95]. When the PTU is

turned on, it goes through a reset cycle and self-calibrates its pan and tilt orientation before coming to rest at the axes home (0,0) position. It also restores its motion settings to the default values.

3.4 Serial Communications

In UNIX, all resources can be accessed as files. Each serial port on a UNIX system has one or more *device files* (files in the */dev* directory) associated with it. For the SPARCstation 20, up two devices can be attached to one 25-pin serial port. The serial port receptacle is labeled A/B.

Communication needs to be established between the host computer and the PTU device via the RS-232 interface before any pan-tilt unit commands can be executed. A communications resource is a physical or logical device providing a single bi-directional, asynchronous data stream. An RS-232 terminal or host computer connects to the female DB-9 connector on the Pan-Tilt Unit Controller (PTU-C). The host terminal or computer should be set to 9600 baud, 1 start bit, 8 data bits, 1 stop bit, and no parity. The PTU controller uses XON/XOFF handshaking to provide for transparent flow control. Using this protocol, the PTU controller sends an XOFF when it can no longer accept characters and an XON when it is again ready to receive characters.

3.5 Computation of Motion Parameters

This section describes the binocular stereo model that used in this research with the computation of the motion parameters of the desired target and the reorientation of the pan-tilt unit (PTU) along the trajectory of the object. The PTU/camera system needs to be calibrated before the PTU's coordinates can be updated. Calibration is done to determine how the motion of the desired target relates to the motion of the PTU in order to keep the desired target near the center of the image frame.

3.5.1 Camera Model

Figure 3.4 illustrates the binocular stereo model that used in this research. In Figure 3.4, p_l and p_r correspond to points of P in the left image and right image respectively. The difference between the two points p_l and p_r is called disparity. In three-dimensional scenes, the plane determined by the point P and the baseline is called the

epipolar plane. The intersection of an epipolar plane with an image plane is called an epipolar line.

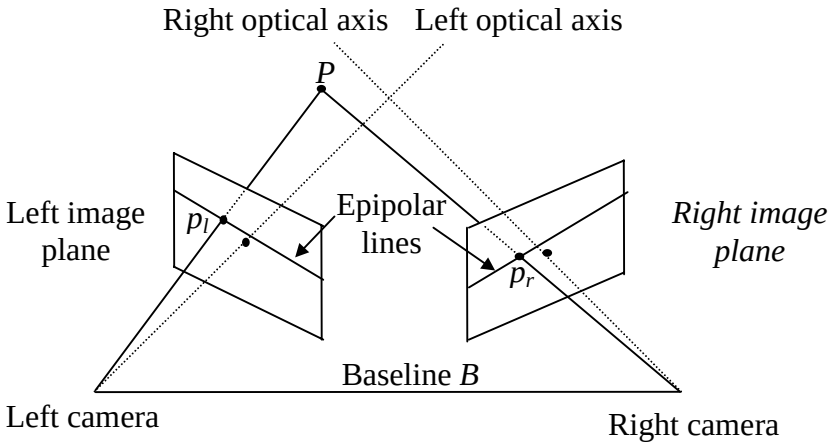


Figure 3.4. The binocular stereo camera model. The points p_l and p_r are corresponding points in left and right image respectively [after Lin96]

3.5.2 Camera Calibration

The purpose of camera calibration is to obtain an estimate of the parameters that determine the transformation of a moving point in the world to pan/tilt steps. This is needed in order to be able to transform between world coordinates and pan/tilt steps to derive the pan-tilt unit to keep the desired object near the center of the image frame.

There is no single method that can be regarded as the ideal camera calibration technique. Various methods have been used for camera calibration [Tsai89, Faug93 and Soat94, Gj98, and R99]. For example, Wagner R. et. al [R99] presented a new method to estimate the ego-motion of a camera from sets of point correspondences taken from a monocular image sequence.

This research concentrates only on a simple calibration of the camera to pan-tilt unit of the visual tracker similar to the one developed by [Mohi98]. It is assumed that the camera model is a perfect perspective projection from the world onto the image plane; therefore, systematic effects, such as lens distortion, and random effects, such as sensor noise, are ignored. Now, we present a complete description of the calibration procedure

Step1: Rotate the camera about the pan and tilt axes between two points in space so that they both lie in the camera's field of view,

Step 2: Capture an image and note the pixel location of each point in the image,

Step 3: Rotate the camera so that the optical axis of the camera lies exactly on one of the points,

Step 4: Record pan and tilt readings,

Step 5: Repeat step 1 to step 4 for the second point,

The ratio of the difference in pan and tilt readings to the distance in pixel between the two points in the image plane gives the λ_x and λ_y calibration factors for the camera in pan and tilt, respectively.

3.5.3 Motion Parameters

In the gaze control module, the control system uses the visual measurements of the target position(s) in the image that results from the matching process to drive the pan/tilt unit(s) to keep the target near the center of an image frame. The pan-tilt unit needs target motion information to adjust its pan and tilt steps. This information is obtained by model matching with the help of Kalman filter. The pan-tilt unit is controlled from the host computer via a serial port. Chapter 5 describes how to utilize state estimate of a Kalman filter in controlling pan-tilt unit to move the cameras.

3.6 Image Acquisition

Image acquisition is performed using a Pulnix TM-7EX gray scale camera (see Figure 3.5) with 12.5 mm lens [Puln99]. Captured images are stored in the video RAM of the frame grabber (MaxVideo 200) in raw format with 256 gray levels and 512×484 pixels (512 pixels image width and 484 image height). Each pixel represents a gray level in the range 0 to 255, requiring one byte per pixel. The camera is mounted on the pan-tilt unit's tool platform, which can change FOV (field of view) of the camera. The origin of the image coordinate system is located at the center row and column. This camera fits easily, both physically and functionally, into all types of machine vision, automated inspection, and related applications. Other uses include remotely piloted vehicles, miniature inspection devices, surveillance, microscopes and medical equipment.

Figure 3.5 Pulnix TM-7EX gray scale camera [Puln99]

3.7 Datacube Image Processing Devices

The Datacube image-processing device [cube99] is a family of pipeline image processing boards sharing common characteristics such as modularity and cascadability. Each device is said to be modular since it is a specialist at performing specific image processing tasks. For instance, some devices are used for image storage; others are used for image acquisition, and others are used to perform image convolutions. Many devices perform more than one type of operation, and certain operations can be carried out by a number of devices. This modularity allows the arrangement of boards to suit the application. In this system (see Figure 3.1), the MaxVideo 200 is attached to a VME backplane, which is in turn connected to the S-BUS of a Sun Sparc station 20 through a performance technologies PT-SBS915 adapter. ImageFlow is a library of functions that allows the Sun Sparc station 20 to interact with the MaxVideo 200.

The Datacube image processing devices share many of the same features, including a VMEbus host environment and MAXbus for sharing data between the devices. The Datacube MaxVideo 200 Image Processor (see Figure 3.6) is a VMEbus-based high-performance video image processing system. The MaxVideo 200 contains a pipeline video processor and can be connected to other VME-based video processor

boards via either or both of the VME P2 connectors and the MaxBus. The MaxVideo 200 board in this system is supplied with the following modules:

- AB (Architecture Adaptor, version B) device,
- AM (Advanced Memory) device,
- AG (Analog Generator) device,
- AP (Advanced Pipeline Processor) device,
- AS (Analog Scanner) device, and
- AU (Arithmetic Unit) device

The MaxVideo 200 board is also accompanied by Datacube ImageFlow 2.6-support software under Solaries 2.5.1. ImageFlow is software for the control of pipeline image processors built from Datacube image processing devices. It consists of a number of “C” callable function libraries. ImageFlow functions work across all Datacube image-processing devices. The same function call is used to control all of the same types of elements on all the different devices. The application code is created entirely from calls to the ImageFlow library and many programming complexities are taken care of by ImageFlow and are hidden from the programmer. All hardware devices and ImageFlow components are objects. The programmer uses ImageFlow to create, manipulate, and dispose of these objects. Examples of these objects include data pipe objects, data surface objects, and image processing device objects.

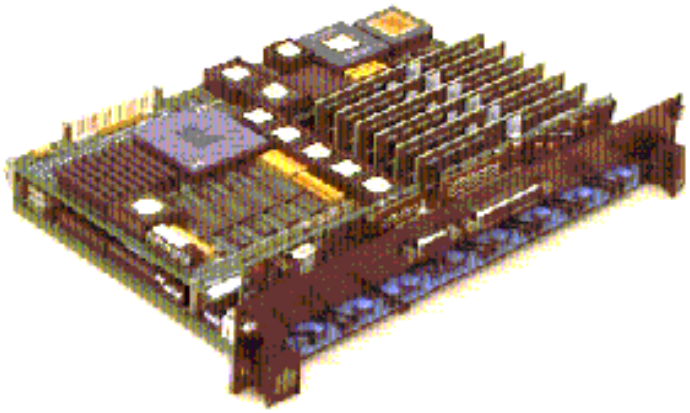


Figure 3.6 Datacube MaxVideo 200 [Cube99]

Chapter 4

Area-based Monocular Visual Tracking

4.1 Introduction

This chapter describes a novel monocular tracking system that combines Kalman-type prediction with steepest-descent search for correspondences, using 2-dimensional affine mappings between images for passive and active camera. This system has been developed to track a moving object for both passive and active cameras. It utilizes a Kalman filtering so that a tracking history can be used to predict a search area for matching and to control the pan-tilt unit. This chapter also presents experimental results for a monocular image sequence to show the performance of the system in different situations.

4.2 Difficulties of the System

This work aim at developing a reliable monocular tracking system to keep a moving object centered in the image plane. The approach has been tested through the use of a region-based, steepest-descent matching technique that has been used successfully for monocular tracking [Shi94]. Early experiments with this technique showed the need for a good initial estimate (starting point) in order to locate the correct correspondence. Experiments also showed that the window size has significant effects on the performance of the matching.

The matching approach can correctly detect the corresponding point only when the starting point selected is close to the corresponding point. To illustrate this, Figure 4.1 shows two successive images with a target point given in the first image. Several starting points were selected manually, each at different distances away from the corresponding point in the second image. For this target point, the matching method worked correctly when the starting point was up to 5 pixels from the corresponding point. The matching method did not work for the other cases.

To provide this estimate, a system that combines Kalman-type prediction with this matching approach has been developed. Kalman-based predictions are integrated with fast area-based search for correspondences to enhance the tracking robustness.

The search technique used here is very sensitive to the choice of window size, and a new approach based on the moment invariants (see Chapter 6) was developed to select that size. Intuitively, a window that is too small will tend be distracted by texture primitives, or (at the other extreme) it may enclose a featureless region that is unsuitable for area-based comparisons. For a window that is too large, the similarity measurements may give incorrect matches due to repeating patterns, occlusion, and perspective differences.

Figure 4.2 shows two successive images with a target point given in the first image. The matching method was tested at the target with different window sizes. The results indicate that the matching approach works for some window sizes only, not all of them.

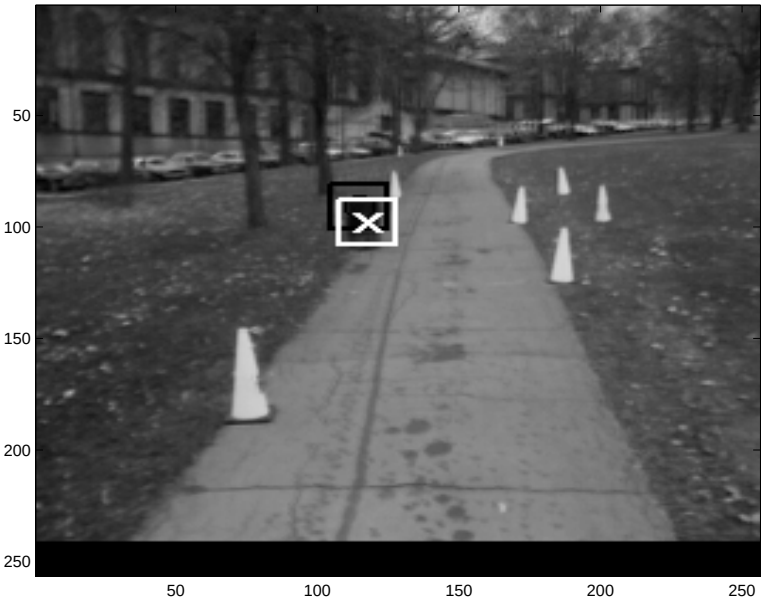
Another difficulties in making this system work well are compensation for the visual processing delay in the vision system and measurement noise. An additional difficulty results from noise in the vision system and lighting effects.

4.3 A Novel Monocular Tracking System

The typical tracking paradigm comprises a repeating cycle of measurement and prediction followed by gaze control (see Figure 4.3). The measurement process, in turn, involves both feature detection and matching [Sala98c].

In the matching phase, the matching is performed between two successive frames using the steepest-descent search technique to compute the motion parameters. The question then is how adjacent two successive images must be. If a time interval between two successive images is too short compared to the movement of an object, these two images may not show clear differences and the detection process may fail to identify movement itself. On the other hand, if the time interval is too long, the detection process may fail to detect the desired target.

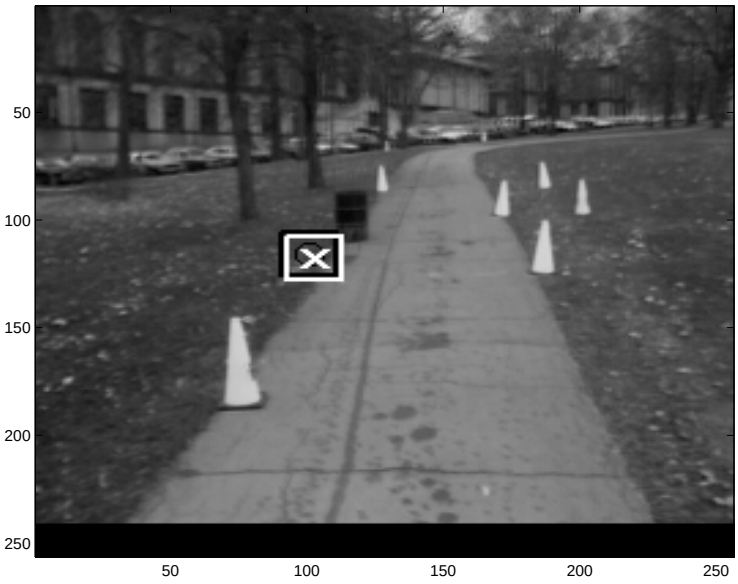
(a)



(b)

Figure 4.1. The effect of starting points on the matching method. The first frame of an image sequence is shown in (a), and the second frame appears in (b)-(d). The center of each black rectangle denotes a starting point selected manually. Each white "×" indicates the target detected by the correspondence search method. (a) The first frame with a given target and a window (25×25) centered at the target. (b) Second frame with starting point 5 pixels away from the corresponding point in the second frame. The correct match is found.

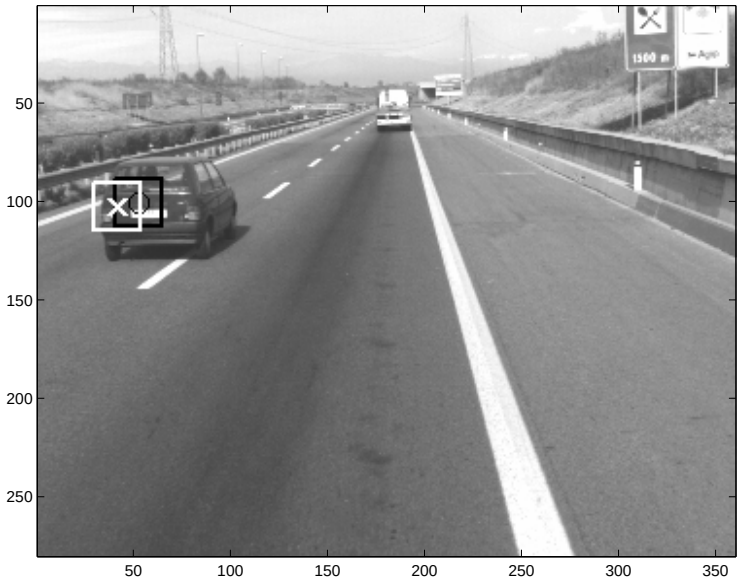
(c)



(d)

Figure 4.1, continued. (c) Second frame with starting point 15 pixels away from the corresponding point in the second frame. An incorrect match is found. (d) Second frame with starting point 20 pixels away from the corresponding point in the second frame. An incorrect match is found.

(a)



(b)

Figure 4.2. The effect of manually selected window sizes on the matching method. The center of each black rectangle denotes a starting point selected manually. Each white "×" indicates the target detected by the correspondence search method. (a) The first frame with a given target and three windows centered at the target. The window sizes are 25×25, 55×55, and 85×85. (b) Second frame using window size 25×25. An incorrect match is found.

(c)

(d)

Figure 4.2, continued. (c) Second frame using window size 55×55. An incorrect match is found. (d) Second frame using window size 85×85. A correct match is found.

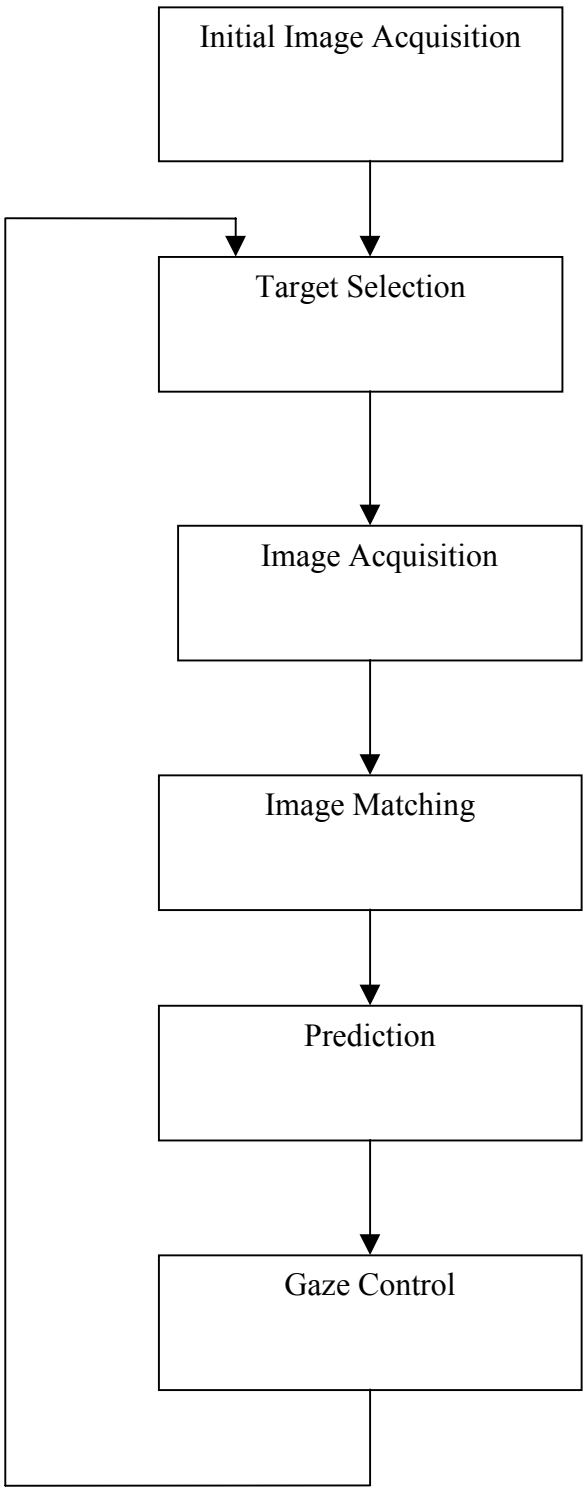


Figure 4.3 Function block diagram of the monocular active tracking system

In the prediction phase, a feedback loop is established by predicting new locations with Kalman filters and using them to guide the extraction of the corresponding location in the next frame. Much tracking work [Ayac89] has focused on the problem of prediction, often with Kalman filtering [Jang97, Mank97, Reid93, Cox96, Fair95], and has assumed that feature detection and matching are relatively straightforward. On the other hand, such studies as [Shi94, Toma92] have assumed the opposite – that prediction is relatively simple, often due to a constraint of small frame-to-frame motion – and have concentrated on the problems of feature detection and matching.

In the gaze control module, the control system will use the visual measurements of the target position in the image to drive the pan-tilt unit to keep the target near the center of an image frame. The pan-tilt unit needs target motion information to adjust its pan and tilt angles. This information is obtained by model matching with the help of Kalman filter. A model of the target is automatically updated during the matching process. The pan-tilt unit is controlled from the hosts via a serial port.

4.4 The Search for Correspondences

Region-based matching typically involves the comparison of images using similarity measures that derive from cross-correlation or sum of squared differences (SSD). In spite of several well-known limitations, including sensitivity to photometry variations and partial occlusion, region-based computations are straightforward and have been used successfully in many situations. Another common disadvantage, sensitivity to changes in viewpoint, is largely overcome in the system described here through 2D affine deformation.

Another approach is that of cross correlation [Mont94]. In order to compute the cross correlation function of two windows, a template window is shifted pixel by pixel across a larger search window. At each position the cross correlation coefficient ρ is computed using

$$\rho = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y)-\mu_1)(J(x,y)-\mu_2)}{\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y)-\mu_1)^2\sum_{x=1}^{M}\sum_{y=1}^{N}(J(x,y)-\mu_2)^2}} \qquad -1 \le \rho \le 1 \tag{4.1}$$

where

$I(x,y)$	Gray values of template window
$J(x,y)$	Gray values of search window
μ_1	Mean of Gray values of template window
μ_2	Mean of Gray values of search window
M,N	Number of rows and columns of template window

The best match between the template and the search window maximum of the resulting cross correlation function defines the position of the best match between the template and the search window.

The method introduced by Shi and Tomasi [Shi94] has been used for monocular tracking, and it uses a search for 2D affine transformation parameters (see Appendix A) that minimizes the sum of squared difference between two images. This section summarizes the matching approach.

The goal of successful target matching using 2D affine mapping is to find values for *A* and *d,* as described below that optimize a similarity criterion for two images. Given a window in an image, the displacement of a pixel *X* within the window can be expressed as:

$$U = DX + d \tag{4.2}$$

where $U = \begin{bmatrix} u \\ v \end{bmatrix}$ is called the displacement vector of *P*, $D = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$ is called the deformation matrix of the window, and *d* is the translation amount. It is not difficult to show

$$\begin{aligned} X + U &= X + DX + d \\ &= AX + d \end{aligned} \tag{4.3}$$

where

$A = I_2 + D,$

and I_2 is the 2×2 identity matrix. From equation (4.2), given a target point in an image *I,* and given a second image *J,* an ideal match between the two images occurs when

$$I(X) = J(AX + d). \tag{4.4}$$

The method to determine the deformation matrix and translation vector is based on a minimization of the sum-of-squared-differences criterion as follows,

$$e = \iint_w [J(AX + d) - I(X)]^2 dX \tag{4.5}$$

where *w* is a given window in image *I.* To minimize *e*, this equation is differentiated with respect to the six coefficients of *D* and *d* respectively, and the results are set to zero. This leads to six equations that, after image *J*(*AX*+*d*) is approximated by its Taylor series expansion to the first order, can be expressed in the following matrix form,

$$Tz = a, \tag{4.6}$$

where

$$T = \iint_w \begin{bmatrix} x^2 g_x^2 & x^2 g_x g_y & xyg_x^2 & xyg_x g_y & xg_x^2 & xg_x g_y \\ x^2 g_x g_y & x^2 g_y^2 & xyg_x g_y & xyg_y^2 & xg_x g_y & xg_y^2 \\ xyg_x^2 & xyg_x g_y & y^2 g_x^2 & y^2 g_x g_y & yg_x^2 & yg_x g_y \\ xyg_x g_y & xyg_y^2 & y^2 g_x g_y & y^2 g_y^2 & yg_x g_y & yg_y^2 \\ xg_x^2 & xg_x g_y & yg_x^2 & yg_x g_y & g_x^2 & g_x g_y \\ xg_x g_y & xg_y^2 & yg_x g_y & yg_y^2 & g_x g_y & g_y^2 \end{bmatrix} dX \tag{4.7}$$

$$z = \begin{bmatrix} d_{xx} & d_{yx} & d_{xy} & d_{yy} & d_x & d_y \end{bmatrix}^T \tag{4.8}$$

$$a = \iint_w [I(X) - J(X)] \begin{bmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{bmatrix} dX \tag{4.9}$$

To solve equation (4.5), the T matrix is computed directly from image J, and vector a (which involves image differences) is calculated from both J and I. Then the unknown vector z (and equivalently the affine deformation matrix A and the translation vector d) can be obtained. Because an initial solution of (4.5) is only a rough approximation, z can be refined though Newton-Raphson styles iteration. When e becomes small, the iterations stop and the final value for d represents the detected match location.

Given two successive images I and J and a window in image I, *monocular tracking* in this dissertation means repeatedly determining the six scalar values that appear in the deformation matrix D and displacement vector d. The quality of this estimate depends on the size of the feature window and the textures of the image within it. When the window is small, the matrix D is harder to estimate, because the variations of motion within it are smaller. However, smaller windows are in general preferable for tracking because they are less likely to straddle a depth discontinuity. In some cases, a pure translation model is preferable during tracking, where the deformation matrix D is assumed to be zero. This is represented as

$$U = d \tag{4.10}$$

4.5 Estimation of Motion Parameters in Monocular Image Sequence

The matching method discussed in the previous section has been shown to work well when the disparity between any two consecutive images in a motion sequence is small. When frame-to-frame disparities are large, a good initial estimate is needed. This section describes how to use a constant-image-velocity Kalman filter [Jang97, Cox96] to perform this prediction. The estimated parameters are then used to reduce the searching scope for model matching and to control the PTU. A state model that is linear and is defined by the following equations is assumed:

$$X_k = \varphi_k X_k + W_k \tag{4.11}$$

At time k, X_k is the system state vector, φ_k is the state transition matrix, W_k is a Gaussian, zero-mean, temporally uncorrelated system noise vector with covariance Q_k.

In the monocular tracking system, the five Kalman filter equations (see Appendix B) are implemented in a straightforward manner with X_k and φ_k

$$X_k = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, \qquad \varphi_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where (x, y) represent the target position and $(\dot{x}, \dot{y})$ represent the target velocity.

The Kalman filter algorithm tries to estimate system states based on a set of measurements. A linear relationship is assumed between system states and a set of measurements as

$$Z_k = H_k X_k + V_k \tag{4.12}$$

where Z_k is the measurement vector (obtained by the matching process between two successive frames, as described in the previous section), H_k is the matrix relating the state vector to the measurement vector, and V_k is a Gaussian, zero-mean, temporally uncorrelated measurement noise vector with covariance R_k. For completeness, Appendix B offers an example and a picture of the operation of the Kalman filter. Z_k and H_k are then formed as

$$Z_k = [x, y, \dot{x}, \dot{y}]^T,$$

$$H_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Once a system model and a measurement model are defined, a recursive Kalman filter algorithm can be applied to predict a search area for matching and to control the pan-tilt unit. The recursive Kalman filter algorithm consists of three phases of operations (see Appendix B): initialization, state estimation, and measurement update. The initialization phase determines initial state estimate $\hat{X}_0^-$, initial error covariance matrix $\hat{P}_0^-$ that represents deviation of $\hat{X}_0^-$ from actual initial state X_0, system error covariance matrix $Q_k = E(W_k W_k^T)$, and measurement error covariance matrix $R_k = E(V_k V_k^T)$.

The values of $\hat{X}_0^-$, is assumed to be the target location and its velocity in the subsequent frame. The values of $\hat{P}_0^-$, Q_k, and R_k are each assumed to be the identity matrices.

The phase of state estimation determines a priori estimate and its error covariance matrix for the current state based on the previous state estimate and error covariance (see Appendix B for details).

The first task during the measurement update is to compute the Kalman gain, K_k. The next step is to obtain measurements Z_k by means of the matching processes, and then to generate a posterior state estimate $\hat{X}_k$. The final step is to obtain a posterior error covariance estimate P_k. After each time and measurement update pair, the process is repeated with the previous posterior estimates used to predict the new priori estimates.

The Kalman filter performance can be improved by tuning the filter parameters Q_k and R_k [Mayb79]. The computation of Kalman gain matrix K_k indicates how much weight is given to a new measurement. If Q_k increase, the Kalman gain will generally increase. If K_k is large, the updated system state is dominated by Z_k. Similarly, a large covariance matrix R_k of measurement noise that corresponds to large uncertainty of the measurement causes small weighting K_k. If K_k is small, the measurement Z_k is ignored from the computation of $\hat{X}_k$ and the updated system state is determined primarily by $\hat{X}_k^-$, which is the state vector predicted by the transition matrix φ_k and the previous state vector at time *k*-1.

4.6 Treatment of Occlusion

This section describes how the system deals with occlusion in the monocular image sequence. Object occlusion is defined as any situation where an object is not completely visible from the viewpoint of the camera. There exist two types of occlusion (see Figure 4.4). The first is partial occlusion, where part of the object is occluded with another object or obstacle; the second is total occlusion, where the entire object is not visible to the camera.

When a target is totally occluded, the template will not match any area of the image very well. Thus, the matching residue is expected to be large, which results in noisy information about the location of the target. Using a larger window increases the chance of searching for the desired target in the next frame. This leads to a delay resulting from longer visual processing. When the occlusion is detected, tracking continues with the predicted locations that are computed by Kalman filtering. Although the method used is a constant velocity Kalman filtering, it has been shown to function reasonably well in a non-constant velocity situation.

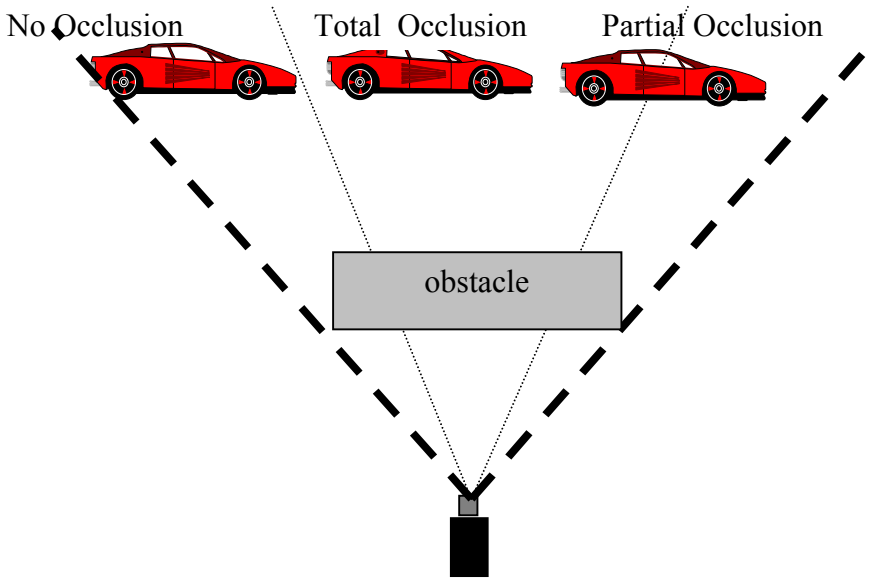


Figure 4.4 Occlusion in monocular tracking system.

In the case of occlusion, the output of the matching process that describes the location of the target is very noisy; therefore, the system should not depend on these measurements but rather continue tracking with the predicted locations using Kalman filtering. This large noise, V_k, would be expressed in a modified covariance, R_k, that would in turn influence the Kalman equations in such a way as to reduce the influence of this measurement. Lynch et al. [Lync89] developed an approach for tracking partially occluded two-dimensional polygonal shapes undergoing unknown two-dimensional translation and rotational motion based on Kalman filtering. The locations of the

extracted corners for the polygon are the tracking features, which produce an estimator that effectively discounts the contribution of the measurements that cause the feature occlusion. This estimator was adapted to apply to tracking a moving object using the area-based matching technique for monocular and binocular image sequences by reducing the influence of the occluded measurements. To develop this approach, a simple diagonal transformation, T_k, was first defined to produce a new noise vector from the original but with some of the components amplified.

$$V_k = T_k V_{k\,(\text{nominal})}$$

where $T_k = \begin{bmatrix} t_{11} & 0 & 0 & .. & .. \\ 0 & t_{22} & 0 & .. & .. \\ 0 & 0 & t_{33} & .. & \\ 0 & 0 & 0 & & \\ .. & .. & .. & .. & .. \end{bmatrix}$, $t_{ij} = 0$ if $i \neq j$, $t_{ij} = 1$ if $i=j$ and no occlusion was detected, and $t_{ij} = \zeta >> 1$ if i=j and occlusion was detected, and where ζ is the residual error of the match. The covariance of V_k (nominal) is related to R_k (nominal) as follows:

$R_{k\,(\text{nominal})} = E[V_{k\,(\text{nominal})}\ V^T_{k\,(\text{nominal})}]$,

then

$$R_k = E[V_k V_k^T] = T_k R_{k\,(\text{nominal})} T_k^T$$

The state estimate and the error covariance extrapolations are unchanged. However, the Kalman gain is modified by

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + T_k R_{k\,(\text{nominal})}\ H_k^T)^{-1} \tag{4.13}$$

To eliminate the effect of the occluded measurement, the limit of the large elements in T_k are set to infinity. It will be easier to use the T_k^{-1} in the following

$$K_k = P_k^- H_k^T \left(T_k^T\right)^{-1} \left(T_k^{-1} H_k P_k^- H_k^T \left(T_k^T\right)^{-1} + R_k\right)^{-1} T_k^{-1} \tag{4.14}$$

From the definition, T_k is a diagonal matrix, hence $T_k = T_k^T$. Define S_k so that $S_k = \lim_{\zeta \to \infty} T_k^{-1}$, where S_k is the measurement subset matrix at step k, $s_{ij} = 0$ if $I \neq j, s_{ij} = 1$ if $i=j$ and no occlusion was detected, $s_{ij} = 0$ if $i=j$ and occlusion was detected. Then

$$K_k = P_k^- H_k^T S_k \left(S_k H_k P_k^- H_k^T S_k + R_k \right)^{-1} S_k \tag{4.15}$$

where the error covariance update is affected by the revised measurement error covariance

$$P_k = \left(I - K_k H_k \right) P_k^- \tag{4.16}$$

The state estimate update is unchanged in form

$$\hat{X}_k = \hat{X}_k^- + K_k \left(Z_k - H_k \hat{X}_k^- \right) \tag{4.17}$$

However, substitution of the Kalman gain expression (4.15) into (4.17) shows that S_k may be grouped with H_k as a factor $H_{s,k} = S_k H_k$

If an element in Z_k represents the occluded measurement, it is converted to zero value by pre-multiplying with S_k

$$\hat{X}_k = \hat{X}_k^- + K_{s,k} \left(S_k Z_k - H_{s,k} \hat{X}_k^- \right) \tag{4.18}$$

where

$$K_{s,k} = P_k^- H_{s,k}^T \left(H_{s,k} P_k^- H_{s,k}^T + R_k \right)^{-1} S_k \tag{4.19}$$

4.7 Experimental Results Using Monocular Image Sequence

This section reports the results obtained by applying the proposed algorithm to real monocular image sequences. The monocular cone, tree, truck, car, and train image sequences were downloaded from the computer vision home page to test the proposed tracking system [Visi99].

In one experiment, as illustrated in Figure 4.5, the system tracked a cone over a monocular sequence of 18 frames. Because the relative motion of the cone is toward the camera, its 2-dimensional image velocity is not constant, and therefore violates a fundamental assumption of the Kalman filter. In spite of this, the tracking system performs well because the matching process is able to compensate.

In the cone sequence, the desired target (a dark barrel) was selected manually in the first two consecutive frames and an initial velocity estimate was also provided. A window size is automatically selected initially that is well suited for the desired target. After that, the Kalman filter provided estimates of new image locations, and each was

used as the starting point for a correspondence search. Figure 4.6 shows the detected trajectory of the dark barrel in the images generated by matching, together with the predicted locations generated by the Kalman filter. In spite of the rapid target movement near the end of the sequence, the predicted location was never more than approximately 8 pixels distant from the actual target location, and this was close enough to initiate the matching process.

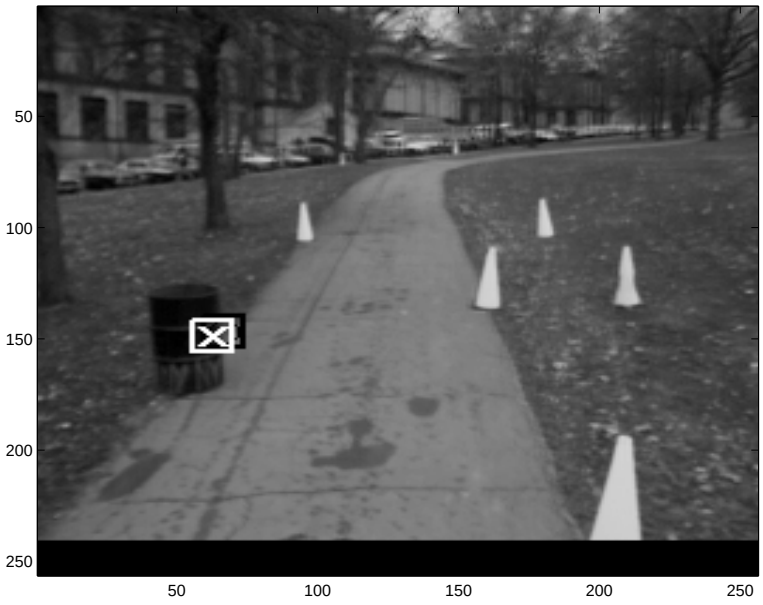
Figure 4.7 shows the matching residue between the image frames. In this figure, it can be seen the matching residues are all quite small, and this is because a good starting point was estimated by Kalman filter, the window size was selected adaptively, and the object was not occluded by another object. Figure 4.8 contains plots of the errors exhibited by the system for this image sequence. The first plot in Figure 4.8 represents the image distance (measured in pixels) from each predicted location to the true target location, and the second plot represents the image distance separating the detected correspondence from the true target location. The worst-case error after matching is approximately 5 pixels, which is acceptable in our application. But perhaps more importantly, these matches often represent a significant correction to the locations predicted by the Kalman filter. Without the predictions provided by the Kalman filter, the steepest search performed very poorly; and without the "measurements" provided by the matching process, the Kalman filter cannot function.

(a)

(b)

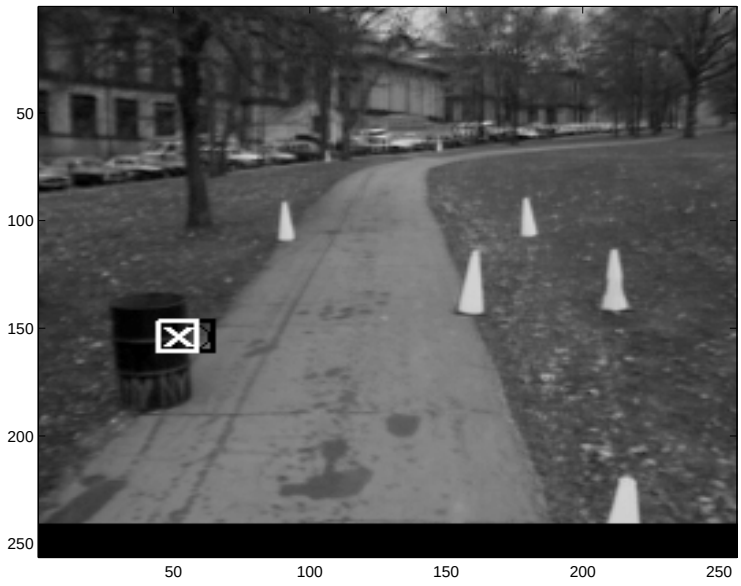
Figure 4.5. Selected images from "cone" image sequence after applying the tracking algorithm. (a) First frame in sequence. (b) Frame 3. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method. A window size is automaticaly selected initially that is well suited for the desired target.

(c)



(d)

Figure 4.5, continued. (c) Frame 6. (d) Frame 14. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



(e)

(f)

Figure 4.5, continued. (e) Frame 15. (f) Frame 18. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.

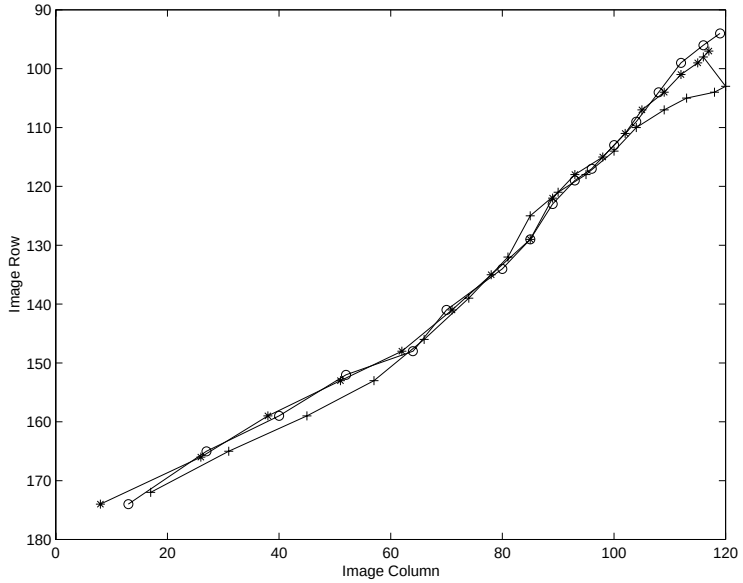


Figure 4.6. Actual trajectory (o), detected trajectory (∗), and points predicted by the Kalman filter (+) for the tracked target Figure 4.5. The target travels toward the lower left of the image.

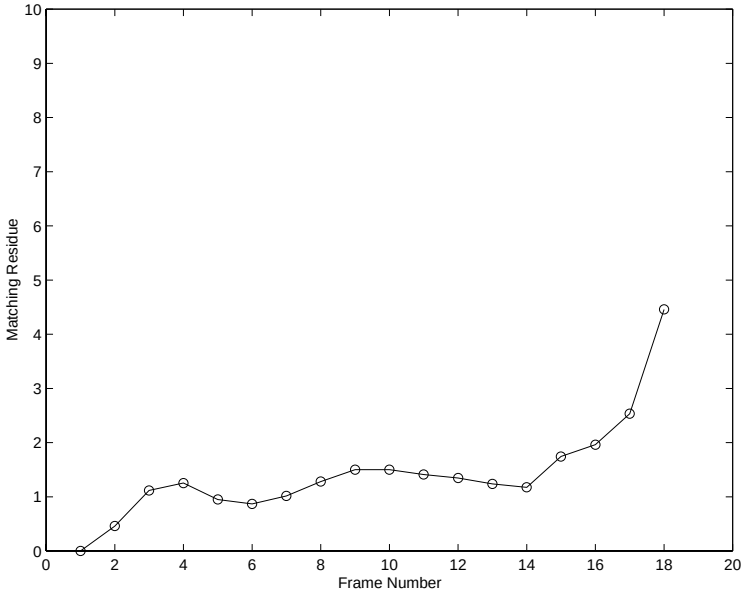


Figure 4.7. Matching residues for the tracked target in Figure 4.5. The matching residues are quite small, and this is because the object is not occluded with other objects.

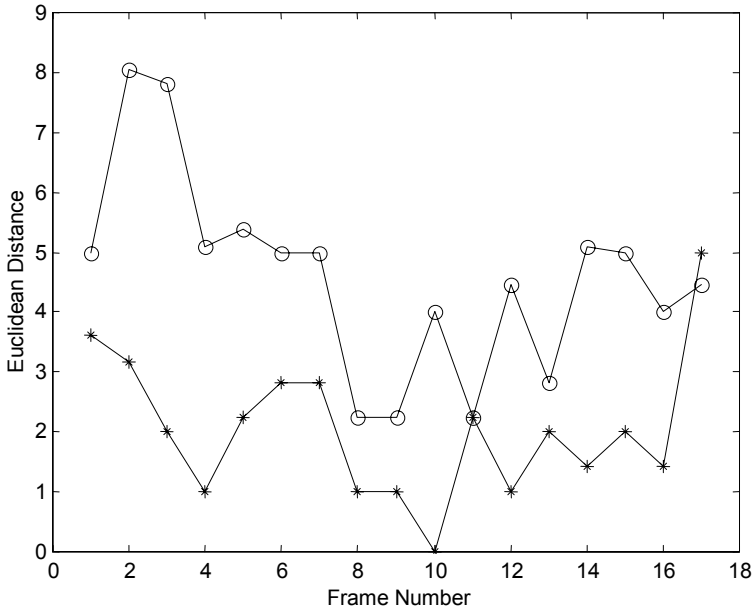


Figure 4.8. Euclidean distance from actual target location to predicted location (o) (in units of pixels), and distance from actual target to detected location (∗) for the tracked target in Figure 4.5.

In another experiment, as illustrated in Figure 4.9, the system tracked a car over a monocular sequence of 46 frames.

In the car sequence, the user has selected the target of interest in the first frame and window size has been selected adaptively in the first frame. After that, the Kalman filter provided estimates of new image locations, and each was used as the starting point for a correspondence search. Figure 4.10 contains plots of the errors exhibited by the system for this image sequence. The first plot in Figure 4.10 represents the image distance (measured in pixels) from each predicted location to the true target location, and the second plot represents the image distance separating the detected correspondence from the true target location. The predicted location was never more than approximately 6 pixels distant from the actual target location, and this was close enough to initiate the matching process. The worst-case error after matching is approximately 3 pixels, which is acceptable in our application. Figure 4.11 shows the matching residue between the image frames.

In another car sequence, as illustrated in Figure 4.12, this time the system tracked a car over a monocular sequence of 61 frames. In the car sequence, the desired target was selected manually in the first two consecutive frames. After that, the Kalman filter provided estimates of new image locations, and each was used as the starting point for a correspondence search. In this example, a window size has selected adaptively in the first frame. During the occlusion event, however, tracking continues with Kalman gains that are computed based on the large assumed covariance for the location of the target. Although the method used a constant velocity model for its development, it has been shown to function reasonably well in a non-constant velocity situation, with the result that tracking is performed successfully throughout the entire image sequence. Figure 4.13 shows the matching residue between the image frames. In this Figure, we can notice that starting from frame 8 the matching residue increased, which means that the white car might have been occluded by the black car. Without the predictions provided by the Kalman filter, the steepest search performed very poorly; and without the "measurements" provided by the matching process, the Kalman filter cannot function.

Figure 4.14 shows another example to track a target over a monocular tree sequence of 46 frames. In this Figure, the user has selected the target of interest in the first frame, and window size has selected adaptively in the first frame. After that, the Kalman filter provided estimates of new image locations, and each was used as the starting point for a correspondence search. It is shown that the tracking system performs well in case of occlusion. Figure 4.15 shows the matching residue between the image frames. In this Figure, we can notice that the matching residue is small despite the occurrence of an occlusion. This is because an object having the same intensity occluded the desired target.

(a)

(b)

Figure 4.9. Selected images from "car" image sequence after applying the tracking algorithm. (a) First frame in sequence. (b) Frame 4. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method. A window size is automaticaly selected initialy that is well suited for the desired target.

(c)

(d)

Figure 4.9, continued. (c) Frame 28. (d) Frame 44.

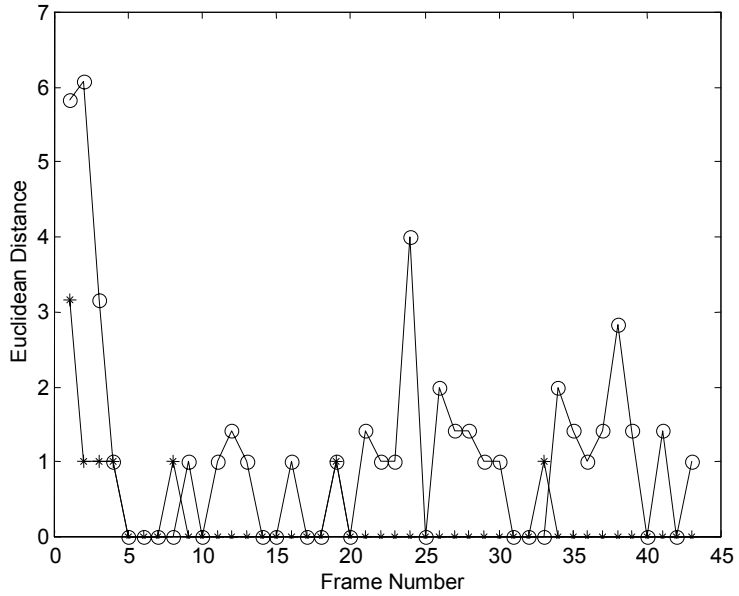


Figure 4.10. Euclidean distance from actual target location to predicted location (o) (in units of pixels), and distance from actual target to detected location (∗) for the tracked target in Figure 4.9.

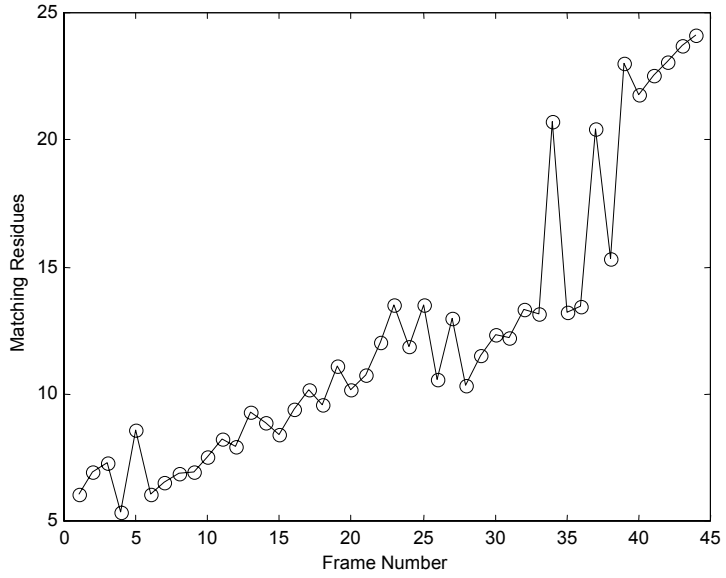


Figure 4.11. Matching residues for the tracked target in Figure 4.9.

(a)

(b)

Figure 4.12. Selected images after applying the tracking algorithm from image sequence taken with the car driving behind another car, which causes partially occlusion in the image sequence. (a) First frame in sequence. (b) Frame 3. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.

(c)

(d)

Figure 4.12, continued. (c) Frame 12. (d) Frame 59. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method. a window size is automatically selected initially that is well suited for the lighter vehicle. During the occlusion event, the tracking continues with Kalman gains that are computed based on the large assumed covariance for the location of the target

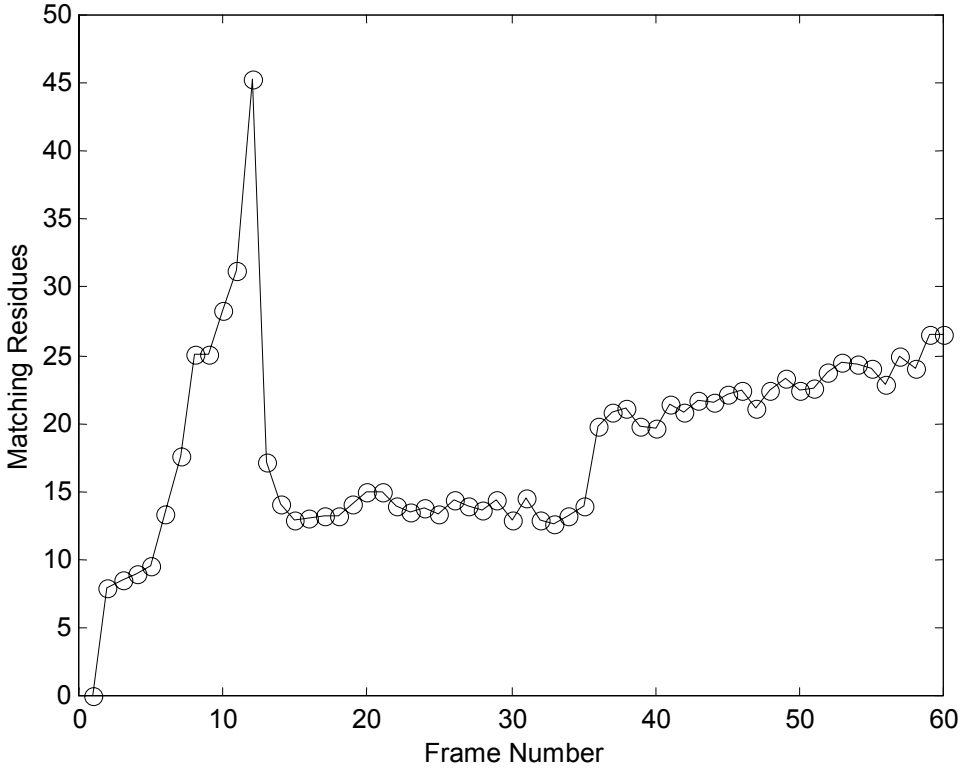
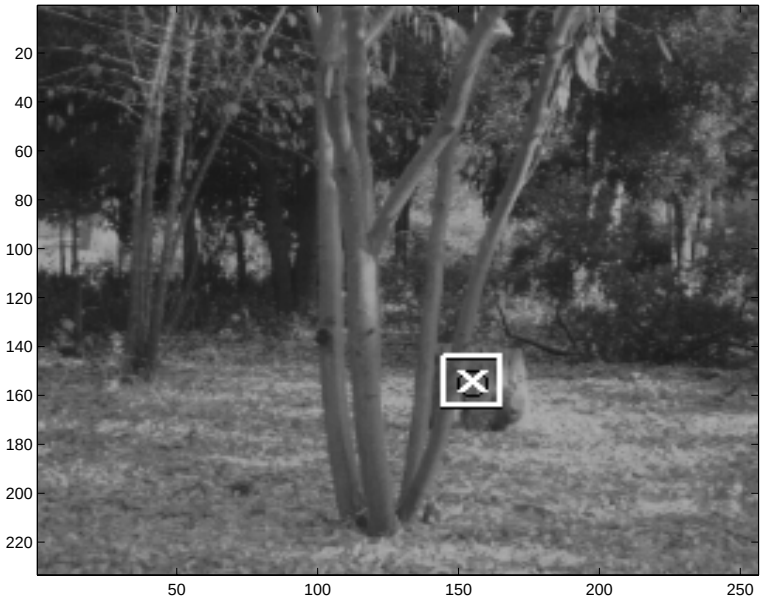


Figure 4.13. Matching residues for the tracked target in Figure 4.12. The occlusion starts at frame 8 and ends at frame 13.

(a)



(b)

Figure 4.14. Selected images after applying the tracking algorithm from image sequence taken with tree moving behind with a moving stump, which causes totally occlusion in the image sequence (a) First frame in sequence. (b) Frame 15. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.

(c)

(d)

Figure 4.14, continued. (c) Frame 31. (d) Frame 46. A window size is automatically selected initially that is well suited for the stump. During the occlusion event, the tracking continues with Kalman gains that are computed based on the large assumed covariance for the location of the target. A window size is automaticaly selected initially that is well suited for the desired target.

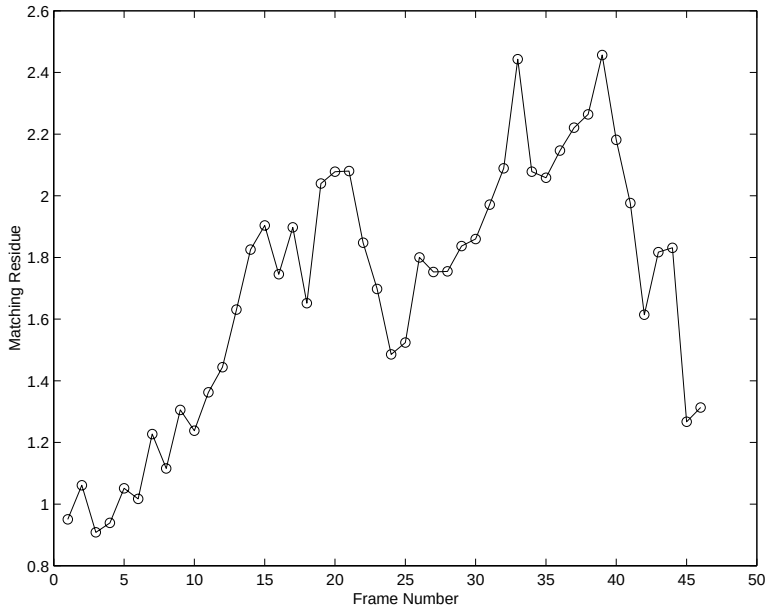


Figure 4.15. Matching residues for the tracked target in Figure 4.14. The matching residues are small although the object is partially occluded, and this is because the tracked object and the tree approximately have the same intensity.

Figure 4.16 shows another example of tracking a target over a monocular sequence of 14 frames. In this sequence, a car is running beside the truck as the car's driver intends to pass the truck and change lane. It is obvious that an occlusion will occur (the truck will come between the camera and the car). In Figure 4.16, the user has selected the target of interest in the first frame and window size has been selected adaptively in the first frame. After that, the Kalman filter provided estimates of new image locations, and each was used as the starting point for a correspondence search. Figure 4.17 shows the matching residue between the image frames. In this Figure, we can notice that starting from frame 10, the car changed lane and became hidden behind the truck. The matching residue increased as a result of the occlusion occurrence.

Figure 4.18 shows another example to track a target over a monocular sequence of 7 frames. In this figure, the user has selected the target of interest in the first frame and window size has been selected adaptively in the first frame. After that, the Kalman filter provided estimates of new image locations, and each was used as the starting point for a correspondence search. Figure 4.19 shows the matching residue between the image frames.

(a)

(b)

Figure 4.16. Selected images after applying the tracking algorithm from image sequence taken with an automobile driving behind a big truck, which causes total occlusion in the image sequence. (a) First frame in sequence. (b) Frame 2. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.

(c)

(d)

Figure 4.16, continued. (c) Frame 8. (d) Frame 14.The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method. a window size is automatically selected initially that is well suited for the desired target. During the occlusion event, the tracking continues with Kalman gains that are computed based on the large assumed covariance for the location of the target

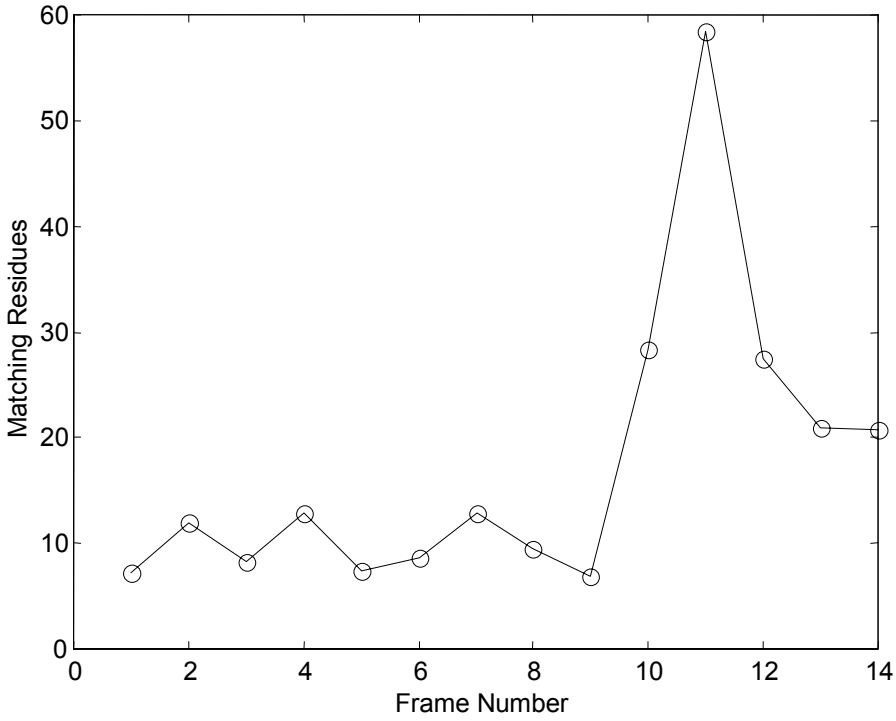
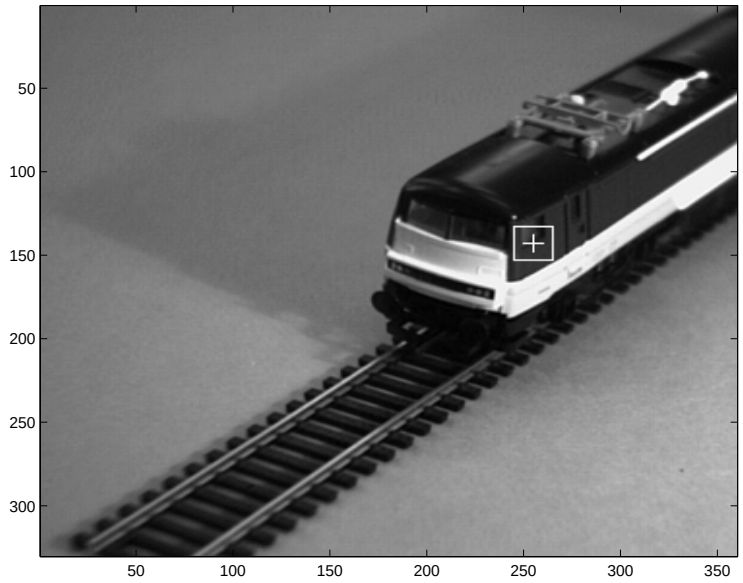
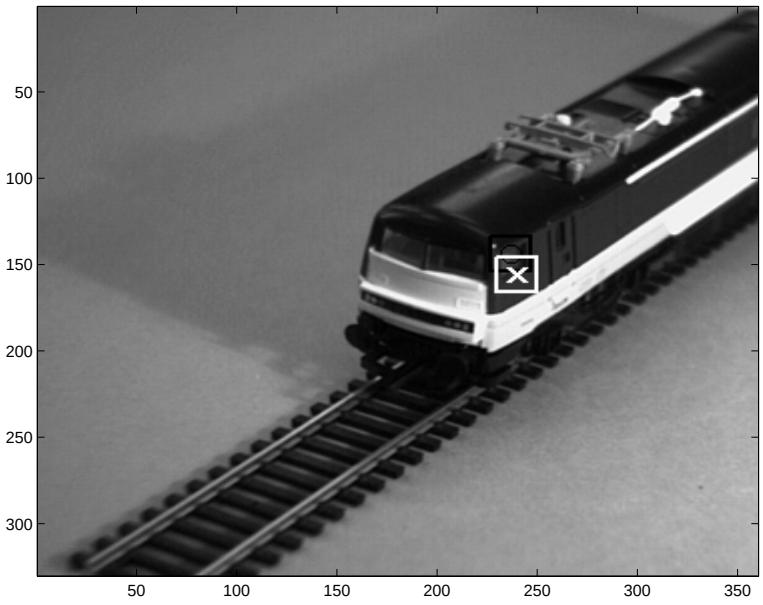


Figure 4.17. Matching residues for the tracked target in Figure 4.16. The occlusion starts at frame 10.

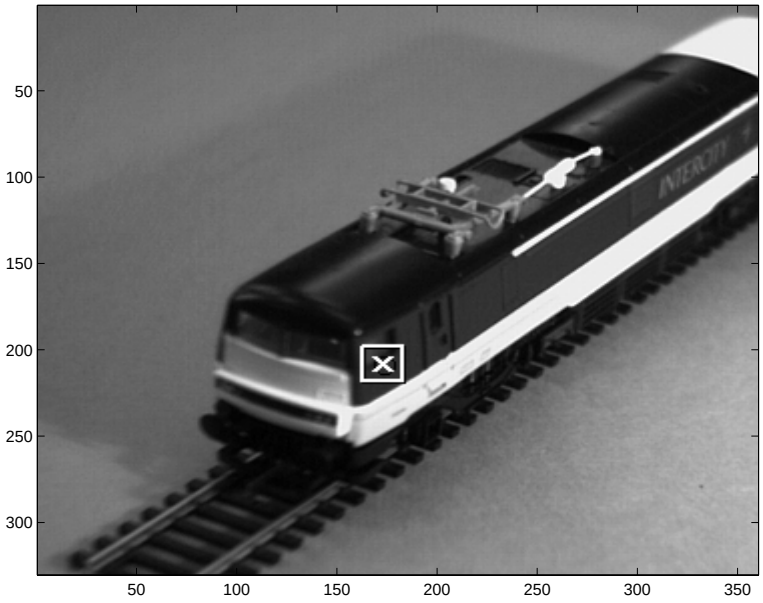


(a)

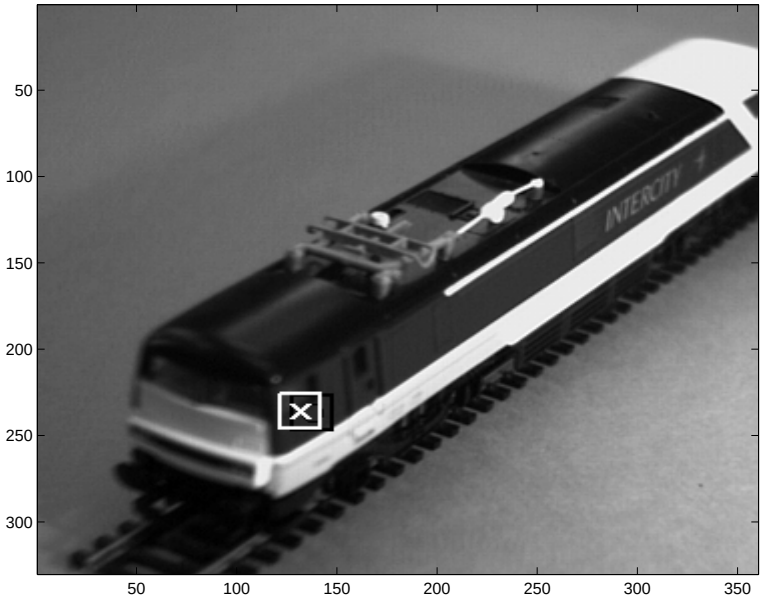


(b)

Figure 4.18. Selected images from "train" image sequence after applying the tracking algorithm. (a) First frame in sequence. (b) Frame 2. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



(c)



(d)

Figure 4.18, continued. (c) Frame 5. (d) Frame 7. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method. A window size is automaticaly selected initially that is well suited for the desired target.

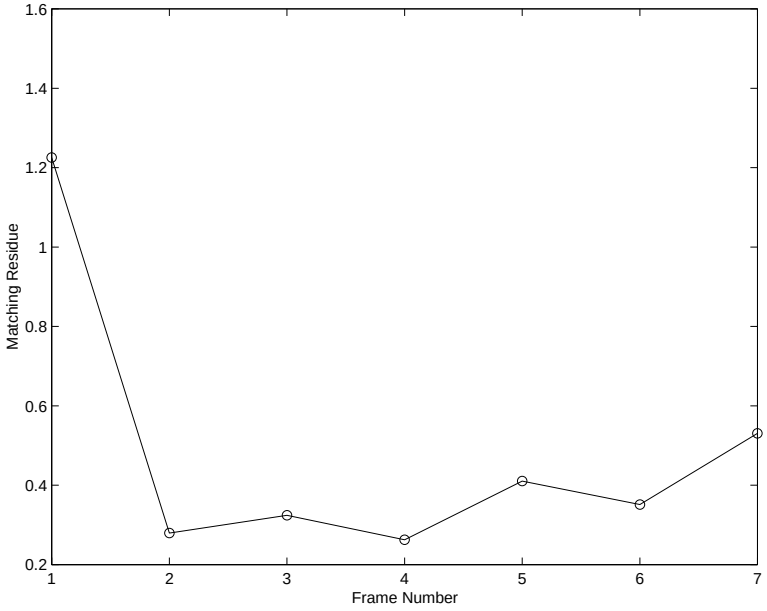
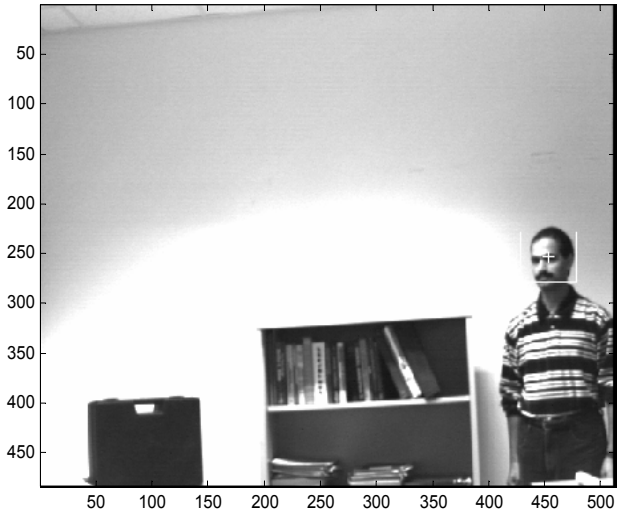
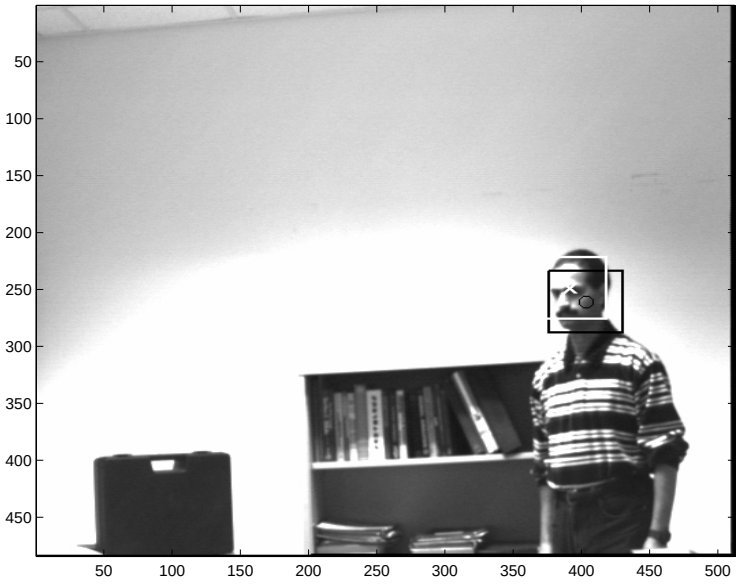


Figure 4.19. Matching residues for the tracked target in Figure 4.18.

Figure 4.20 shows another example of tracking a person walking through the laboratory over a monocular sequence of 12 frames in the field of view of the camera. In this sequence, the desired target was selected manually in the first frame. After that, the Kalman filter provided an estimate of the new image location, and it was used as the starting point for a correspondence search. We used an adaptive search window for the desired target in the first frame whose size is determined by the sudden increasing or decreasing in the first Maitra moment invariant. Figure 4.21 shows the matching residue between the image frames. We can notice that the tracked target has been well tracked. Figure 4.22 shows the detected trajectory of the walking person in the images generated by matching, together with the predicted locations generated by the Kalman filter.

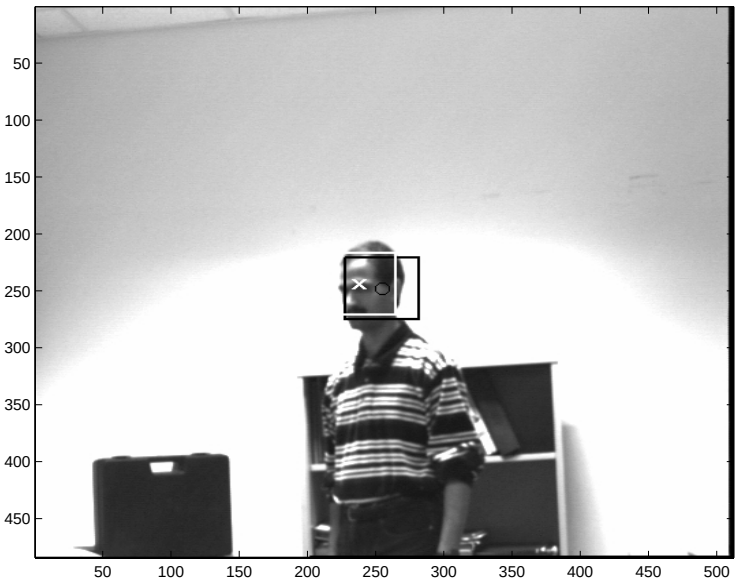


(a)



(b)

Figure 4.20. Selected images from "walking person" image sequence after applying the tracking algorithm. (a) First frame in sequence. (b) Frame 3. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



(c)

(d)

Figure 4.20, continued. (c) Frame 7. (d) Frame 12. The center of each black rectangle denotes a starting point predicted by the Kalman filter. The person walking in the field of view of a stationary camera.

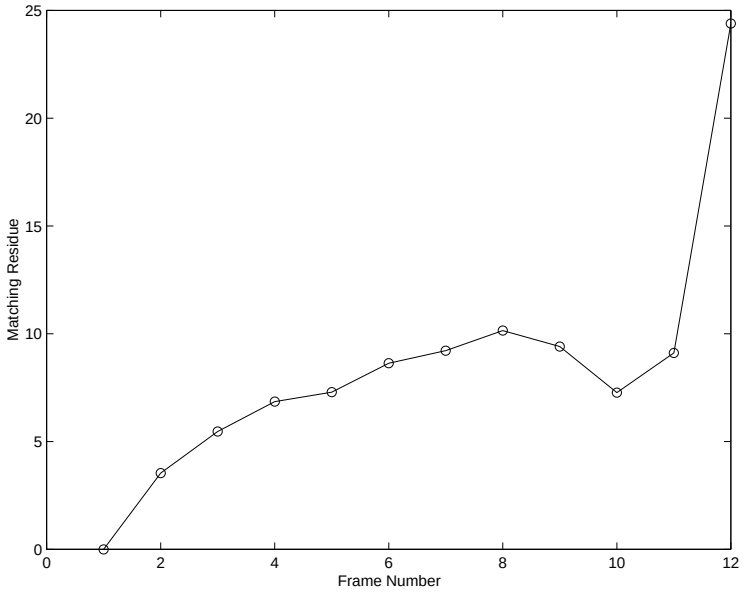


Figure 4.21. Matching residues for the tracked target in Figure 4.20. The residuals are quite small, and this is because the object is not occluded. At frame 12 the residual is large because the effect of noise due to lighting.

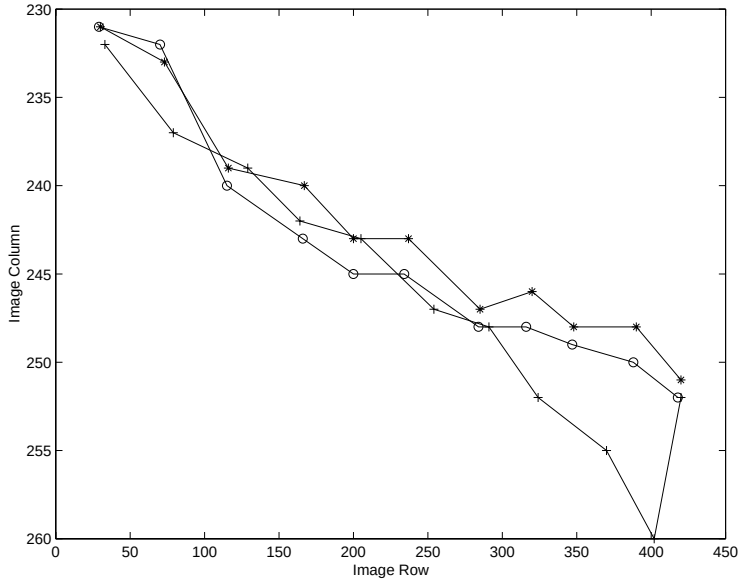


Figure 4.22. Actual trajectory (o), detected trajectory (*), and points predicted by the Kalman filter (+) for the tracked target Figure 4.20.

Chapter 5

Area-based Binocular Visual Tracking

5.1 Introduction

This chapter examines the problem of tracking a moving object using a stereo camera system. The monocular tracking algorithm is extended to work for binocular image sequences. This chapter also discusses how to handle the occlusion of the moving object in the binocular image sequence. Finally, this chapter reports the results obtained from applying the proposed algorithm to real stereo pair image sequences.

5.2 Overview of the Binocular Tracking System

There are good reasons to track a moving object using more than one camera (stereo). There are always errors in tracking a moving object using one camera, and the use of multiple cameras usually permits a more accurate estimation of the location of the tracked object by providing redundant information about the tracked object.

The proposed binocular tracking system is flowcharted in Figure 5.1. This algorithm operates in four phases, namely image acquisition, image matching, prediction, and gaze control. In the image acquisition phase, the cameras acquire new stereo pair images for the moving object. In the matching phase, two matches are performed – one between the first and the current frame from the dominant (left) camera and a second match between the first frame from the left camera and the current frame in the right camera. In the prediction phase, the feedback loop is established by predicting new locations with Kalman filters and using them to guide the extraction of the corresponding locations in the next stereo pair images. Finally, the control system uses the visual measurements of the target position in the images to drive the pan/tilt units to keep the target near the center of the images.

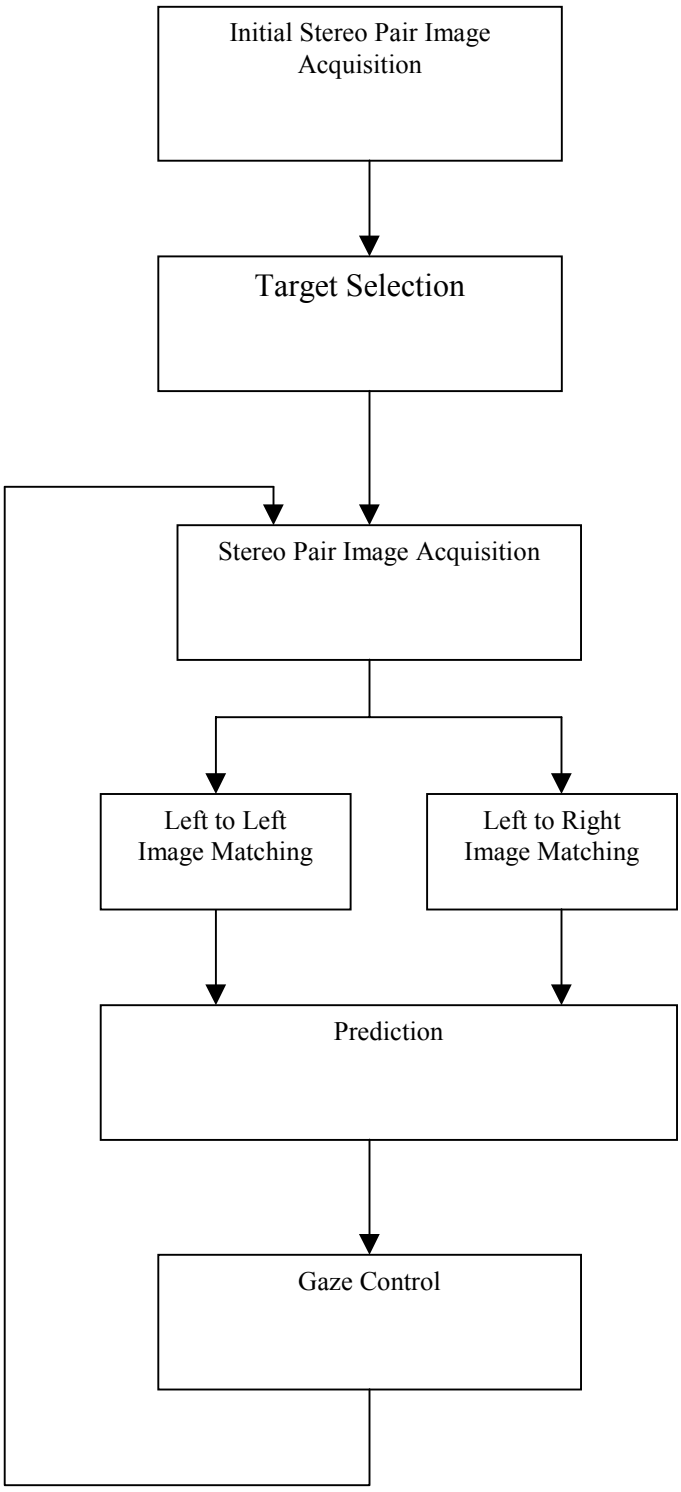


Figure 5.1. Function block diagram of the binocular active tracking system

5.3 Binocular Fixation and Matching

Binocular fixation, as the term is used here, is the process of directing the gaze of two cameras to a single location of interest in a three-dimensional (3D) scene. Recent progress in the development of motorized camera positions has spurred interest in the areas of visual target selection [Abbo92, Brun94] and visual tracking [Coom92, Maki93, Shi94, Reid95]. However, before stereo tracking can be effective, both cameras need to be fixated at the same object point. Typically this requires the estimation of binocular disparity for the target of interest, followed by vergence and version movements of the two cameras (or eyes) to fixate the target. This process is referred to as initial fixation when disparities must be estimated within new or unexpected visual surroundings.

Initial fixation is inherently difficult because, fundamentally, the well-known stereo correspondence problem is encountered. Nearly all-stereo matching algorithms rely on an initial depth/disparity estimate and accurate knowledge of camera calibration parameters (or, equivalently, they are provided with images that have been rectified).

Most previous work on binocular fixation and vergence control has relied on such cues as motion [Reid95], focus-based range estimation [Ahuj93, Krot89], or an assortment of cues [Clar88] to assist in system initialization. If such cues are absent, then the problem becomes much more difficult. Olson and others [Olso89, Olso91, Tayl94] used cepstral filtering with some success in such cases, under the assumption that one image of the stereo pair is a shifted version (echo) of the other. Abbott and Zheng [Abbo95] developed an approach that employs attentional shifts and image warping to estimate the needed disparity for a given target.

In the matching phase, two matches are performed – one between the first and the current frame from the dominant (left) camera and a second match between the first frame from the left camera and the current frame in the right frame in right camera (see Figure 5.2).

The monocular affine motion model described in Chapter 4 is extended to represent the image motion for binocular image sequences. An affine motion field may now be represented as

$$U_{LL} = D_{LL}X_L + d_{LL} \tag{5.1}$$

$$U_{LR} = D_{LR}X_L + d_{lR} \tag{5.2}$$

where $X_L = \begin{bmatrix} x_L \\ y_L \end{bmatrix}$ is a location in the first frame in left camera, $U_{LL} = \begin{bmatrix} u_{LL} \\ v_{LL} \end{bmatrix}$ is the displacement vector between the first frame in the left camera and the current frame in the left camera, and $U_{LR} = \begin{bmatrix} u_{LR} \\ v_{LR} \end{bmatrix}$ represents the displacement between the first frame in the left camera and the current frame in the right camera. Similarly, D_{LL} is the deformation matrix between the first frame in the left camera and the current frame in the left camera, D_{LR} is the deformation matrix between the first frame in the left camera and the current frame in the right camera, d_{LL} is the translation vector of the center of the window of interest between the first frame in the left camera and the current frame in the left camera, and d_{LR} is the translation vector of the center of the window of interest the first frame in the left camera and the current frame in the right camera. Then, a point X_L in left image I_L of the first stereo pair image moves to point $A_{LL}X_L + d_{LL}$ in the left image J_L in the second stereo pair image, and a point X_L in the left image I_L of the first stereo pair image has corresponding point $A_{LR}X_L + d_{LR}$ in the right image J_R in first stereo pair image, where $A_{LL} = I + D_{LL}$, $A_{LR} = I + D_{LR}$ and I is the 2x2 identity matrix *:*

$$I_L(X_L) = J_L\left(A_{LL}X_L + d_{LL}\right). \tag{5.3}$$

$$I_L(X_L) = J_R\left(A_{LR}X_L + d_{LR}\right). \tag{5.4}$$

Given two stereo image pairs $(I_L,\ I_R)$ and $(J_L,\ J_R)$ in a motion sequence and a window in image I_L, binocular tracking means repeatedly determining the twelve scalar values that appear in the deformation matrices D_{LL} and D_{LR}, and the displacement vectors d_{LL} and d_{LR}.

In the area-based tracking techniques, there are two mechanisms to apply the matching. The first mechanism considers the pattern of the image window around the desired target point at time t as the reference pattern for searching for the displacement at time $t + \Delta t$. This mechanism of updating the reference pattern at every sampling instant

tends to the accumulation error. This accumulation error results from the accumulation of the error in measuring the displacement of the tracked image pattern. The second mechanism considers the pattern of the image window around the desired target point in the first frame as the reference pattern for searching the displacement in the subsequent image frame. The second matching mechanism is used in the proposed tracking system to avoid the problem of the accumulation error.

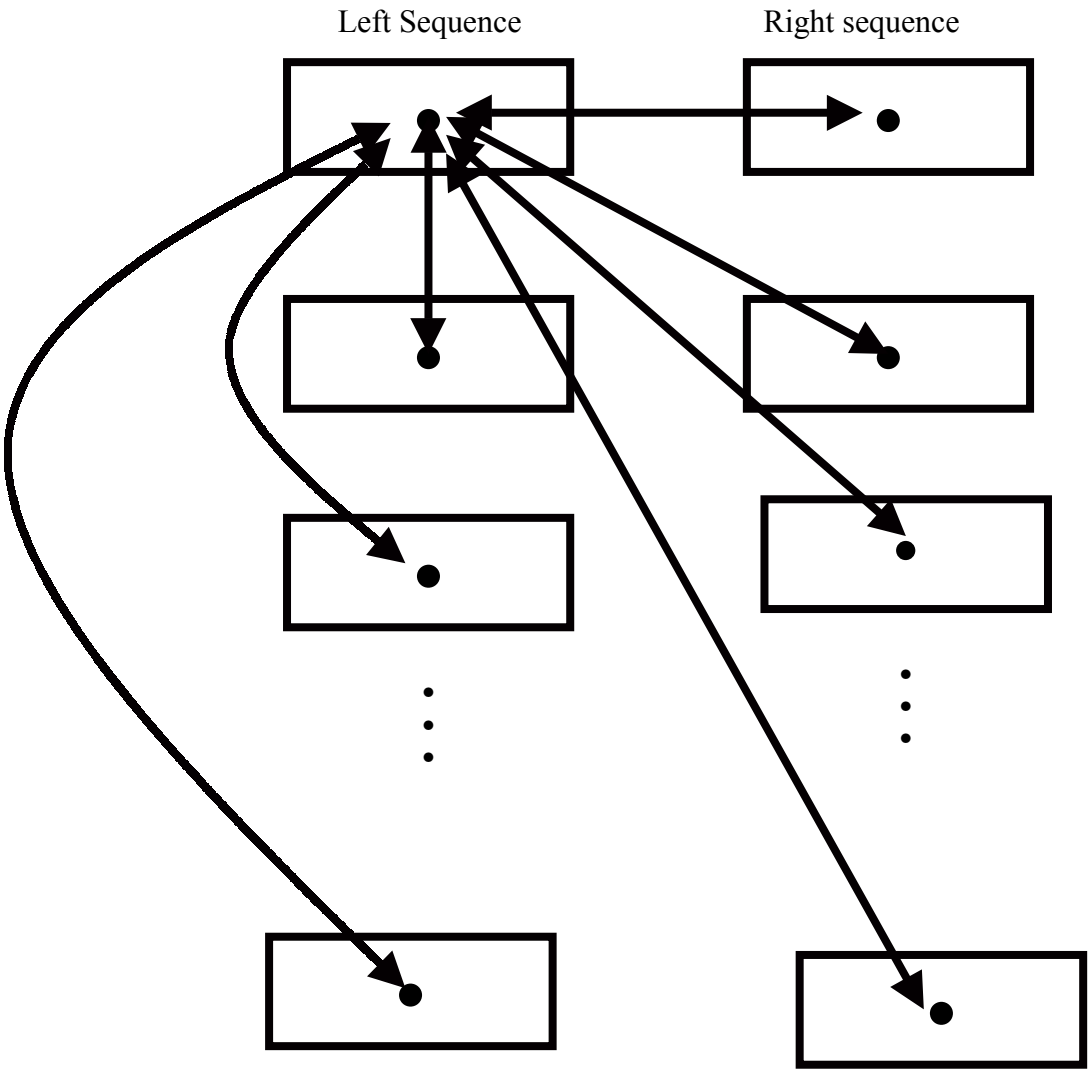


Figure 5.2. Matching in the binocular tracking system

5.4 Binocular Motion Prediction by Kalman Filtering

In Chapter 4, the Shi-Tomasi search technique was adapted to work for monocular tracking with large interframe movements. In this chapter, the monocular algorithm is extended to work for binocular image sequences. The fact that 2D parameters are found means that difficult 3D recovery is not required, and careful camera calibration is not needed.

This section describes how to use a constant-image-velocity Kalman filter to perform this prediction for binocular image sequences. A state model that is linear and is defined by the following equations is assumed:

$$X_{k+1}=\varphi_k X_k+W_k \tag{5.5}$$

$$Z_k=H_k X_k+V_k \tag{5.6}$$

At time k, X_k is the system state vector, φ_k is the state transition matrix, W_k is the system noise vector, Z_k is the measurement vector (obtained by the matching process), H_k is the matrix relating the state vector to the measurement vector, and V_k is the measurement noise vector. In binocular implementation, the state vector is defined as

$$X_k=\begin{bmatrix} x_L \\ y_L \\ x_R \\ y_R \\ \dot{x}_L \\ \dot{y}_L \\ \dot{x}_R \\ \dot{y}_R \end{bmatrix}$$ where (x_L, y_L) and $(\dot{x}_L, \dot{y}_L)$ represent the 2-dimensional position and velocity for the target in the left image, and (x_R, y_R) and $(\dot{x}_R, \dot{y}_R)$ represent the position and velocity for the target in the right image. The state therefore represents two matches – one between the first and the current frame from the dominant (left) camera and a second between the first frame from the left camera and the current frame in the right frame. The measurement vector is defined as

$Z_k = \begin{bmatrix} x_L \\ y_L \\ x_R \\ y_R \\ \dot{x}_L \\ \dot{y}_L \\ \dot{x}_R \\ \dot{y}_R \end{bmatrix}$, where the first four values are the result of two separate matching processes as described in the previous section and the second four values represent the velocity for the target in both left and right images.

In this system, the five Kalman filter equations are implemented in a straightforward manner with φ and H matrices given by

$$\varphi_k = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

The initial values for the system noise matrix Q_k and the measurement noise matrix R_k in this system were chosen as the identity matrix.

5.5 Gaze Control

This section describes how to utilize state estimate of a Kalman filter in a controlling pan-tilt unit to move the cameras. In the motor-control module for the binocular case, the control system will use the visual measurements of the target position in the image to drive the pan-tilt unit to keep the target near the center of an image frame. This information is obtained by the matching of two images with the help of Kalman filter. The pan-tilt unit is controlled from the hosts via a serial port.

The estimated location values (x_L, y_L), and (x_R, y_R) of $\widetilde{X}_k^-$ will then be subtracted from the desired target location, which is at the center of the image. The

difference values Δx_k, and Δy_k will then be converted to pan and tilt steps by which a pan-tilt unit controller is to be rotated.

As an illustration of such an operation, let us assume that (x_T, y_T) denotes the desired target location at the first frame in the left camera, (x_s, y_s) denotes the estimated location of the target to start search in frame captured at time $(t + \Delta t)$, and $(\Delta x(t), \Delta y(t))$ denotes the estimated amount of displacement that will occur during the period of Δt. In order to place the target at the center of the next frame, the pan-tilt unit needs to rotate a camera by panning steps and tilt steps. The extent of these steps can be computed as follows:

Pan steps = $\Delta x * \lambda_x$

Tilt steps = $\Delta y * \lambda_y$

where λ_x, and λ_y are calibration factors for the camera in pan and tilt, respectively, as described in Chapter 3.

5.6 Occlusion Handling

The visual tracking method described here is improved to detect the occlusion based on the residual error computed by the matching approach. If the residual matching error exceeded a user-defined threshold, this means that the tracked object may be occluded with another object. When occlusion is detected, tracking continues with the predicted locations based on Kalman filtering. If both cameras can see the object, the object is not occluded. If the object is not visible by either the left or right camera, the object is partially occluded from that camera (see Figure 5.2). Meanwhile, if neither camera can see the object, the object is completely occluded.

If the matching residue of either the left or right camera exceeds the user-defined threshold, then the object is not visible by that camera, which means that the object is partially occluded. In this case, the corresponding elements in the Z_k represent the occluded measurements. Hence, these measurements are converted to zero value by pre-multiplying with S_k, as described in detail in Chapter 4. Similarly, if the matching residues for both the left and right cameras exceed the user-defined threshold, then the

object is not visible and is totally occluded. In this case, the corresponding elements in the Z_k represent the occluded measurements. Hence, these measurements are converted to zero value by pre-multiplying with S_k.

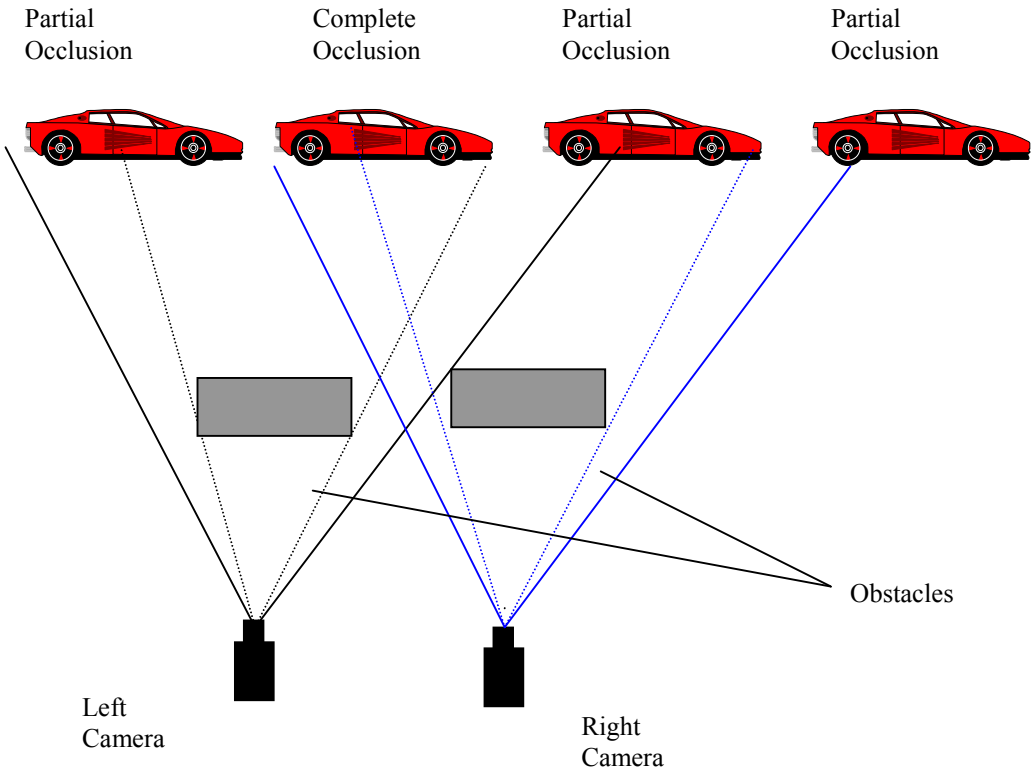
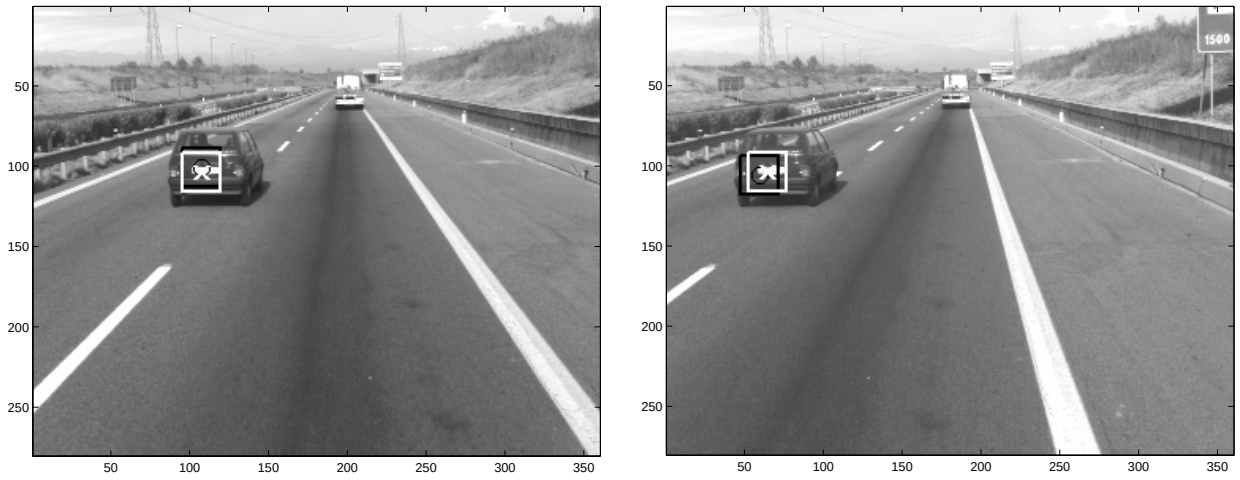


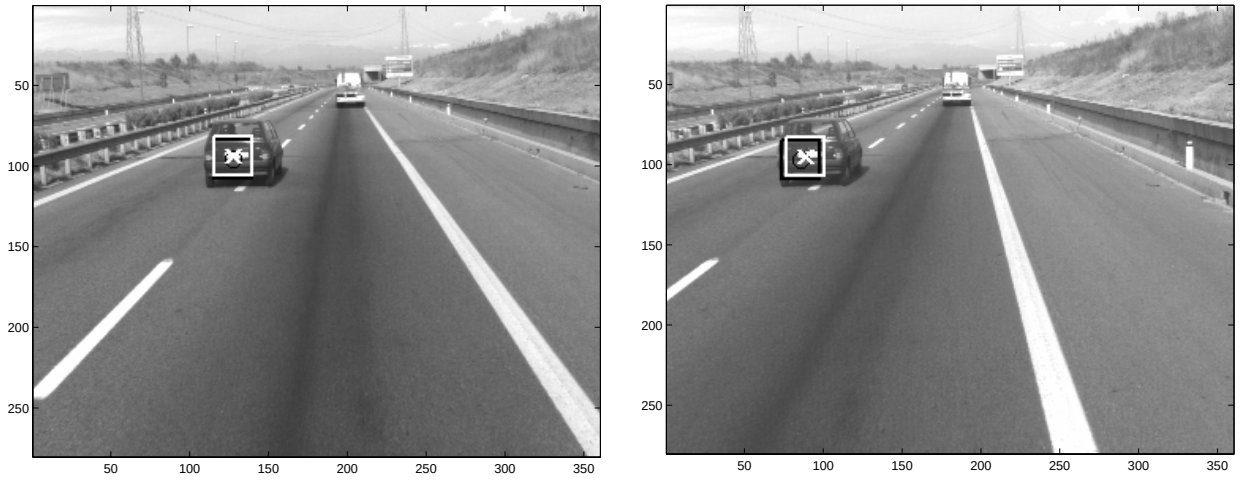
Figure 5.3 Occlusion in the binocular tracking system.

5.7 Experimental Results Using Binocular Image Sequences

This section reports the results obtained by applying the proposed algorithm to real stereo pair image sequences. The binocular truck, and car image sequences were downloaded from the computer vision home page to test the proposed tracking system [Visi99]. Figure 5.4 presents an example of successful tracking. The system tracks an automobile in a stereo image sequence over 15 frames. The system was initialized by manual selection of corresponding points for a target in the initial left and right images, and in the subsequent left image. Figure 5.5 shows the matching residue for the left image sequence together with the right image sequence. In this case, the errors are smaller than for the monocular cone sequence (see Section 4.7), primarily because the target's motion away from the cameras is relatively small.

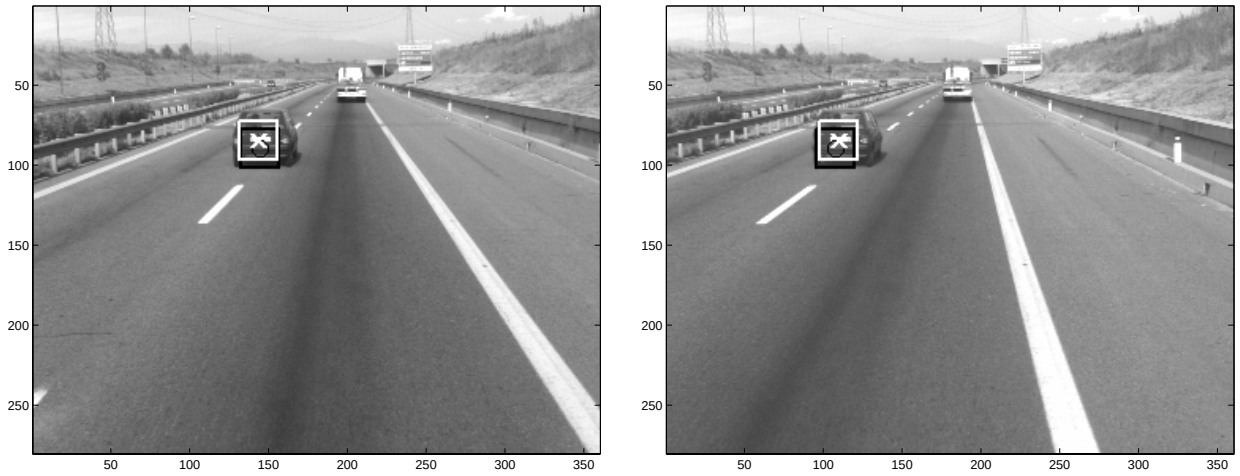


(a)

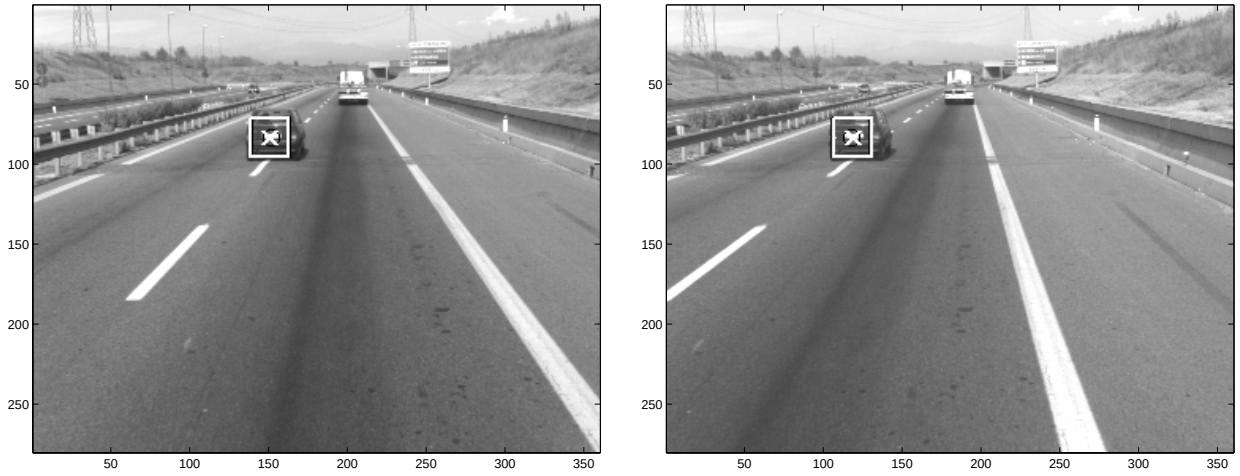


(b)

Figure 5.4. Selected images from stereo "road" image sequence after applying the tracking algorithm. (a) First image pair in sequence. (b) Image pair 5. The center of each black square denotes a point predicted by the Kalman filter. Each white "×" denotes the target detected by the correspondence search method.



(c)



(d)

Figure 5.4, continued. Selected images from stereo "road" image sequence after applying the tracking algorithm. (a) First image pair in sequence. (c) Image pair 10. (d) Image pair 14.

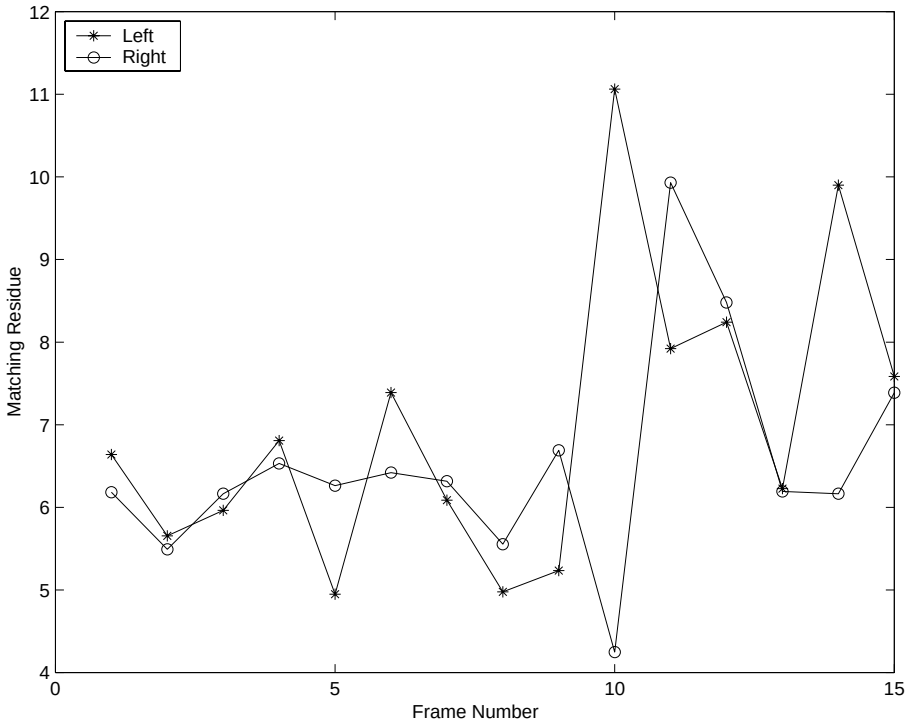
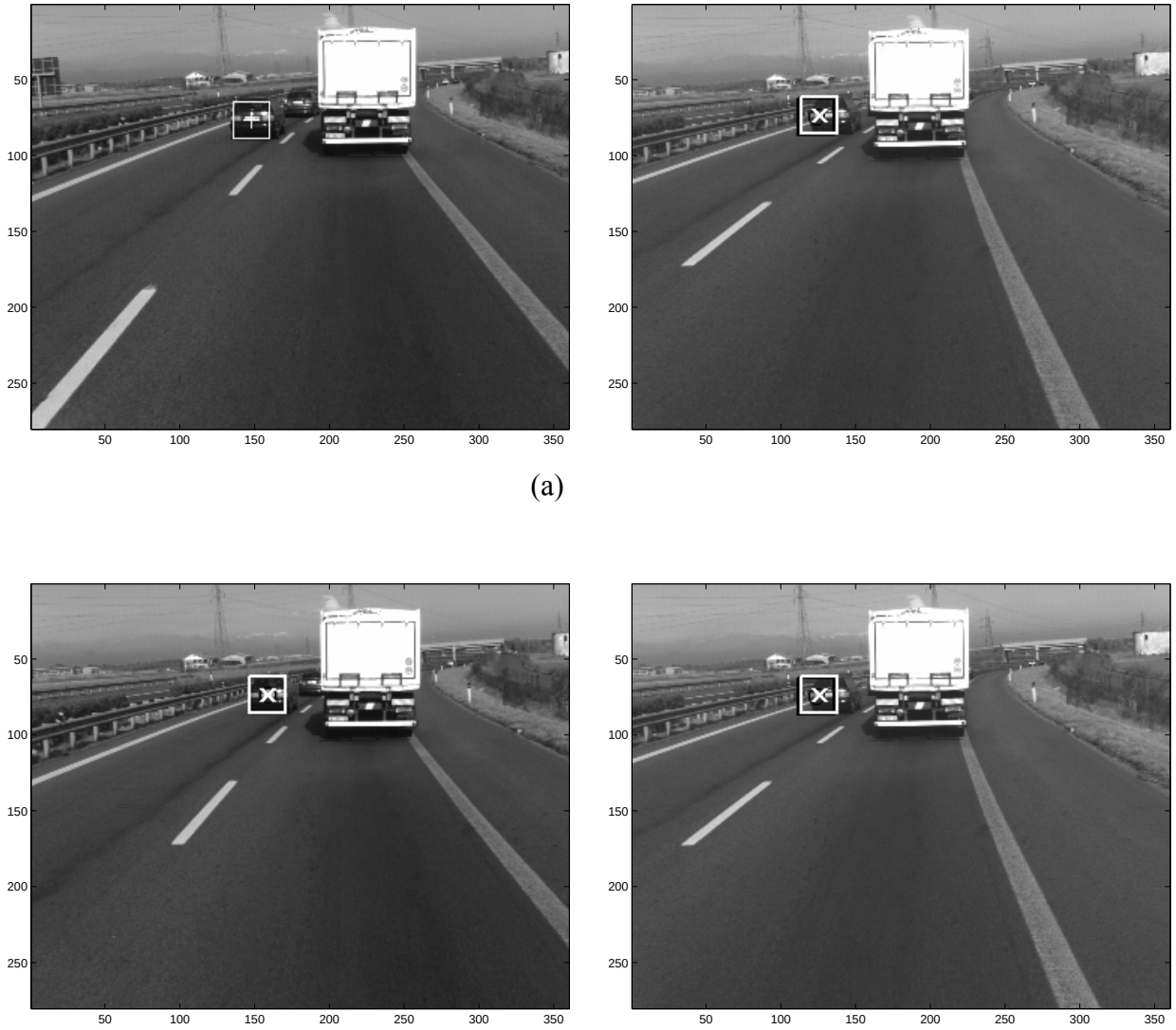


Figure 5.5 Matching residues for the tracked target in Figure 5.4. Each "o" denotes the left to left matching residue for the tracked target. Each "∗" denotes the left to right matching residue for the tracked target.

Figure 5.6 presents another example of successful tracking in case of occlusion. The system tracks an automobile in a stereo image sequence over 14 frames. The system was initialized by manual selection of corresponding points for a target in the initial left and right images, and in the subsequent left image. In this example, the automobile is going to change lanes and become occluded by a big truck. Figure 5.7 shows the matching residue for the left image sequence together with the right image sequence.

In another binocular road image sequence, the system tracks a car moving on a highway (see Figure 5.8). The system was initialized by manual selection of corresponding points for a target in the initial left and right images, and in the subsequent left image. Once this car goes under the bridge, it will enter a shadow zone. The matching residue increases as shown in Figure 5.9 and the system detects an occlusion. In this case, the system continues to track the car based upon the predicted locations by Kalman filter, not by the detected locations by matching. Once the car passes the shadow zone, the matching residue decreases to be quite small again and the system tracks the car based upon the detected locations by matching. In Figure 5.9, it can be noticed that the car

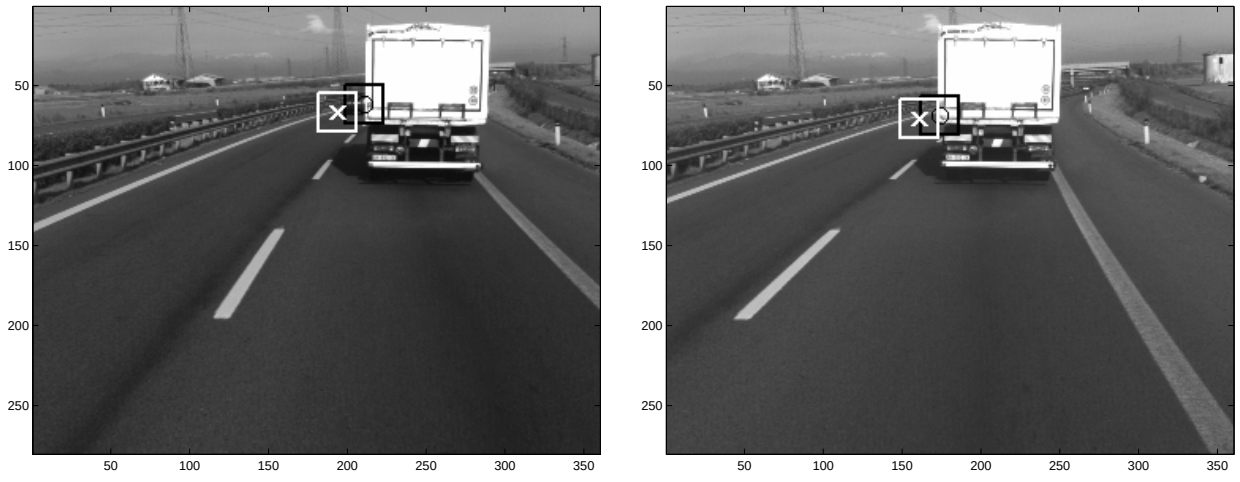
entered the shadow zone in the right camera view starting at frame 11, and in the left camera view starting at frame 13.



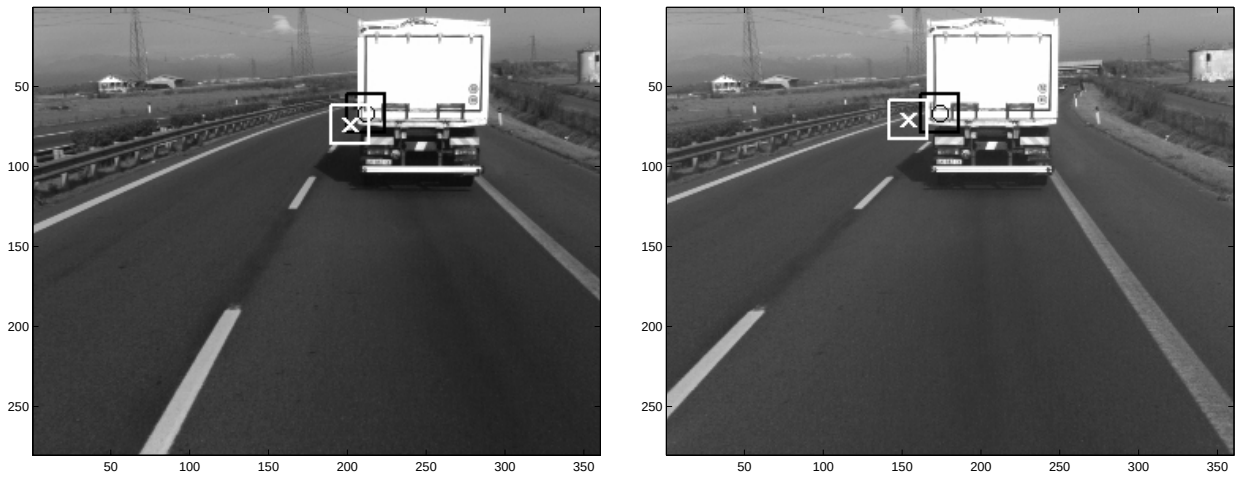
(a)

(b)

Figure 5.6. Selected images from stereo "road" image sequence after applying the tracking algorithm taken with an automobile driving behind a big truck that causes total occlusion. (a) First image pair in sequence. (b) Image pair 3. The center of each black square denotes a point predicted by the Kalman filter. Each white "×" denotes the target detected by the correspondence search method.



(c)



(d)

Figure 5.6, continued. Selected images from stereo "road" image sequence after applying the tracking algorithm. (a) First image pair in sequence. (c) Image pair 9. (d) Image pair 14.

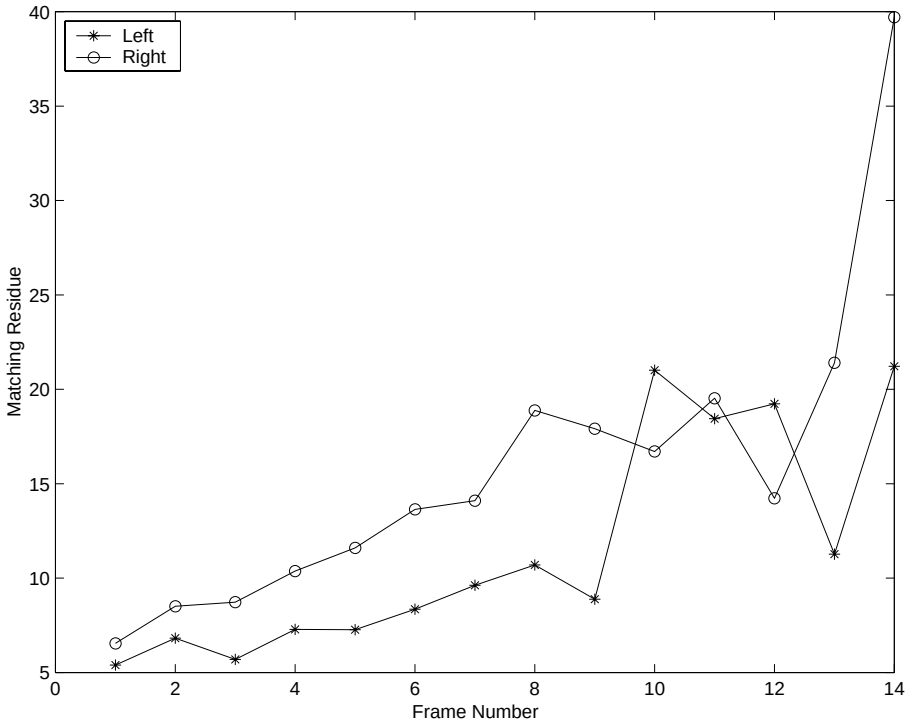
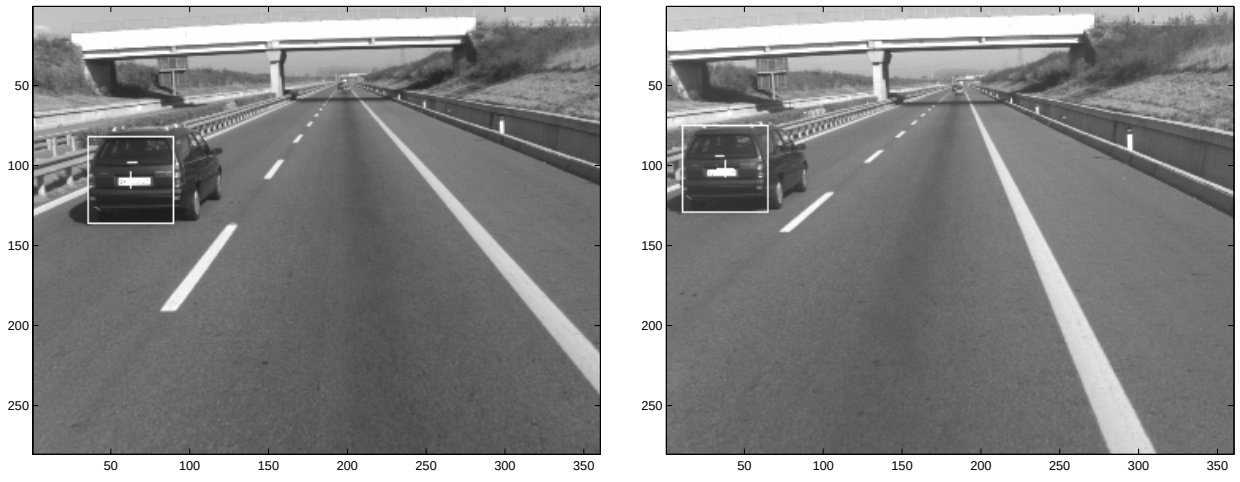
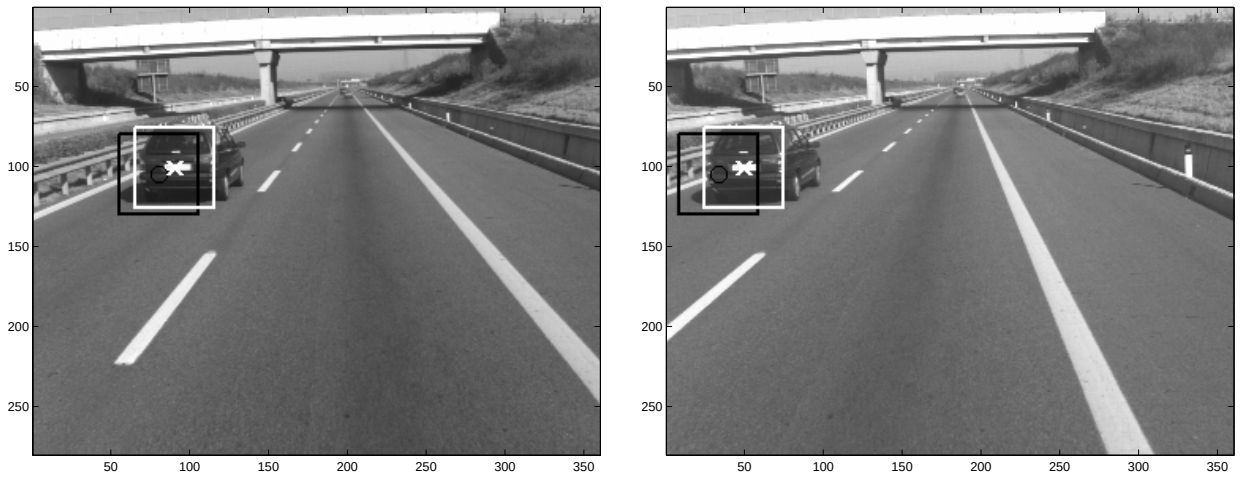


Figure 5.7. Matching residues for the tracked target in Figure 5.6. Each "o" denotes the left to left matching residue for the tracked target. Each "∗" denotes the left to right matching residue for the tracked target.

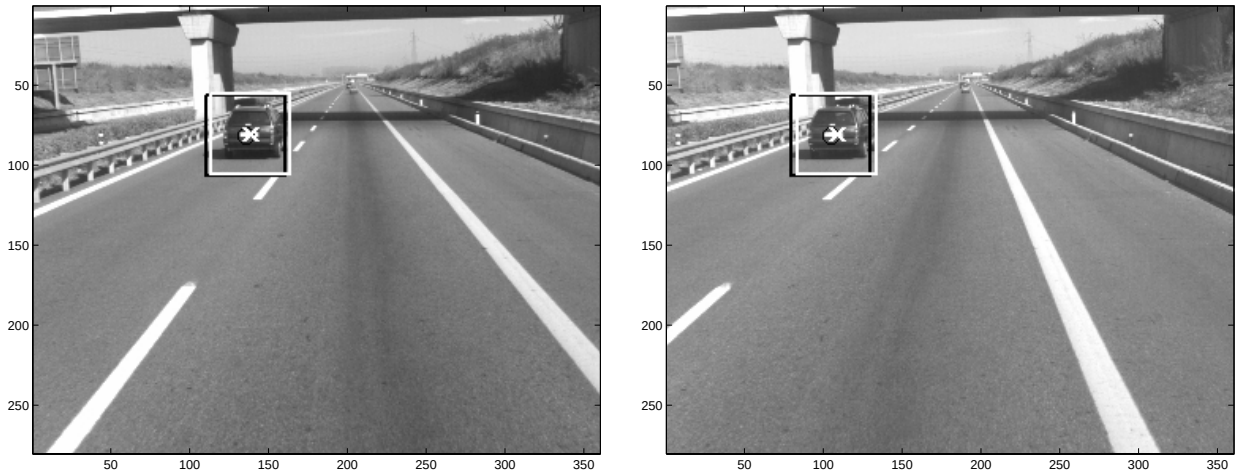


(a)

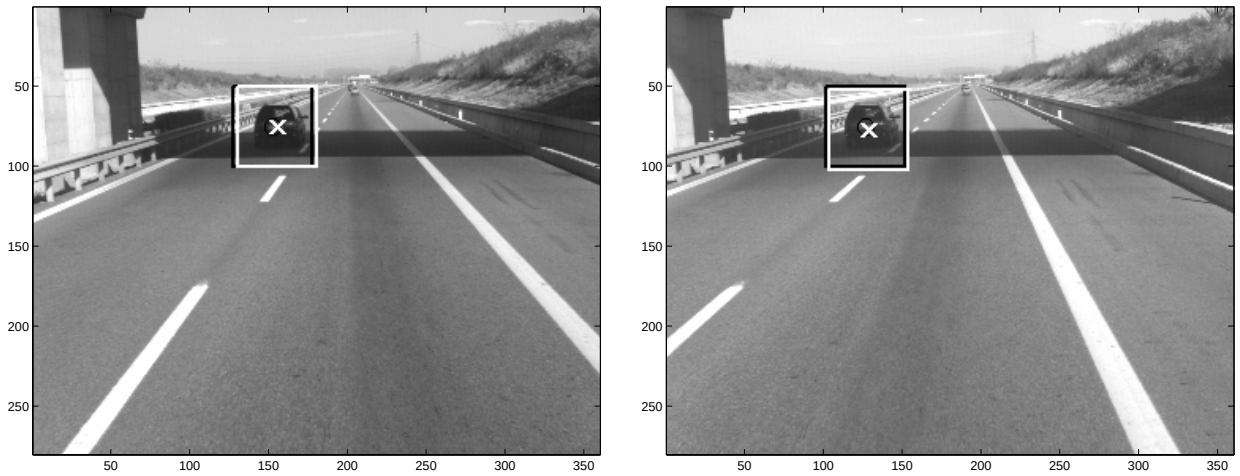


(b)

Figure 5.8. Selected images from stereo "road" image sequence after applying the tracking algorithm taken with an automobile driving toward a bridge that causes total occlusion. (a) First image pair in sequence. (b) Image pair 3. The center of each black square denotes a point predicted by the Kalman filter. Each white "×" denotes the target detected by the correspondence search method.

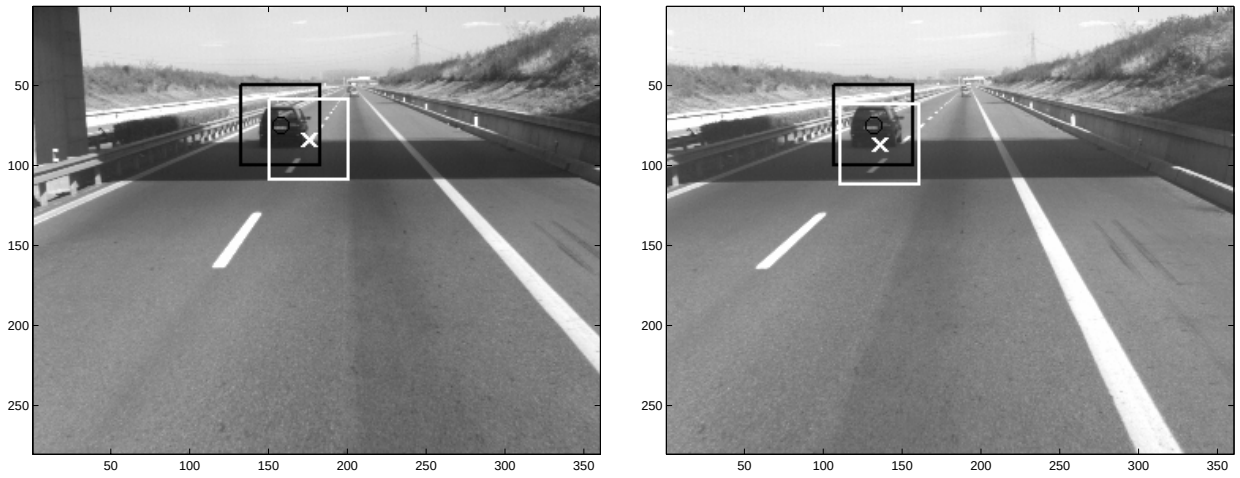


(c)

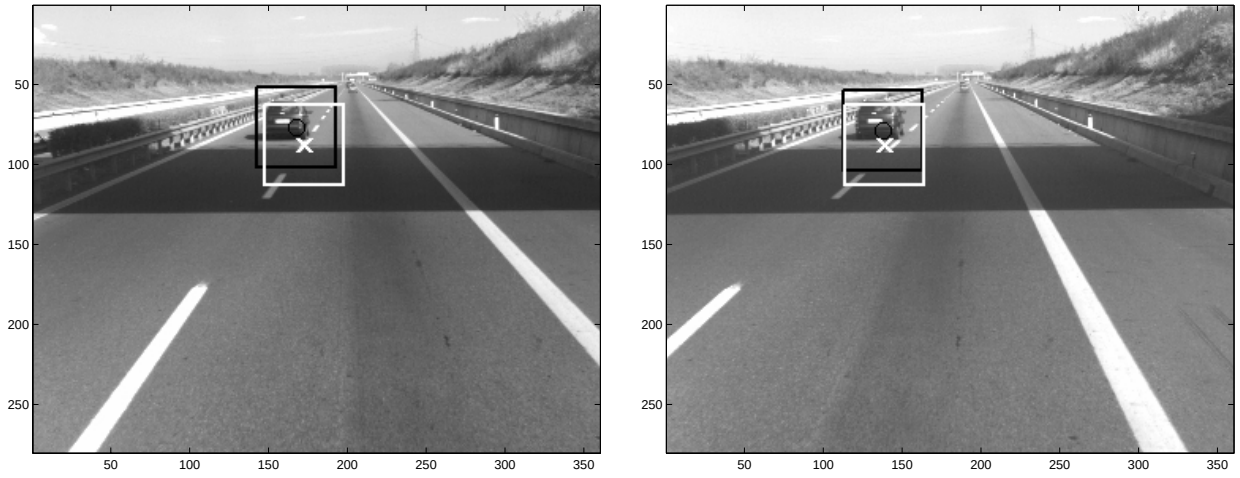


(d)

Figure 5.8, continued. Selected images from stereo "road" image sequence after applying the tracking algorithm. (c) Image pair 10. (d) Image pair 14.



(e)



(f)

Figure 5.8, continued. Selected images from stereo "road" image sequence after applying the tracking algorithm. (e) Image pair 15. (f) Image pair 16.

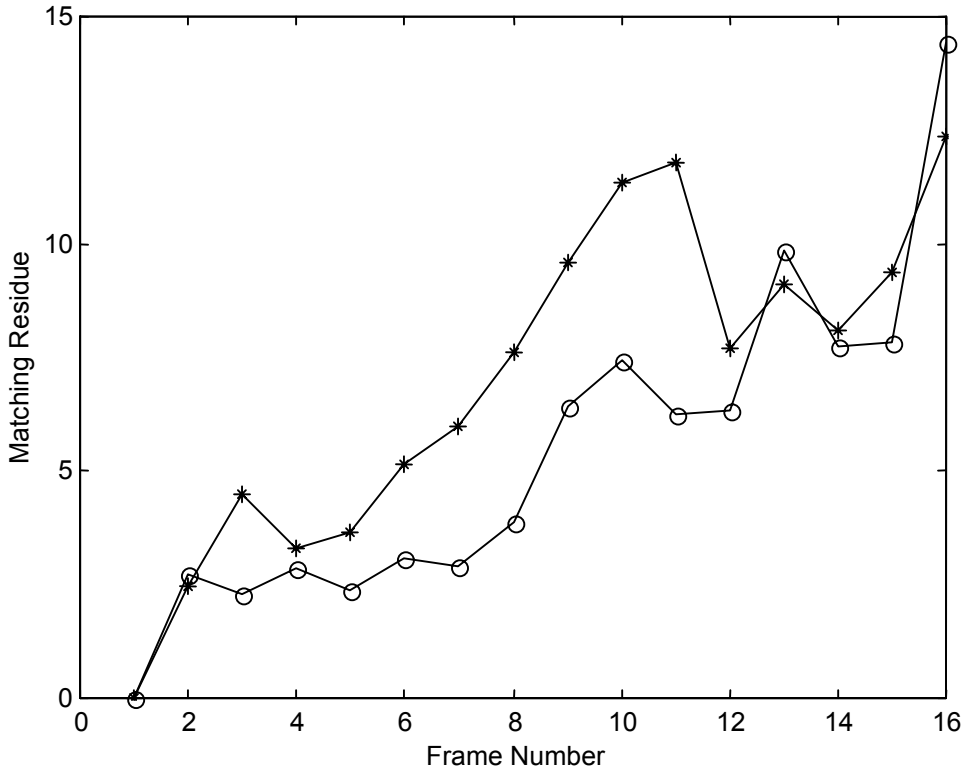


Figure 5.9 Matching residues for the tracked target in Figure 5.8. Each "o" denotes the left to left matching residue for the tracked target. Each "*" denotes the left to right matching residue for the tracked target.

Chapter 6

Moment-based Window Size Selection

6.1 Introduction

A central problem in area-based stereo matching lies in selecting an appropriate window size. The window size must be large enough to include enough intensity variation for reliable matching, but small enough to avoid serious problems from projective distortion or occlusion. This chapter introduces the definitions of two-dimensional moments, summarizes the Hu moment invariants, and describes affine moment invariants. A novel approach based on Hu moment invariants for window size selection is developed in Section 6.3 to guide the automatic window size selection.

This chapter also considers the effect of spatial quantization on several moment invariants. Of particular interest are the affine moment invariants, which have emerged in recent years as a useful tool for image reconstruction, image registration, and recognition of deformed objects. Traditional analysis assumes moments and moment invariants for images that are defined in the continuous domain. In practice, however, the digitization process introduces errors that violate the invariance assumption. Section 6.4 presents an analysis of quantization-induced error on (two-dimensional) Hu moment invariants and affine moment invariants. Error bounds are given in several cases.

6.2 Two-dimensional Moment Invariants

6.2.1 Basic Definitions

Two-dimensional moments of order *(p+q)* for a function $f : \Re^2 \to \Re$ are defined as

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy \ , \qquad (p, q = 0, 1, 2,...). \tag{6.1}$$

It should be noted that the moments in (6.1) are not in general invariant under translation, rotation, or scale changes in the domain of *f*. Translation invariance can be achieved by using central moments, which are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy\ , \qquad (p, q = 0, 1, 2, ...) \tag{6.2}$$

where $\bar{x} \equiv m_{10}/m_{00}$ and $\bar{y} \equiv m_{01}/m_{00}$. Scale invariance can then obtained through normalization,

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{6.3}$$

where $\gamma = \frac{p+q}{2} + 1$ and $p + q = 2, 3...$

6.2.2 Hu Moment Invariants

Hu [Hu62] introduced seven functions of second and third moments that are invariant to translation, scale and rotation in 2 dimensions. The first four Hu moment invariants are given by

$$\phi_1 = \eta_{20} + \eta_{02} \tag{6.4}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{6.5}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{6.6}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{6.7}$$

where η_{pq} are the normalized central moments defined in (6.3). All but the first invariant ϕ_1 vanish if the image is symmetric with respect to the line $x = y$.

Moment invariants represent a complete set of image features, and are therefore of fundamental importance for pattern recognition. Invariance is achieved only in the continuous domain, however, and higher-order invariants are often too sensitive to noise to be of practical use for image analysis [Prok92, Sala98a].

6.2.3 Affine Moment Invariants

The Hu invariants of the previous section are invariant under changes in translation, rotation, and scale, but not under general 2-D affine transformation. Affine moment invariants (AMIs) were introduced in [Flus93] to address this, and a comprehensive discussion of their properties can be found in [Flus94b]. The first two AMIs are given below, and have been successfully used for pattern recognition applications. However, no analysis of quantization error has been found in the literature.

$$I_1 = \frac{1}{\mu_{00}^4}\left(\mu_{20}\,\mu_{02} - \mu_{11}^2\right) \tag{6.8}$$

$$I_2 = \frac{1}{\mu_{00}^{10}}\left(\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\,\mu_{12}^3 + 4\mu_{03}\,\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2\right) \tag{6.9}$$

In the remainder of this section, we represent the affine transformation using the following notation (see Appendix A),

$$\hat{x} = a_1 x + a_2 y + d_1 \tag{6.10}$$

$$\hat{y} = a_3 x + a_4 y + d_2$$

where $(x,\ y)$ and $(\hat{x}, \hat{y})$ are coordinates in the image plane before and after the transformation, respectively. The constants d_1 and d_2 represent translation, and the constants a_1, a_2, a_3, and a_4 determine rotation, scale, and skew of the transformed image.

6.3 Window Size Selection

6.3.1 Related Work

The ability to perform image matching is of fundamental importance for many higher-level tasks. Visual tracking and stereo matching, in particular, rely on the ability to determine corresponding points in separate images. (For example, see [Reid93, Fusi97, Mank97].) The choice of window size has a profound effect on the results obtained during image matching. In a visual tracking system, for example, a window that is too large may overlap several occluded objects and lead to false matches.

One approach to determining correspondences, known as area-based matching, is to define a measure of similarity between two image locations that is based on the cross-correlation or sum of squared differences (SSD) between portions of the two images. As typically implemented, however, this requires the specification of a window size, and the window size is often given as a parameter that must be adjusted empirically.

A problem with fixed window sizes is that matching performance degrades when the image content is not well suited to the given size. For example, a window size that is too large may lead to incorrect matches because of perspective differences between the two images, and because occlusion among three-dimensional (3D) objects can

dramatically affect their appearance in the images. On the other hand, a window size that is too small will often lead to many local extrema in similarity computations because sensor noise and visual texture begin to dominate.

This section will describe a new method for selecting window size based on the moment invariants of the image region. This window-size selection approach will be evaluated specifically for its ability to facilitate visual tracking, for both monocular and binocular image sequences.

In an effort to address the window size selection problems, a few researchers have considered adaptive window-size selection within the context of image matching. Levine, et al. [Levi73], for example, changed the window size based on the intensity pattern. Kanade and Okutomi [Okut92, Kana94] present a stereo matching algorithm using an adaptive window. This algorithm selects the size and shape of the matching window adaptively for each pixel on the basis of a local evaluation of the variation in both the intensity and disparity. The variance of pixel values is used as the intensity criterion, and a measure of uncertainty is used as the disparity criterion. An initial disparity estimate is computed over the image using a rectangular window with constant size; this is then iteratively refined. Lotti and Giraudon [Lott94] extend this approach, constraining window sizes additionally by using intensity edge information. They test their approach using aerial stereo images. Scherer et al. [Sche98] presented an integrating approach of adaptive window matching with local description matching. Starting with the basic statistical framework, necessary for adaptive window matching (Kanade and Okatumi). They propose two simplifications. First, the disparity variation is set to 1 and only the intensity fluctuation is computed. The second simplification concerns the disparity measurement itself. If none or only a coarse disparity measurement is available, an internal refinement runs into the danger of divergence.

Boykov et al. [Boyk98] described a method for choosing an arbitrarily shaped connected window, in a manner that varies at each pixel. The true intensity at each pixel is estimated by selecting the intensity value that maximizes the number of pixels in the window. This approach can be applied to many problems, including image restoration and visual correspondence.

Abbott and Zheng [Abbo95] describe a stereo matching system that utilizes an ARMA (autoregression and moving average) texture model to determine window sizes. To simplify processing, vertical and horizontal image directions are considered separately. For a given direction, an exponential fit is made to the one-dimensional autocorrelation, and the window width is determined by the location at which the curve drops to 5% of its peak value.

Sung, Chien, and Kim [Sung97] propose an adaptive window-selection algorithm with four direction sizing factors. This algorithm defines eight districts: four side districts and four corner districts, and it determines four sizing directions and four sizing factors using information extracted from associated corner and side districts.

Lin [Lin96] proposed an approach based on local variance of pixel values to guide the automatic window size selection. The variance can be considered as a function of window size. The proper window size for a particular search at a given point can be determined based on the results of the variance vs. window size. Menard and Kropatsch [Mena97] have developed a method to detect the optimal scale for determining the correspondences in a given stereo pair. For each chosen region in a stereo pair, adjusting the scale can change adaptively in depending on the gray-level and disparity information, the size and shape of the search window.

Kanade and Okkatumi [Kana94] model the distribution of disparity within a window. They perform a greedy search of the space of rectangular windows, in order to minimize the uncertainty of their estimate. We will provide an empirical comparison of our results with this approach.

Scherer et al. [Sche98] presented an integrating approach of adaptive window matching with local description matching. Starting with the basic statistical framework, necessary for adaptive window matching [Kana94], they propose two simplifications. First, the disparity variation is set to 1 and only the intensity fluctuation is computed. The second simplification concerns the disparity measurement itself. If none or only a coarse disparity measurement is available, an internal refinement runs into the danger or divergence.

These methods are still restricted to rectangular windows, and impose significant computational overhead. We found that the search technique used for correspondence is

very sensitive to the choice of window size, and we have developed a new approach to select that size. Intuitively, a window that is too small will tend be distracted by texture primitives, or (at the other extreme) it may enclose a featureless region which is unsuited for area-based comparisons. For a window that is too large, the similarity measurements may give incorrect matches due to repeating patterns, occlusion, and perspective differences.

6.3.2 Window Size Selection using Moment Invariants

As seen in the previous sections, moments are one of the most useful features that can be extracted from an image. They can be invariant to translation, changes in size, intensity and rotation of the object, which forms the image.

We consider an extension of the Hu moment invariants, as introduced by Maitra [Mait79]. In addition to the invariance properties noted in the previous section, these are additionally invariant to changes in scale of image intensity, and are defined as follows:

$$\beta_1 = \frac{\sqrt{\phi_2}}{\phi_1} \qquad \beta_2 = \frac{\phi_3 \mu_{00}}{\phi_1 \phi_2}$$

$$\beta_3 = \frac{\phi_4}{\phi_3} \qquad \beta_4 = \frac{\sqrt{\phi_5}}{\phi_4} \tag{6.11}$$

$$\beta_5 = \frac{\phi_6}{\phi_1 \phi_4} \qquad \beta_6 = \frac{\phi_7}{\phi_6}$$

Many problems in early vision require the estimation of some local property of an image from noisy data. These include intensity, disparity and texture. They often vary smoothly at most points, but change dramatically at the edges of objects. In order to withstand noise, statistics must be collected over the pixels in a local window. The shape of this window is of great importance. Fixed window approaches yield good results when all the pixels in the window come from the same population as the reference pixel. However, difficulties arise when window overlaps a discontinuity. Due to discontinuity, the data comes from a bi-modal population.

We have evaluated one of the Maitra moment invariants, β_1, for its utility in window size selection. First, it is instructive to consider changes in β_1, as a function of

window size, for several cases. The first of these is a simple image involving only two intensity values, f_1 and f_2, as shown in Figure 6.1. A point of reference has been manually chosen, as indicated by the symbol "+", at a distance of *d* pixels from the intensity boundary. Using square windows of size $w \times w$ that are centered at the target, β_1 can be shown to have the form

$$\beta_1(w) = \begin{cases} 0 & w \le 2d \\ \dfrac{-16d^4(f_1-f_2)^2 - 32d^3(f_1^2-f_2^2)w - 8d^2(f_1-f_2)^2w^2 + 8d(f_1^2-f_2^2)w^3 + 3(f_1-f_2)^2w^4}{16d^4(f_1-f_2)^2 + 32d^3(f_1^2-f_2^2)w + 40d^2(f_1-f_2)^2w^2 + 24d(f_1^2-f_2^2)w^3 + (5f_1^2+22f_1f_2+5f_2^2)w^4} & w > 2d \end{cases} \tag{6.12}$$

As the window size increases without bound, the asymptotic value for $\beta_1(w)$ is

$$\lim_{s\to\infty} \beta_1(w) = \frac{3(f_1 - f_2)^2}{(5f_1^2 + 22f_1f_2 + 5f_2^2)} \tag{6.13}$$

As illustrated in Figure 6.2, β_1 for this case is unimodal, having a single maximum at a location that depends on *d*. For a given pair of intensity values f_1 and f_2, we note that the peak location of β_1 occurs at $w = \alpha d$, where α depends on f_1 and f_2.

For the case that the difference between the two intensity values f_1 and f_2 is large, for example $f_1 >> f_2$, it is helpful to consider the case

$$\beta_1(w)\Big|_{f_2=0} = \frac{-4d^2 - 4dw + 3w^2}{4d^2 + 4dw + 5w^2}. \tag{6.14}$$

Accordingly, when $f_2 >> f_1$, it is helpful to consider β_1 for $f_1 = 0$:

$$\beta_1(w)\Big|_{f_1=0} = \frac{-4d^2 + 4dw + 3w^2}{4d^2 - 4dw + 5w^2}. \tag{6.15}$$

In both cases, as illustrated in Figure 6.3, the asymptotic value of $\beta_1(s)$ for large window sizes is

$$\lim_{s\to\infty} \beta_1(w)\Big|_{f_1=0} = \lim_{s\to\infty} \beta_1(w)\Big|_{f_2=0} = \frac{3}{5}. \tag{6.16}$$

It can be seen from (6.14) and (6.15) that $\beta_1(w)\Big|_{f_1=0} = \beta_1(-w)\Big|_{f_2=0}$.

The monotonic nature of β_1 suggests its utility in the selection of size for a processing window. For example, a target has been manually chosen in Figure 6.1, as indicated by the symbol "+", at a distance of d = 20 pixels from the intensity boundary. Using square windows that are centered at the target, of size $w \times w$, β_1 was computed as a function of w. The resulting graph is shown in Figure 6.2. As expected, β_1 has value 0 for $w \le 40$, because the window contains uniform intensity values for these cases. (The dark box in the image represents the window corresponding to w = 40.) After the window has grown large enough to include some of the dark pixels, β_1 begins to increase; β_1 increases until $w \approx 80$, when approximately 77% of the pixels in the window have the same intensity as the target. After this maximum, β_1 decreases monotonically.

A second case appears in Figure 6.4. The image now consists of a single band of intensity f_1 and height $2d$ on a background of intensity f_2. Again using square windows of size $w \times w$ that are centered at the target, β_1 can be shown to have the form

$$\beta_1(w) = \begin{cases} 0 & w \le 2d \\ \dfrac{d(f_1 - f_2)(-4d^2 + w^2)}{4d^3(f_1 - f_2) + d(f_1 - f_2)w^2 + f_2 w^3} & w > 2d \end{cases} \tag{6.17}$$

Plots of this function are given for several cases of d in Figure 6.5. For large window sizes, the asymptotic value for $\beta_1(w)$ is $\lim_{s\to\infty} \beta_1(w) = 0$.

For the case that the difference between the two intensity values f_1 and f_2 is large, for example $f_1 >> f_2$, then it is helpful to consider the case

$$\beta_1(w)\Big|_{f_2=0} = 1 - \frac{-8d^2}{4d^2 + w^2}. \tag{6.18}$$

Accordingly, when $f_2 >> f_1$, then it is helpful to consider β_1 for $f_1 = 0$:

$$\beta_1(w)\Big|_{f_1=0} = -\frac{d(2d + w)}{2d^2 + dw + w^2}. \tag{6.19}$$

As illustrated in Figure 6.6 the asymptotic value of $\beta_1(w)$ for the former case is $\lim_{s\to\infty} \beta_1(w) = 1$, whereas the asymptotic value for the latter case is $\lim_{s\to\infty} \beta_1(w) = 0$.

The third case appears in Figure 6.7. The image now consists of alternating dark and light bands, and a target has been selected in the middle of a light band. Again, as

shown in Figure 6.8, $\beta_1 = 0$ as long as the intensity in the window is uniform; then β_1 fluctuates with local extrema at values of s that depend on the width of the intensity bands in the image. The maximum of β_1 occurs at $w \approx 40$, which again corresponds to the case that approximately 77% of the pixels in the window have the same intensity as the target. The dark box in this case corresponds to the case $w = 20$, with the window enclosing only light pixels.

This discussion suggests some criteria that we have developed [Sala98b] for selecting window size. In this paper, we have suggested the following criteria that we have developed for selecting window size: (1) If $\beta_1 = 0$ initially, then this represents an image region of uniform intensity, and the window size is chosen to correspond with the largest value of w for which β_1 is still 0. (2) If the first condition is not satisfied, then a local minimum in β_1 is sought, ignoring the extreme values of w under consideration; if one or more local minima are present, then the window size is chosen based on the value of w for the first local minimum. (3) Finally, if the first two conditions are not met, then the window size is chosen to correspond with the value of w for, which β_1 is maximum, over the range of w being considered.

The first criterion addresses image regions of uniform intensity, and it seeks a window size having maximum extent over such regions. The second criterion is intended to address textured image regions, and it favors smaller windows so as to avoid problems associated with occlusion and 3D perspective. The third criterion handles all remaining cases.

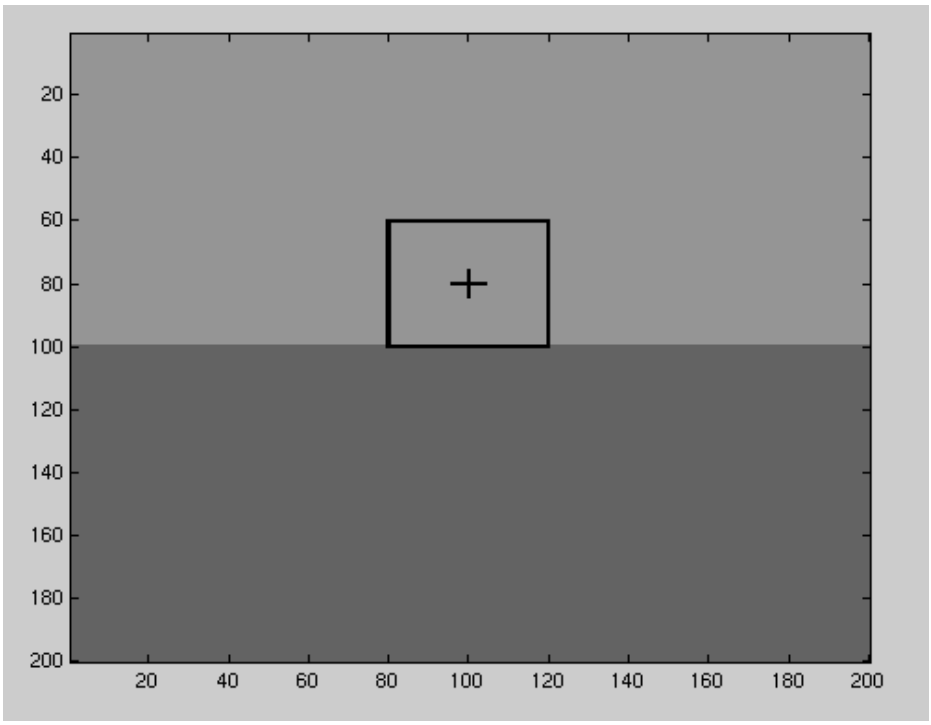


Figure 6.1. Example of window selection with a single discontinuity in image intensity. The symbol "+" represents the target point of reference. The dark box in the image indicates the largest value of w for which $\beta_1(w) = 0$.

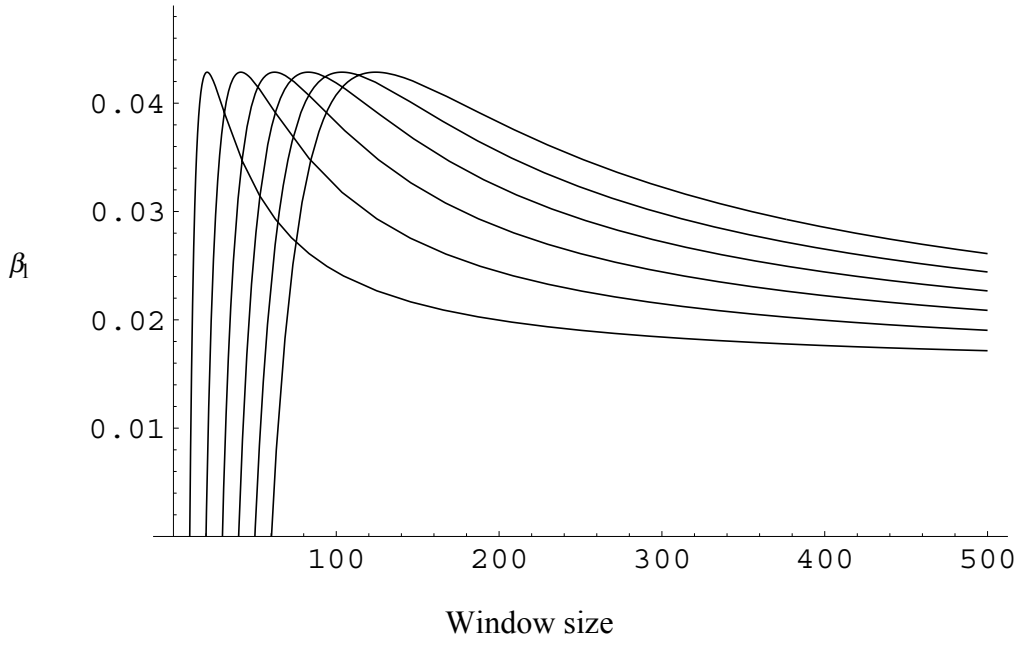


Figure 6.2. Example plots of β_1 for the image in Figure 6.1. For given values of image intensity, the location of the peak value is proportional to the distance of the target point from the intensity discontinuity. The plot at the far left corresponds to $d = 10$, and the plot at the far right corresponds to $d = 60$.

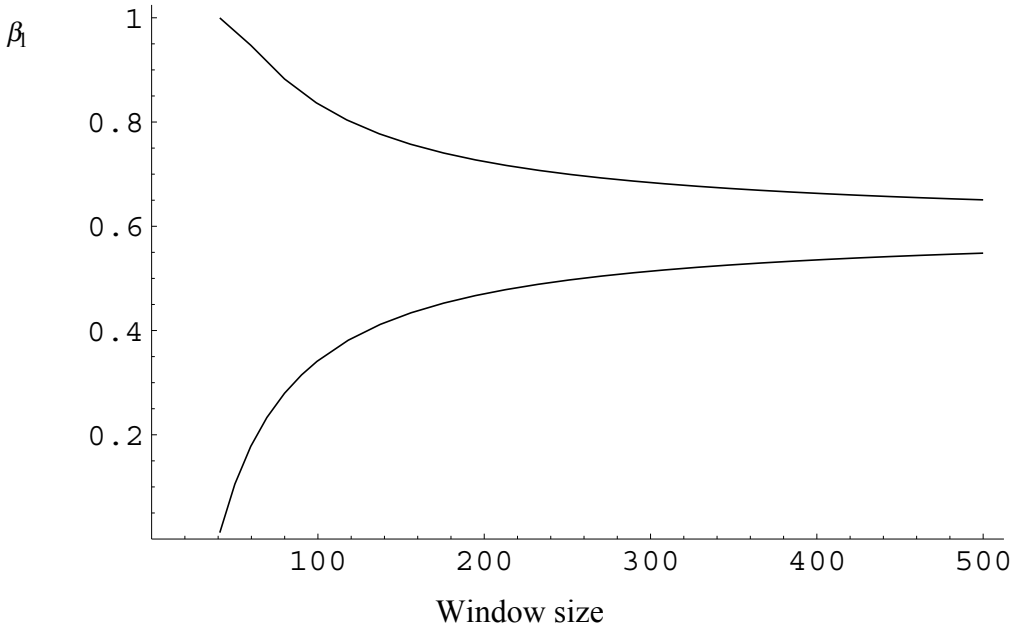


Figure 6.3. Example plots of β_1 for the image in Figure 6.1, for the cases $f_2 = 0$ (upper plot) and $f_1 = 0$ (lower plot). In both cases, the asymptotic value for large window sizes is 3/5.

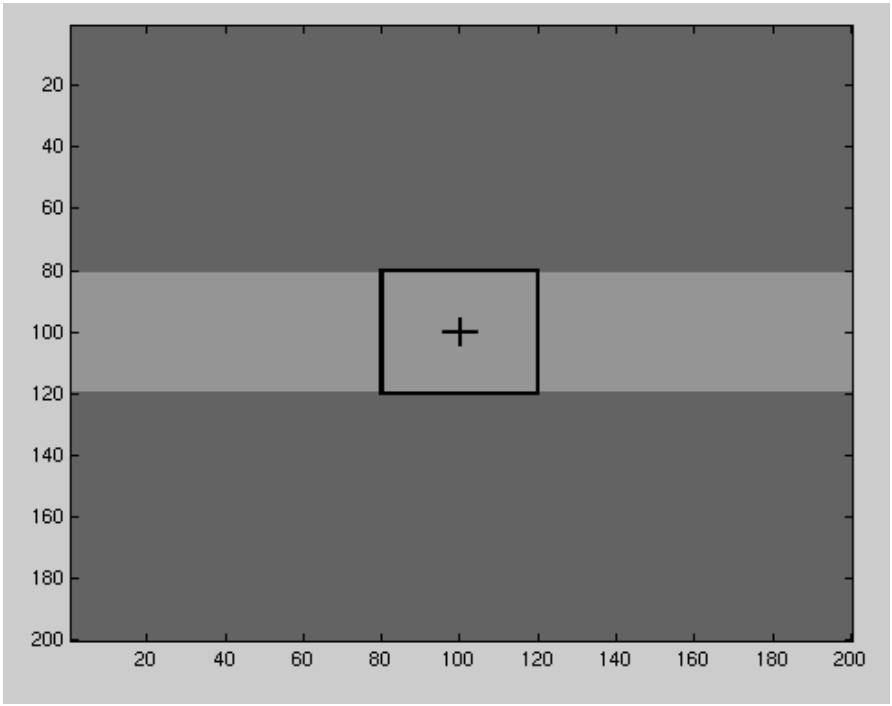


Figure 6.4. Example of window selection with 2 discontinuities in image intensity. The symbol "+" represents the target point of reference. The dark box in the image indicates the selected window of size $2d \times 2d$.

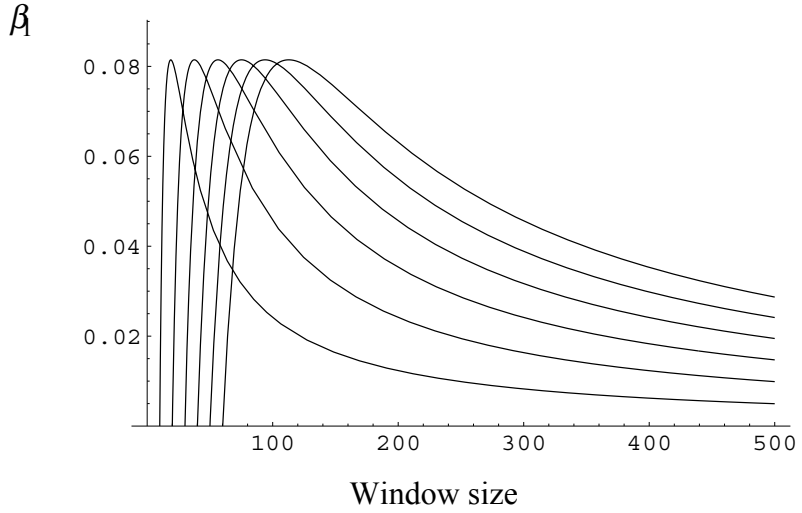


Figure 6.5. Example plots of β_1 for the image in Figure 6.4. For given values of image intensity, the location of the peak value is again proportional to the distance of the target point from the intensity discontinuity. In this case, however, the asymptotic value of β_1 is 0. The plot at the far left corresponds to d = 10, and the plot at the far right corresponds to d = 60.

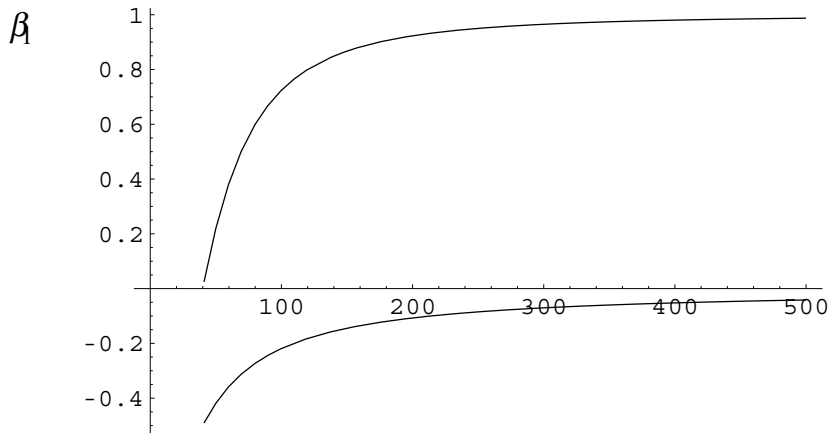


Figure 6.6. Example plots of β_1 for the image in Figure 6.4, for the cases $f_2 = 0$ (upper plot) and $f_1 = 0$ (lower plot). Unlike the previous cases, the asymptotic values differ for large window sizes.

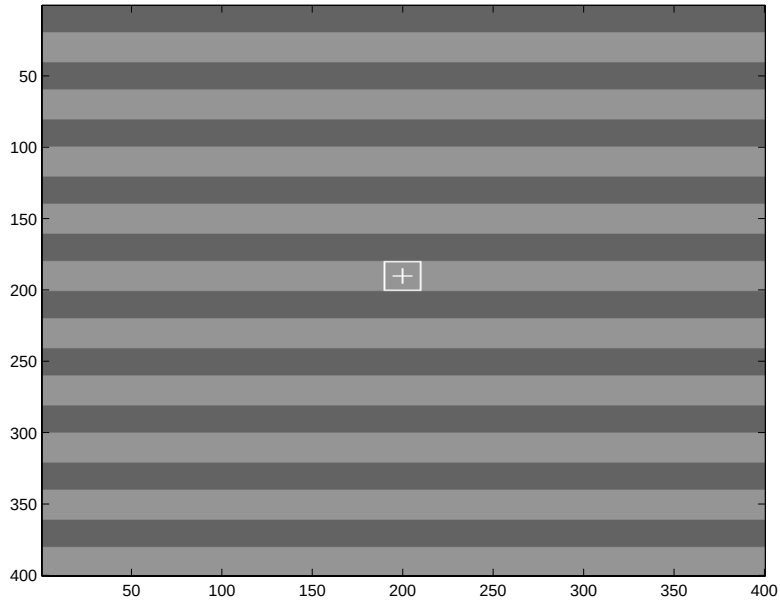


Figure 6.7. Example of window selection for an image with alternating intensity bands. The symbol "+" represents the target point of reference. The white box in the image indicates the selected window of size $2d \times 2d$.

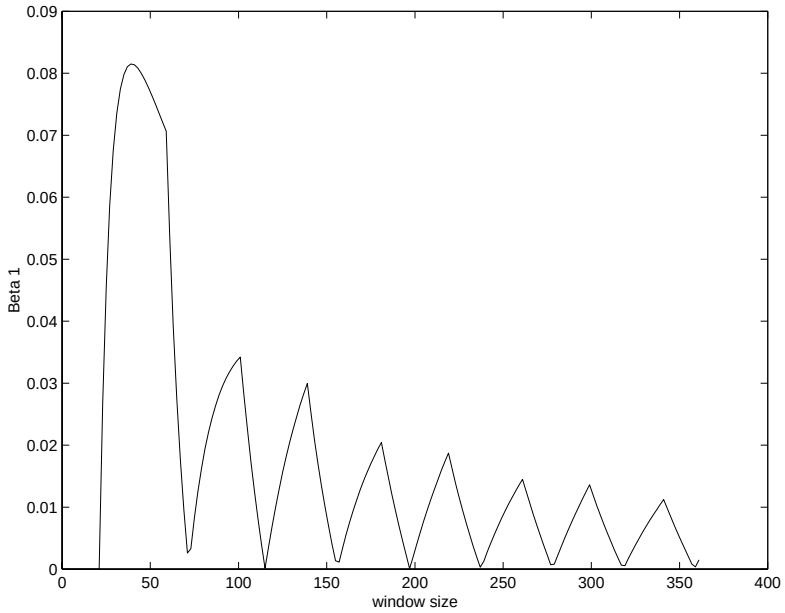


Figure 6.8. Example plot of β_1 for the image in Figure 6.7. The fluctuations of β_1 are nearly regular, resulting from the periodicity of the image.

The above approach is heuristic in nature and was entirely based on experimental observation. The first Maitra moment β_1 is truly invariant under translation, rotation, scale and scale of image intensity for functions defined in the continuos domain. Image acquisition by a digital system imposes spatial and intensity quantization that, in turn, introduce errors into moment and invariant computation. Figure 6.9 contains an example image along with plots of β_1 for a target point in a noise free image using the heuristic method (section 6.3.2.). Figure 6.9(b), for example, shows β_1 as a function of w for a target that has been selected on the white car. Using criterion 2 in the heuristic approach, a window size corresponding to $w = 15$ is chosen.

It is apparent that the first Maitra moment invariant β_1 is sensitive to the image distortion. To illustrate this, we added some artificial noise to the image (see Figure6.10) near the target point. Then, the same target in Figure 6.9 has been manually chosen in Figure 6.10(a), as indicated by the symbol "+". Using square windows that are centered at the target, of size $w \times w$, β_1 was computed as a function of w. The resulting graph is shown in Figure 6.10(b). As expected, the value of β_1 is changed suddenly when the window contains the artificial noise. Using criterion 2 in the initial approach, a window size corresponding to $w = 7$ is chosen.

It is clear that the estimated window size of the noise free image is different from the distorted image window size. This constituted a basic motive to smooth β_1 values before the selection of window size. In the next section, we suggest to use a nonparametric estimation (Huber estimator) for smoothing β_1 to make it robust against noise. Let $\underline{z}$ represent the first Maitra moment invariant β_1 over a particular range of window sizes for the target point in the image. It is assumed that $\underline{z}$ is ideally linear, and is corrupted by additive noise.

6.3.3 Regression Model

The least mean squares regression model is optimal when the noise distribution is Gaussian with zero mean. However, in cases where the noise is not Gaussian, the least mean squares estimator becomes unreliable. The breakdown point of an estimator may be defined as the smallest amount of outlier contamination that may force the value of the

estimate outside an arbitrary range. For example, the least square has a breakdown point of 0%, because a single outlier may have a substantial impact on the estimated parameters [Rous87].

The least-squares method tries to minimize $\sum_i r_i^2$, which is unstable if there are outliers present in the data. The M-estimators try to reduce the effect of outliers by replacing the squared residuals r_i^2 by another function of the residuals, yielding $\min \sum_{i=1}^{m} \rho(r_i)$, where ρ is a symmetric, positive-definite function with a unique minimum at zero, and is chosen to be less increasing than square.

Let $\psi(x) = \frac{d\rho(x)}{dx}$ where $\psi(x)$ is the influence function that measures the influence of datum on the value of the parameter estimate. For example, for the least squares with $\rho(x) = \frac{x^2}{2}$, the influence function is $\psi(x) = x$, that is, the influence of a datum on the estimate increases linearly with the size of its error, which confirms the non-robustness of the least-squares estimate. For a set of *m* observed values, the model may be expressed as follows [Hube81]:

$$\begin{aligned} z_1 &= a_0 + a_1 y_1 + e_1 \\ z_2 &= a_0 + a_1 y_2 + e_2 \\ &\cdots \\ z_m &= a_0 + a_1 y_m + e_m \end{aligned} \tag{6-20}$$

where $\{z_i\}$ represent observed first Maitra moment invariant β_1 values, $\{y_i\}$ represent a subset of the possible range values, $\{e_i\}$ represent noise, $\{a_i\}$ and represent the variable to be estimated.

Converting to matrix form yields $\underline{Z} = \underline{\underline{H}}\,\underline{x} + \underline{e}$ (6.21)

where

$$\underline{Z} = [z_1, z_2, ..., z_m]^T, \qquad \underline{x} = [x_0, x_1, ..., x_{n-1}]^T, \quad \underline{e} = [e_1, e_2, ..., e_m]^T$$

$$H = \begin{bmatrix} 1 & y_1 & y_1^2 & .. & y_1^n \\ 1 & y_2 & y_2^2 & .. & y_2^n \\ : & : & : & :: & : \\ 1 & y_m & y_m^2 & .. & y_m^n \end{bmatrix}$$

To estimate the vector $\underline{x}$, we apply the Huber M-estimator [Hube81] to minimize the objective function

$$J(\underline{x}) = \sum_{i=1}^{m} \rho\left(\frac{r_i}{s}\right) \tag{6.22}$$

where ρ is a symmetric positive-definite function with a unique minimum at zero, r_i represents the residue value, $r = [r_1, r_2, ..., r_m] = \underline{Z} - \underline{H}\,\underline{x}$, also s represents the median absolute deviation from the median (MAD), $s = 1.4826 \operatorname{med}_i \left| z_i - \operatorname{med}_j z_j \right|$.

$$\rho\left(\frac{r}{s}\right) = \begin{cases} \frac{1}{2}\left(\frac{r}{s}\right)^2 & \left|\frac{\mathrm{r}}{\mathrm{s}}\right| \le b \\ b\left|\frac{r}{s}\right| - \frac{b^2}{2} & \left|\frac{\mathrm{r}}{\mathrm{s}}\right| > b \end{cases}$$

$$\psi\left(\frac{r}{s}\right) = \begin{cases} \frac{r}{s} & \left|\frac{r}{s}\right| \le b \\ b\,\mathrm{sgn}\left(\frac{r}{s}\right) & \left|\frac{r}{s}\right| > b \end{cases} \tag{6.23}$$

$$q(r/s) = \frac{\psi(r/s)}{r/s}$$

where the value b is the standard deviation of Huber M-estimator, the ψ function is the influence function that measure the influence of datum on the value of the parameter estimate (first derivative for the ρ function), and the q is the weight function.

The solution must satisfy the following necessary condition:

$$\frac{\partial}{\partial \underline{x}} J(\underline{x}) = \sum_{i=1}^{m} \frac{\partial \rho\left(\frac{r_i}{s}\right)}{\partial \underline{x}} = 0$$

Our approach uses the iterative reweighted least squares (IRLS) method to find the estimated vector x. The intermediate solution at iteration *k+1* is given by

$$x^{(k+1)} = \left(\underline{H}^T \underline{R}^{-1} \underline{Q}^{(k)} \underline{H}\right)^{-1} \cdot \underline{H}^T \underline{R}^{-1} \underline{Q}^{(k)} \underline{Z} \tag{6.24}$$

where $\dim(\underline{Z}) = m$, $\dim(\underline{H}) = m \times n$, $\dim(\underline{x}) = n$, $m >> n$, and $Rank(\underline{H}) = n$. Also, m is the number of observations, and n is the length of the estimated vector x.

Using $Q = diag\left(q\left(\frac{r_i}{s}\right)\right)$, $q = \frac{\psi(r/s)}{r/s}$, and $R^{-1} = diag\left(\frac{1}{\sigma_i^2}\right)$. Iteration continues until the following stop condition is satisfied:

$$\left|\frac{x^{(k+1)} - x^{(k)}}{x^{(k)}}\right| \le 10^{-2}$$

We later studied the problem to understand the pattern in which β_1 changes. We find the following explanation convincing, which relates to the spatial location of the object relative to its neighbors. The first Maitra moment invariant β_1 will remain constant for as long as the window enlargement does not result in a new object appearing in the window.

A sudden increase or decrease in β_1 therefore represents evidence that a new object may be coming into view within the growing image window. Because of this, we began to consider derivatives of β_1, as a function of window size *s*, as a mechanism for selecting window size. To achieve improved performance in the presence of noise, we apply a robust smoothing operator [Hube81] to calculated values of β_1 before computing derivative approximations. Let $\hat{\beta}_1$ represent the smoothed moment invariant.

We first considered maxima in the magnitude of the first derivative, $\frac{\partial \hat{\beta}_1(s)}{\partial s}$, for window size selection. This yielded good results in many example images. However, we obtained even better results by using a method based on the second derivative.

6.3.4 Robust Adaptive Window-size Selection Algorithm

In the previous sections, we have developed a theory for computing the estimates of the first Maitra moment invariants over a particular range of window sizes. Now, we

present a complete description of our iterative algorithm to estimate the window size that corresponds to the sudden increasing or decreasing in $\hat{\beta}_1$.

Step1: select a reference point in the image, about which a window size is to be chosen.

Step2: Center a window of size $w \times w$ at the reference point and compute $\beta_1(w)$, w=3,5,7,…., until w=101or the window reaches the image boundary.

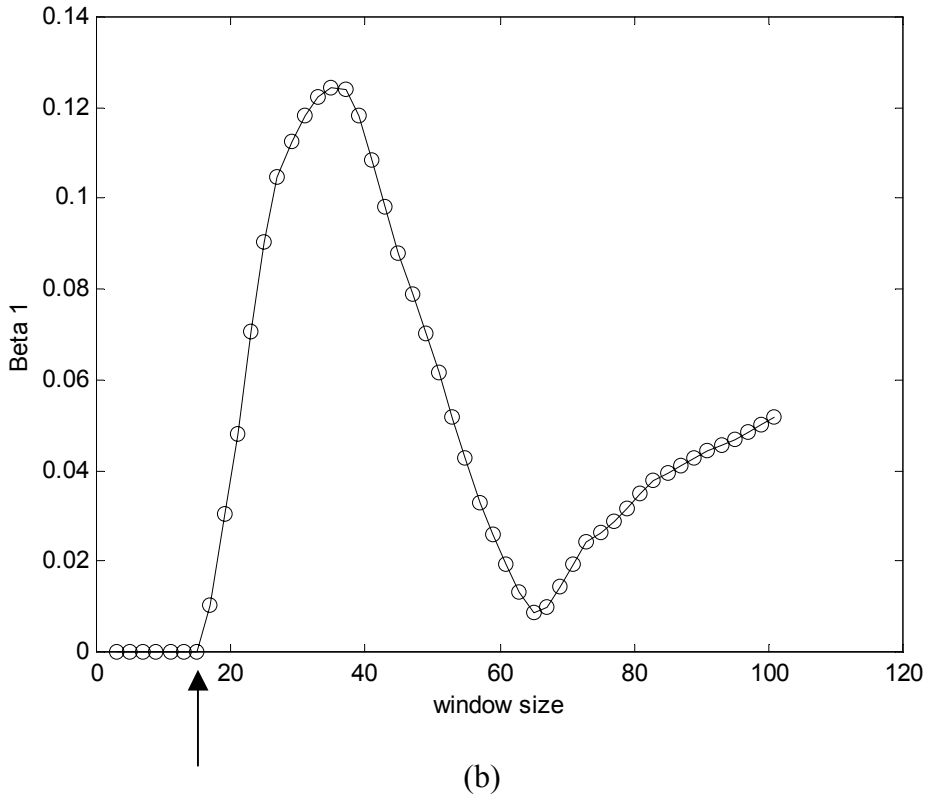
Step3: Smooth $\beta_1(w)$ using linear regression model to obtain $\hat{\beta}_1(w)$.

Step4: Compute the second derivative $\frac{\partial^2 \hat{\beta}_1(w)}{\partial w^2}$.

Step5: Determine w that corresponding to the $\arg\max_w \left| \frac{\partial^2 \hat{\beta}_1(w)}{\partial w^2} \right|$.

Figure 6.11 contains an example for the same target in Figure 6.10 has been manually chosen in Figure 6.11(a), as indicated by the symbol "+". Using square windows that are centered at the target, of size $w \times w$, β_1 was computed as a function of w. The resulting graph is shown in Figure 6.11(b). After applying our approach, a window size corresponding to w = 15 is chosen. It is shown that new approach robust against noise compared with the initial approach.

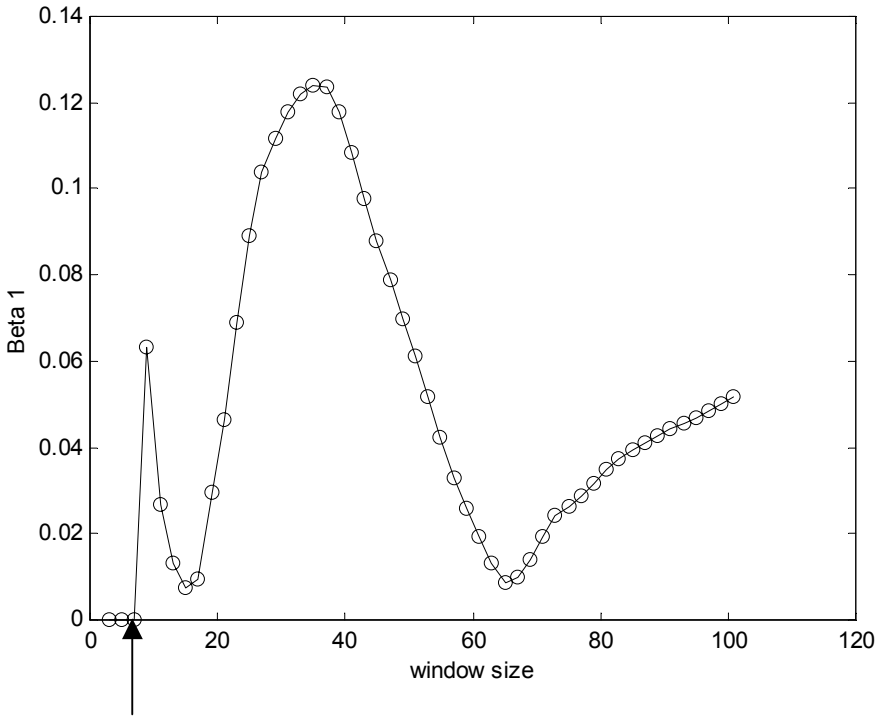
(a)



(b)

Figure 6.9. Examples of window size selection with a real image using the initial approach. (a) The "car" noise free image. (b) Plots of β_1 as a function of w for the target indicated by "+". Arrow indicates the window size that was selected.

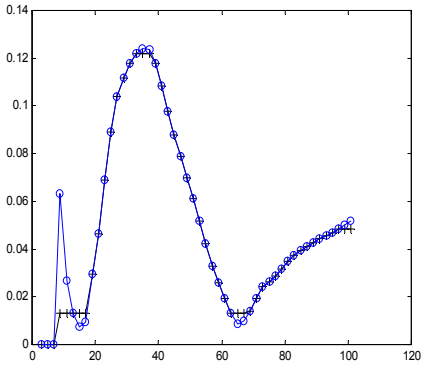
(a)

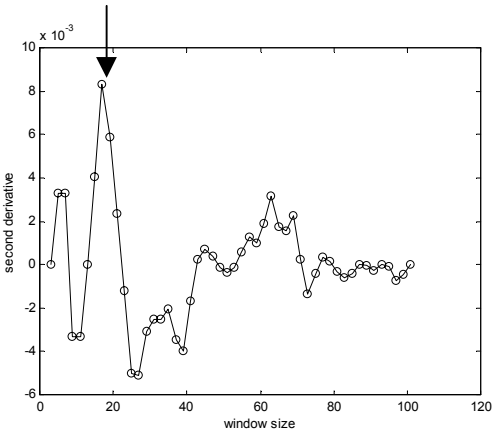


(b)

Figure 6.10. Examples of window size selection with a real image using the initial approach. (a) The "car" image contains artificial noise. (b) Plots of β_1 as a function of *w* for the target indicated by "+". Arrow indicates the window size that was selected.

(a)





(b) (c)

Figure 6.11. Examples of window size selection with a real image using the new approach. (a) The "car" image contains artificial noise. (b) Plots of β_1 together with $\hat{\beta}_1$ as a function of w for the target indicated by "+". (c) The second derivative of $\hat{\beta}_1$. Arrow indicates the window size that was selected.

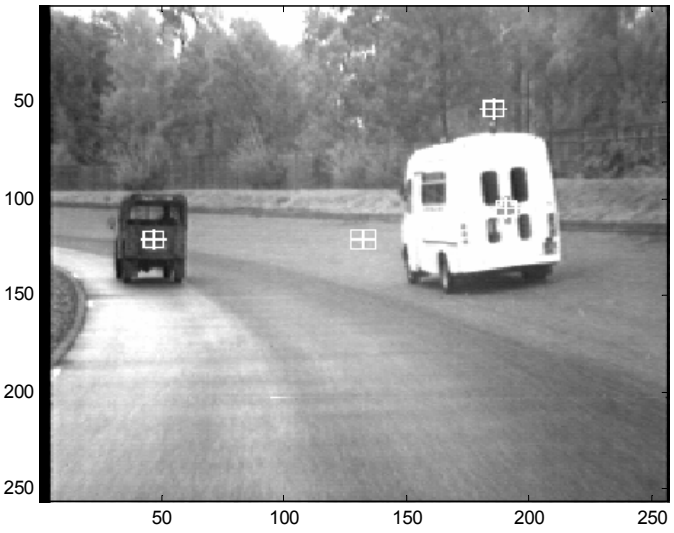
6.4 Window Size Selection Methods Experimental Results with Real Images

In the previous sections, we have discussed a new method for selecting window size adaptively. In this section, we examined results from our new method on real images. We also provided results compared to Kanade and Okotumi's adaptive window scheme [Kana94], which is a well-known method, under the two simplifications that proposed by Scherer et al. [Sche98]. The first simplification is setting the disparity variation to 1 and only the intensity fluctuation is computed. The second simplification concerns the disparity measurement itself. Also, we implemented this method in matlab. We have tested the methods for selecting window size using several real images.

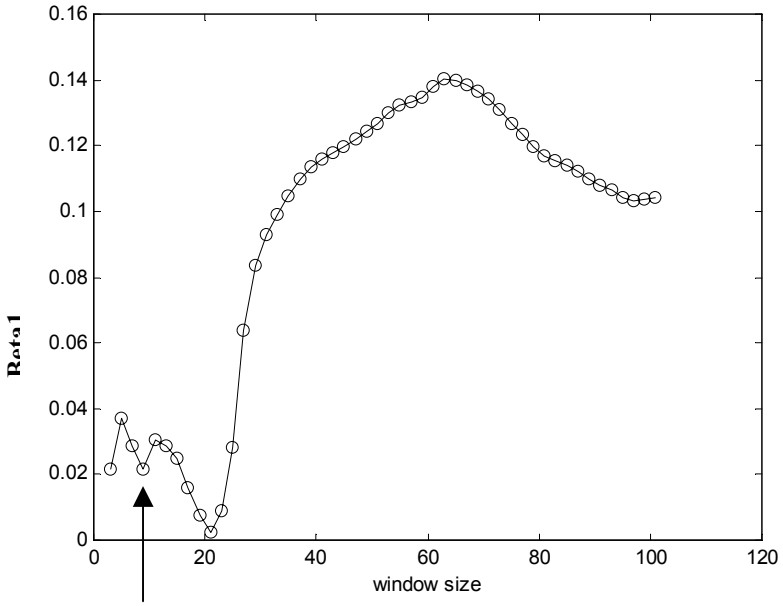
Figure 6.12 contains an example image along with plots of β_1 for several target points in the image using the initial method (section 6.3.2.). The image contains a large amount of fine textures. Figure 6.12(c), for example, shows β_1 as a function of w for a target that has been selected on the black car. Using criterion 2, a window size corresponding to $w = 9$ is chosen. Figures 6.12(c-e) correspond to targets chosen based on criterion 2.

Figure 6.13 contains an example image along with plots of $\hat{\beta}_1$ and its second derivative. In the image of Figure 6.13(a), several targets ("+") have been selected manually, and the automatically selected windows are indicated. For example, the left part of Figure 6.13(b) contains plots of β_1 (before smoothing) and $\hat{\beta}_1$ (after smoothing) for a target, which lies on a background. The selected window contains pixels from the background area only. The right part of Figure 6.13(b) contains a plot of the second derivative. The maximum magnitude occurs at $w = 23$, and this is the window size that is chosen. Figures 6.13(c-e) contain similar plots for the other targets. In all cases, the selected windows reach texture boundaries, but do not extend across those boundaries to a significant degree.

Figure 6.14 illustrates the windows that are chosen for the same targets of Figure 6.13 using the approach of [Kana94], with simplifications proposed by [Sche98]. The resulting criterion depends on a measure of the energy of intensity gradient, and a single threshold value must be provided empirically. We selected this threshold to yield windows of approximately the same size for this image.

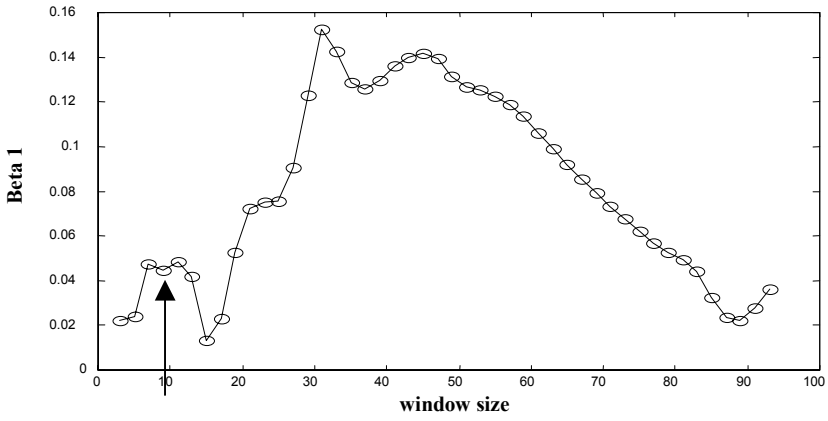


(a)

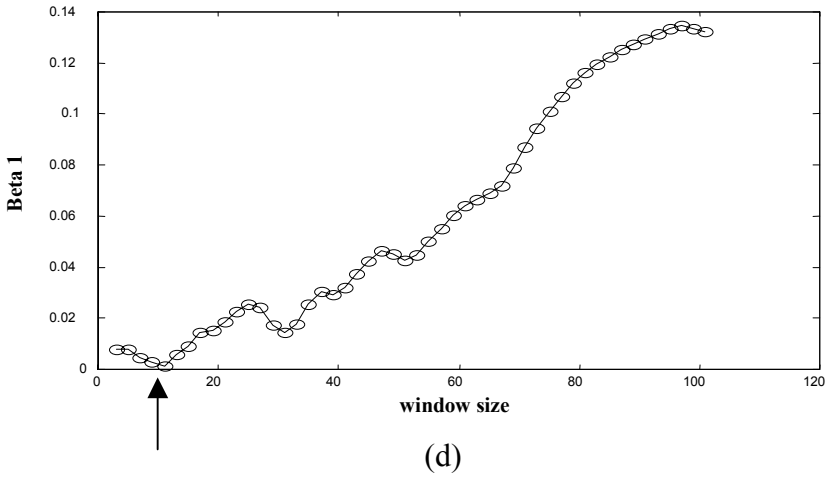


(b)

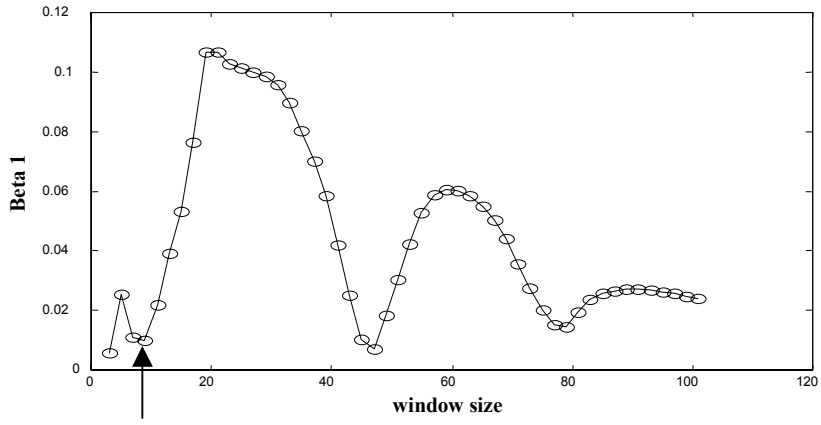
Figure 6.12. Examples of window size selection with real image using initial approach. (a) The "car" image. (b)-(e) Plots of β_1 as a function of w for the four targets shown ordered (left to right and top to bottom).



(c)



(d)



(e)

Figure 6.12. Continued, (c)-(e) In each case, the final value of *w* was chosen to coincide with the first local minimum using criteria 2. Arrows indicate the window size that was selected.

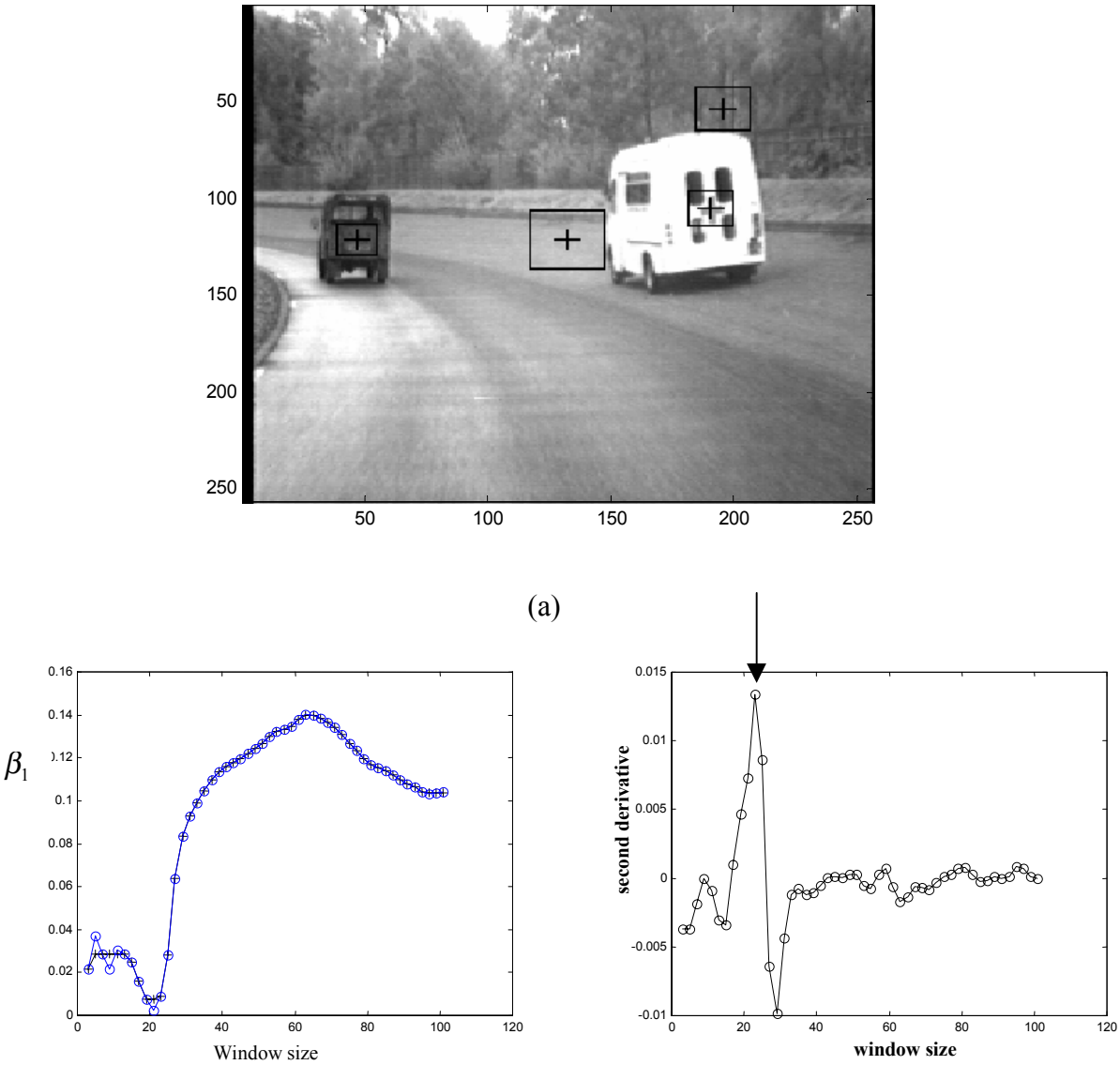
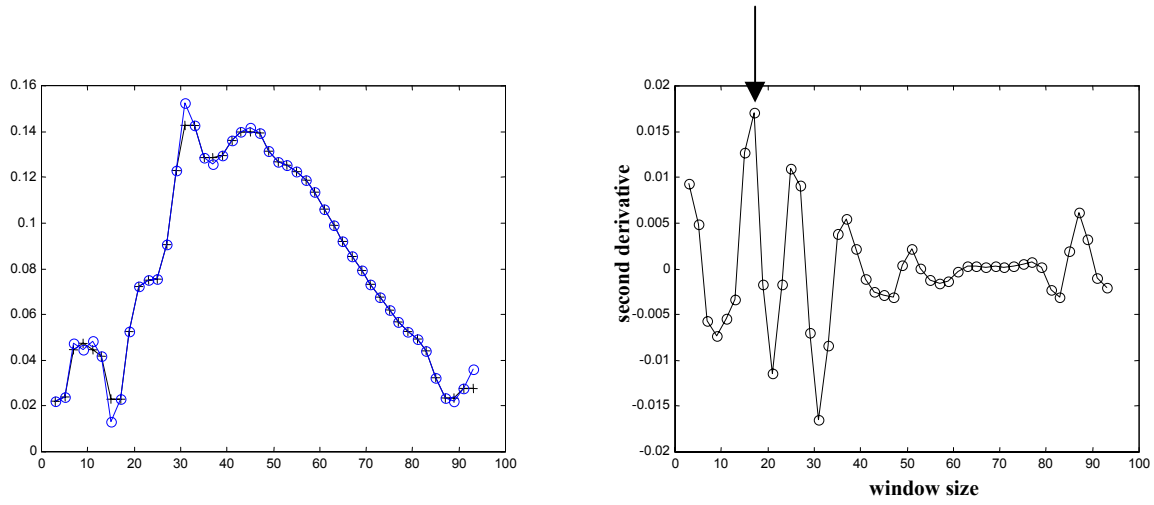
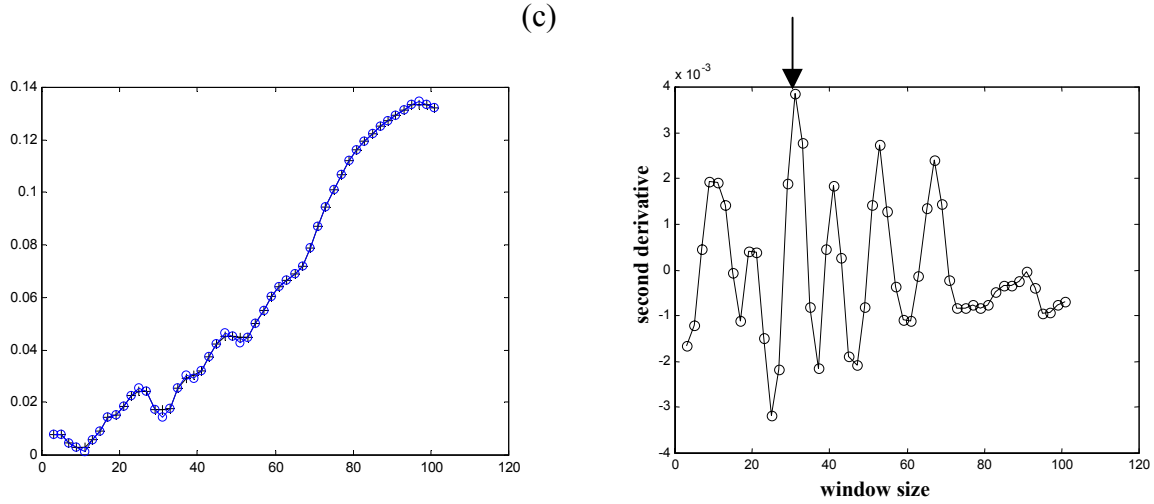


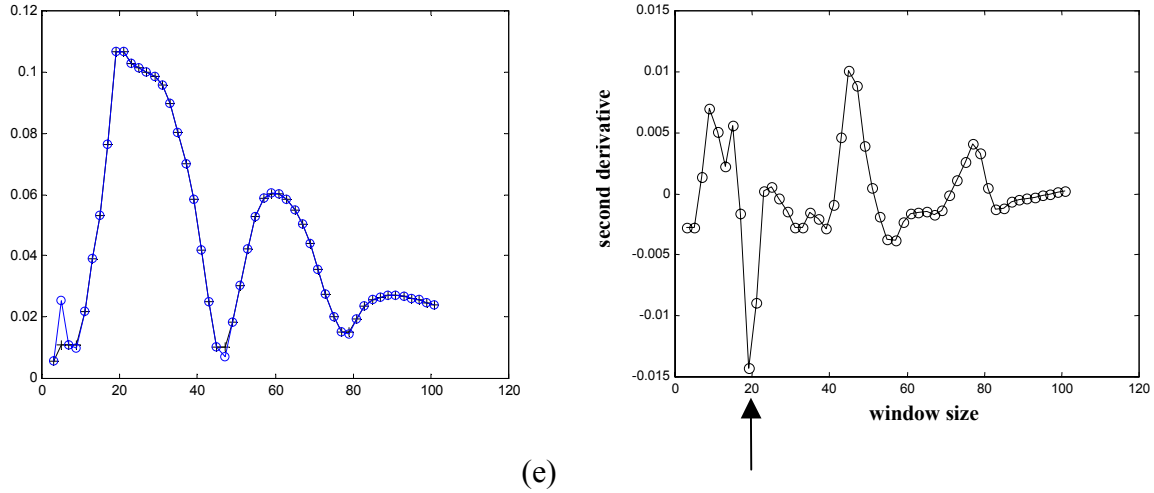
Figure 6.13. Examples of window size selection with a real image using the new approach. The same targets in Figure 6.12 were used. (a) The "car" image. (b)-(e) Left figure plots the $\hat{\beta}_1$ (+) together with the β_1 values (o) as a function of *w* for the four targets and right figure plots the second derivative of $\hat{\beta}_1$ as a function of *w* for the four targets. Arrows indicate the window size that was selected.



(c)

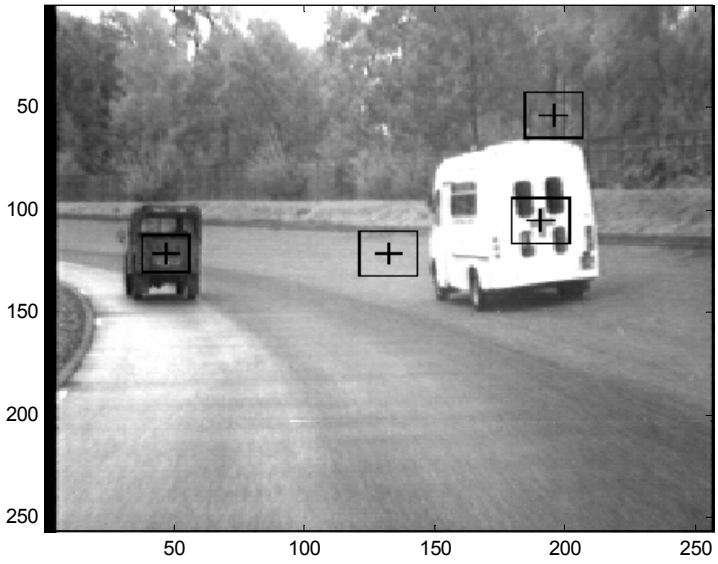


(d)

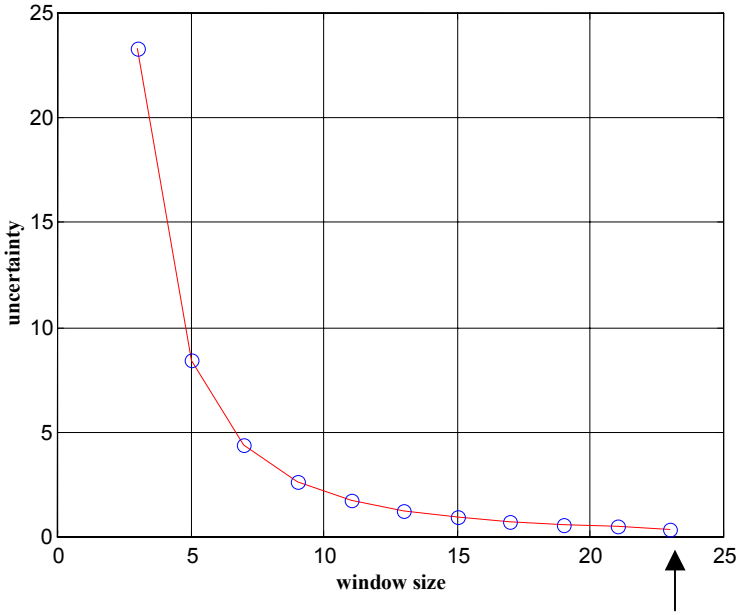


(e)

Figure 6.13. Continued, (c)-(e) A considerable degree of texture is present in the image.

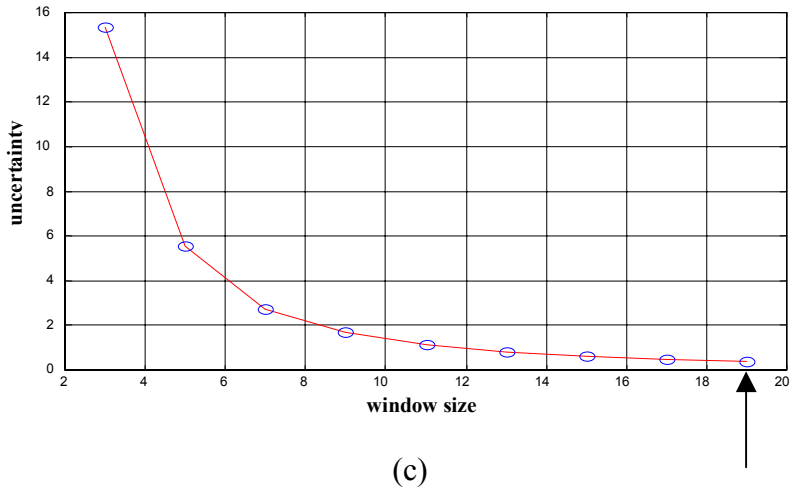


(a)

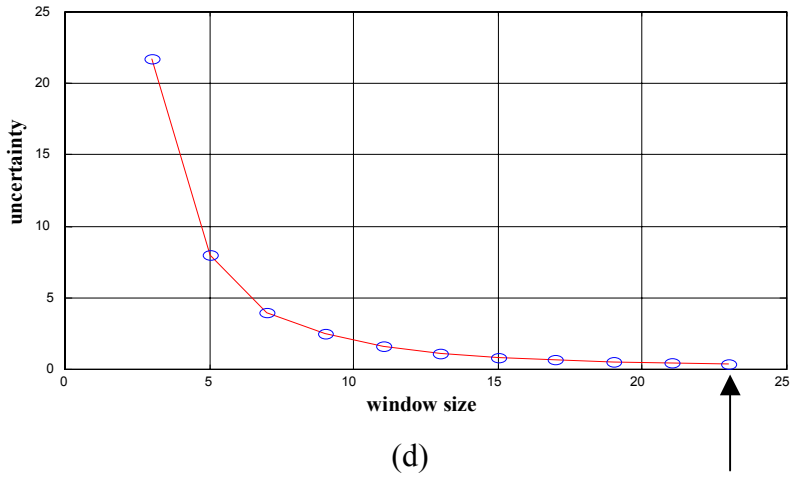


(b)

Figure 6.14. Examples of window size selection with real image using Kanade approach. The same targets in Figure 6.13 were used. (a) The "car" image. (b)-(e) Plots of uncertainty as a function of *w* for the four targets shown ordered (left to right and top to bottom).



(c)



(d)

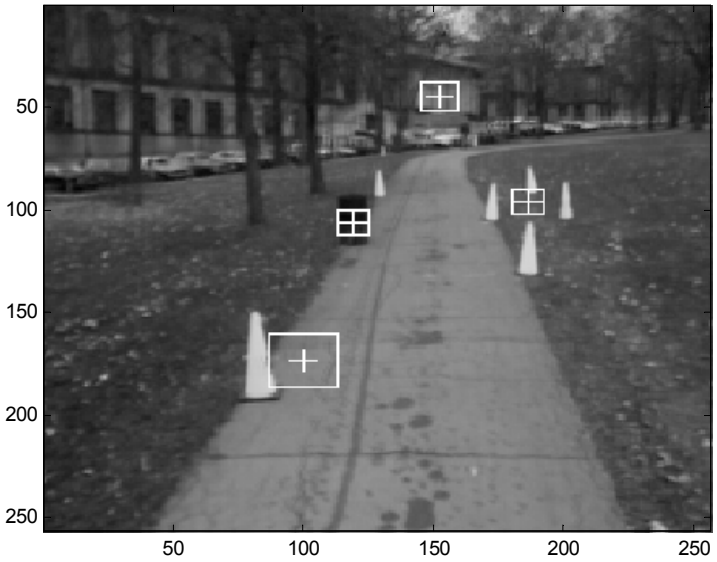
(e)

Figure 6.14. Continued, (c)-(e). Arrows indicate the window size that was selected.

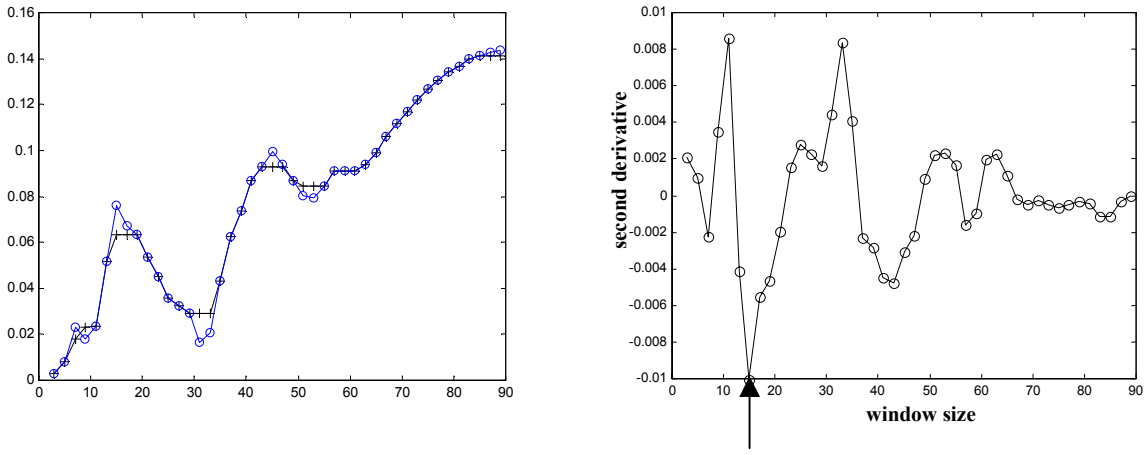
Figure 6.15 contains an example image along with plots of $\hat{\beta}_1$ and its second derivative. In the image of Figure 6.15(a), several targets ("+") have been selected manually, and the automatically selected windows are indicated. The left part of Figure 6.15(c) contains plots of β_1 (before smoothing) and $\hat{\beta}_1$ (after smoothing) for a target, which lies on a grassy area between several cones. The selected window contains pixels from the grassy area only. The right part of Figure 6.15(c) contains a plot of the second derivative. The maximum magnitude occurs at $w = 13$, and this is the window size that is chosen. Figures 6.13(c-e) contain similar plots for the other targets. Again in all cases, the selected windows reach texture boundaries, but do not extend across those boundaries to a significant degree.

To compare the results of the new approach with a method that is based on the variance of image intensities that was proposed by [Lin96]. Also, we implemented this method in matlab. Figure 6.16 contains the same targets that were used in Figure 6.15 along with plots of image variance σ^2 for the same target points. Figure 6.16(b), for example, shows a plot of σ^2 for the desired target over a particular range of window sizes. Window sizes are chosen to correspond to a local minimum of variance, as a function of window size. The resulting window sizes are too large, in this example, and in general the approach [Lin96] exhibits difficulties for the cases of no local minimum, or of multiple local minima.

Figure 6.17 illustrates the windows that are chosen for the same targets of Figure 6.15 using the approach of [Kana94], with simplifications proposed by [Sche98]. The approach of [Kana94] was developed for stereo matching, and binocular disparity is used explicitly in window selection. The simplification of [Sche98] sets the disparity uniformly to 1, effectively ignoring disparity and enabling the use of the algorithm with a single image. The resulting criterion depends on a measure of the energy of intensity gradient, and a single threshold value must be provided empirically. Again, we selected this threshold to yield windows of approximately the same size for this image.

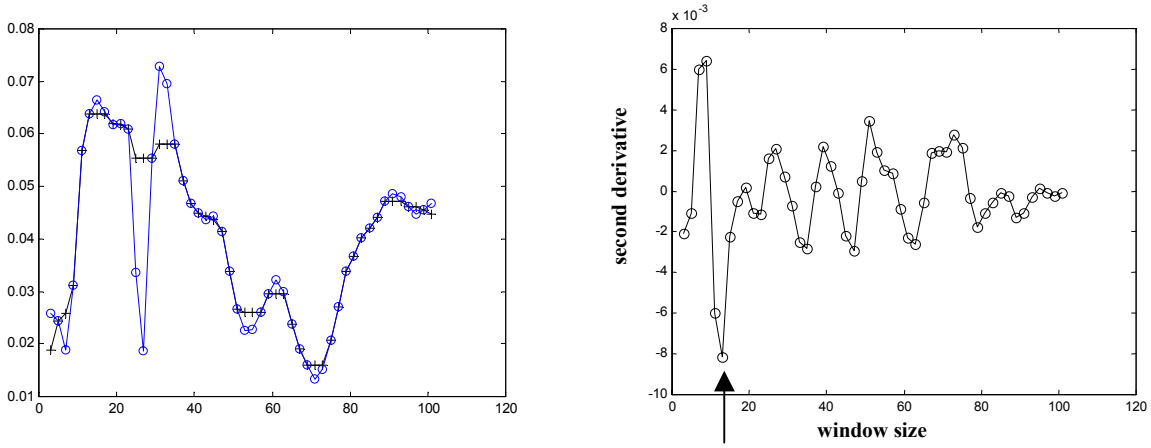


(a)

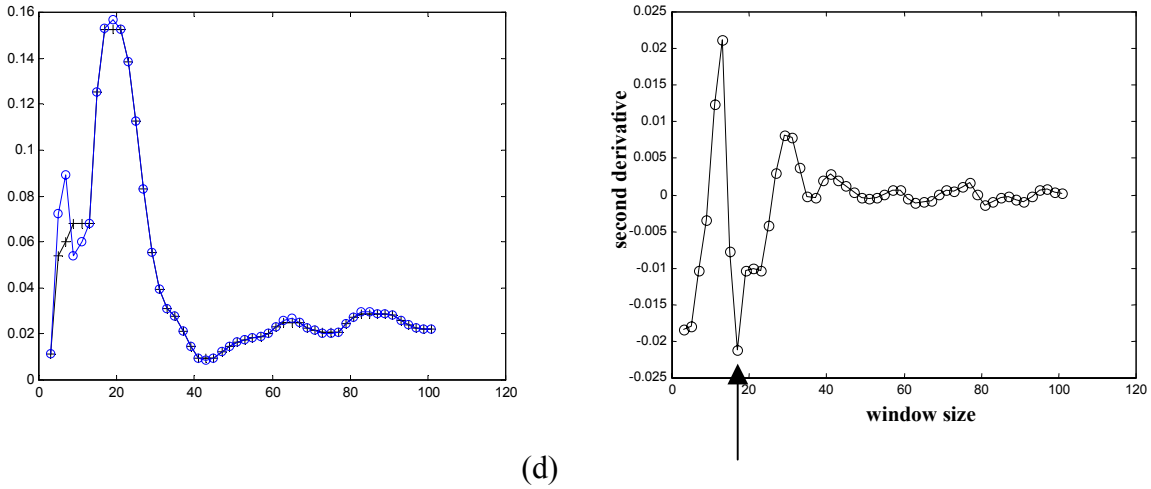


(b)

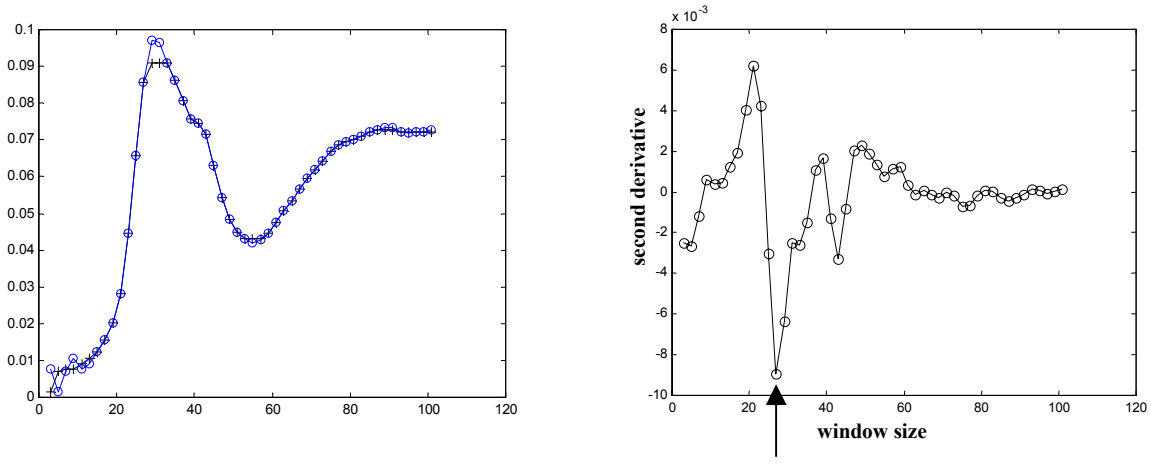
Figure 6.15. Examples of window size selection with a real image using the new approach. The same targets in Figure 6.14 were used. (a) The "cone" image. (b)-(e) Left figure plots the $\hat{\beta}_1$ (+) together with the β_1 values (o) as a function of w for the four targets and right figure plots the second derivative of $\hat{\beta}_1$ as a function of w for the four targets. Arrows indicate the window size that was selected.



(c)

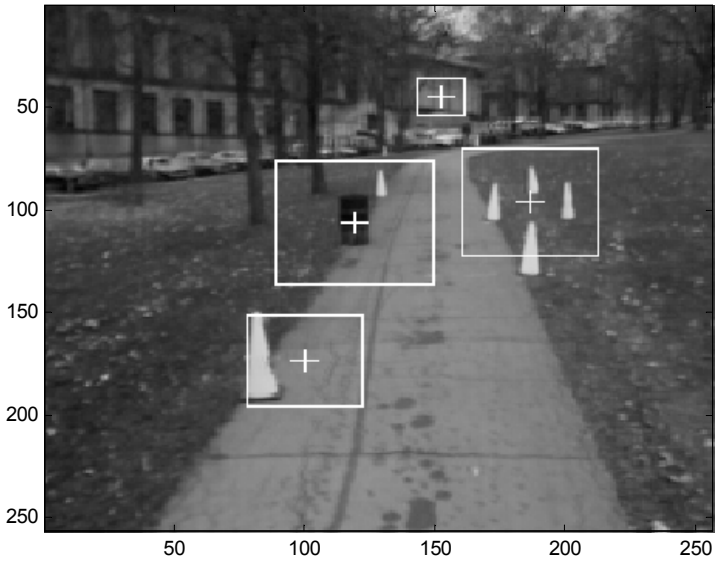


(d)

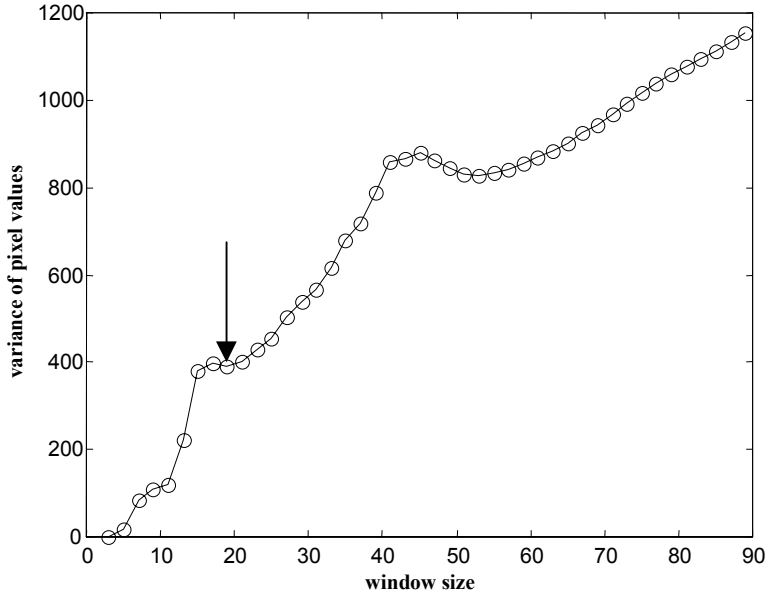


(e)

Figure 6.15. Continued, (c)-(e) A considerable degree of texture is present in the image.

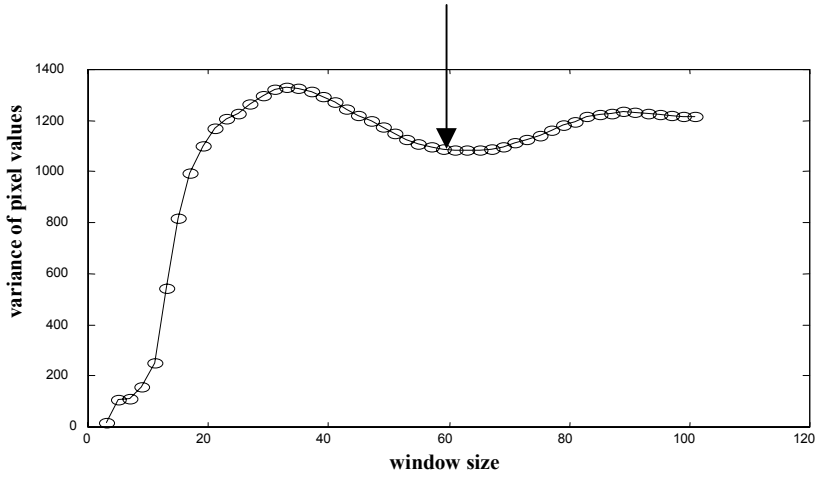


(a)

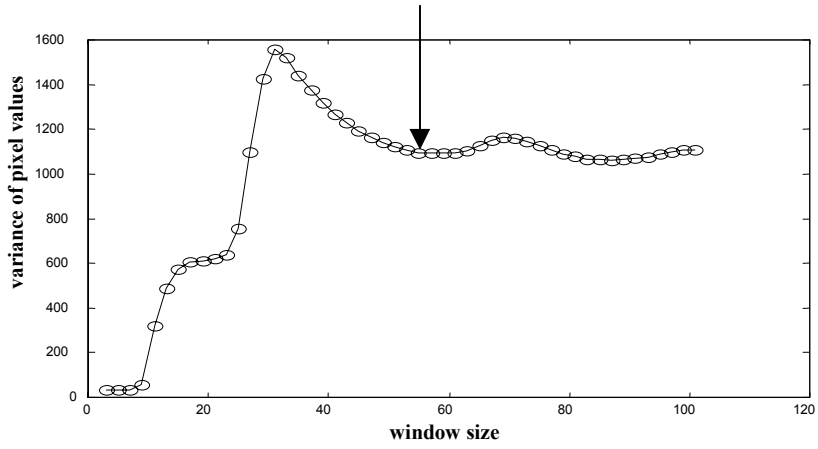


(b)

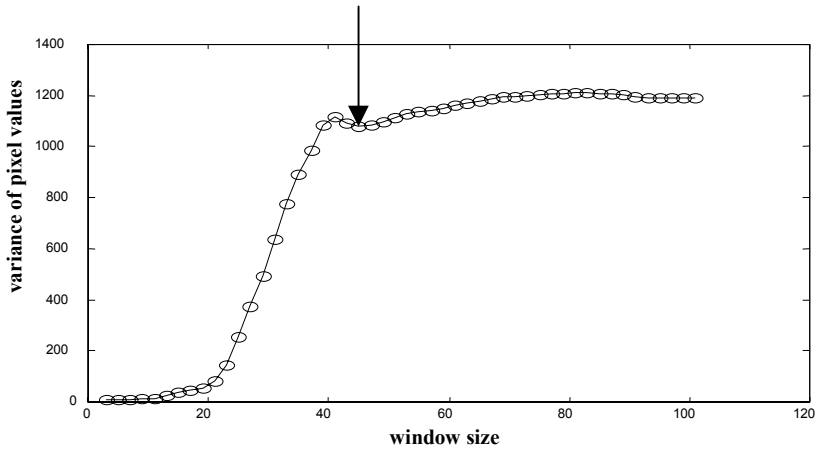
Figure 6.16. Examples of window size selection with real image using image variance approach. The same targets in Figure 6.15 were used. (a) The "cone" image. (a) The "cone" image. (b)-(e) Plots of σ^2 as a function of w for the four targets shown ordered (left to right and top to bottom).



(c)

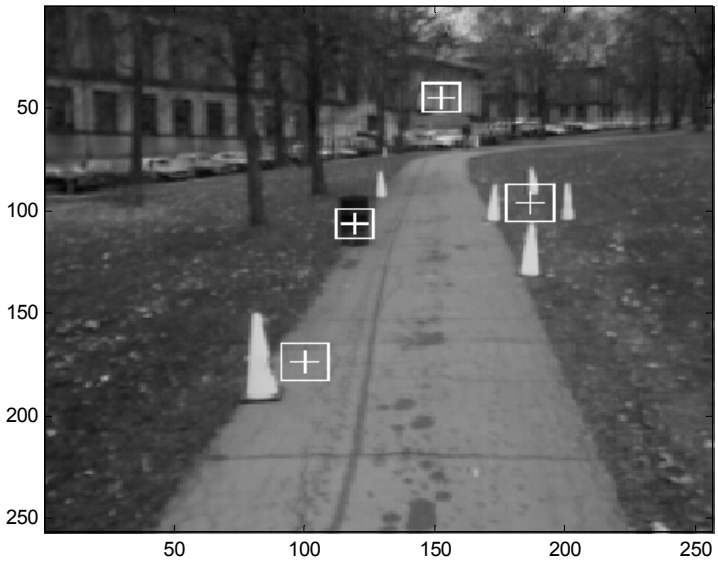


(d)

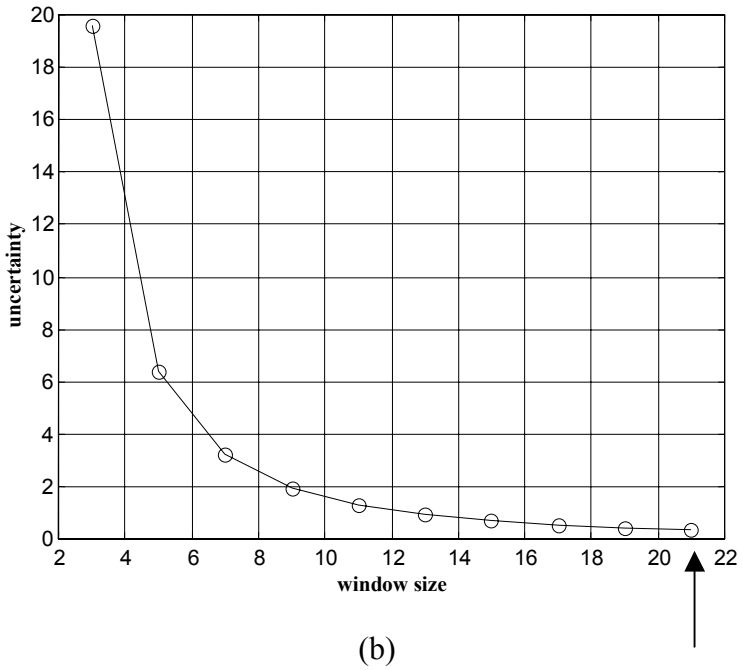


(e)

Figure 6.16. Continued, (c)-(e) In each case, the final value of w was chosen to coincide with the first local minimum in σ^2. Arrows indicate the window size that was selected.

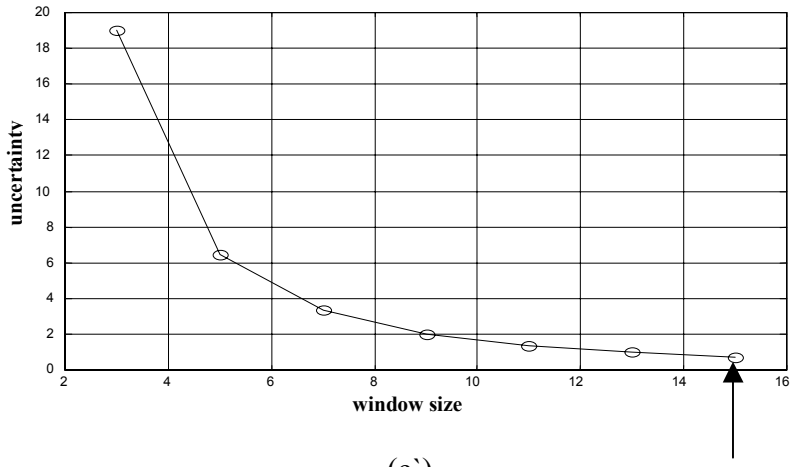


(a)

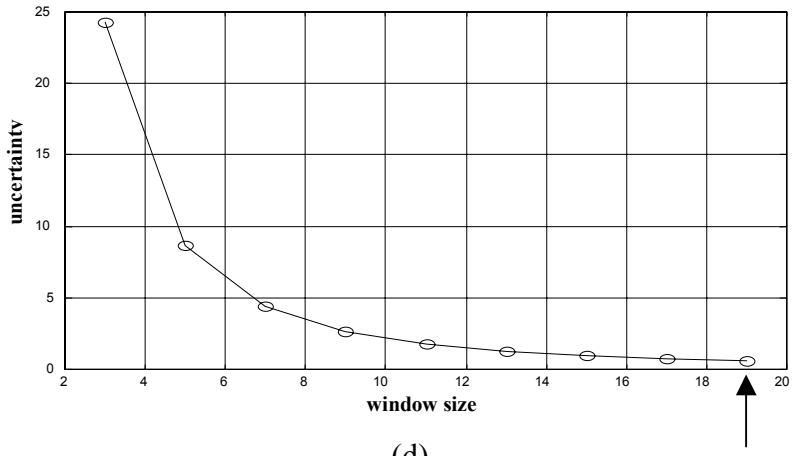


(b)

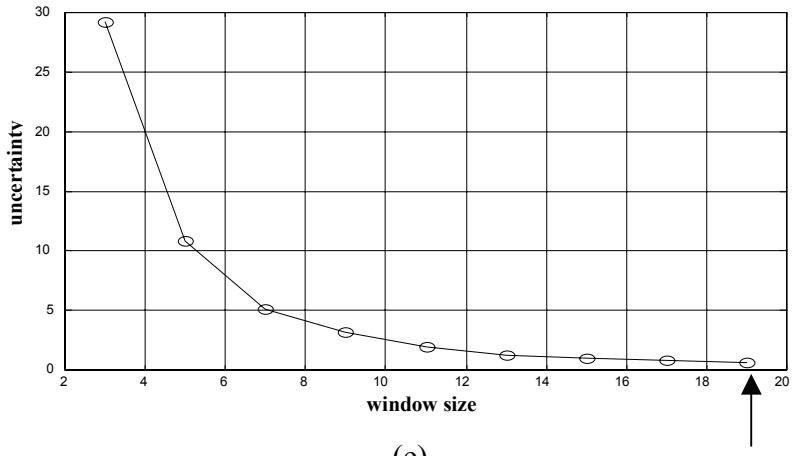
Figure 6.17. Examples of window size selection with real image using Kanade approach. The same targets in Figure 6.15 were used. (a) The "cone" image. (b)-(e) Plots of uncertainty as a function of *w* for the four targets shown ordered (left to right and top to bottom).



(c`)



(d)



(e)

Figure 6.17. Continued, (c)-(e). Arrows indicate the window size that was selected.

Figure 6.18 contains another example image for several target points in the image using the new method. To compare the results of the new approach with the Kanade approach, we applied the Kanade approach on the same targets that were used in Figure 6.18. In this example, we selected the threshold to yield windows of approximately the same size like the new approach for this image (see Figure 6.19).

Figure 6.20 contains another example image for several target points in the image using the new method. It can be noticed that the windows are good matches to the image textures. To compare the results of the new approach with the Kanade approach, we applied the Kanade approach on the cone image for the same targets that were used in Figure 6.20. Several of these windows seem too small for optimum matching (see Figure 6.21).

Figure 6.22 contains another example image for several target points in the image using the new method. To compare the results of the new approach with the Kanade approach, we applied the Kanade approach with an appropriate threshold on the same targets that were used in Figure 6.22. It can be noticed that the window sizes that were selected by Kanade approach are the same like the new approach (see Figure 6.23).

Figure 6.18. Examples of window size selection with "tree" image using the new approach. Four targets are shown, along with the resulting windows. A considerable degree of texture is present and the resulting windows are good matches to the image textures.

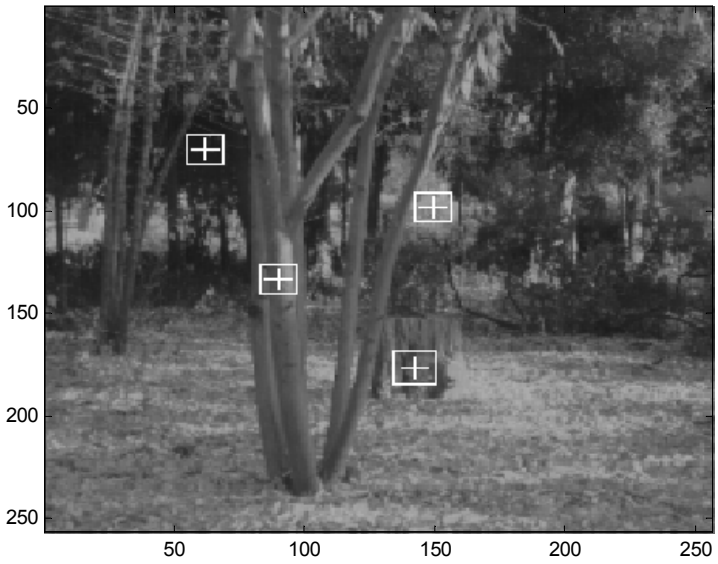


Figure 6.19. Examples of window size selection with "tree" image using Kanade approach. The same targets in Figure 6.18 were used. Four targets are shown, along with the resulting windows. A considerable degree of texture is present and these results in a relatively small window size.

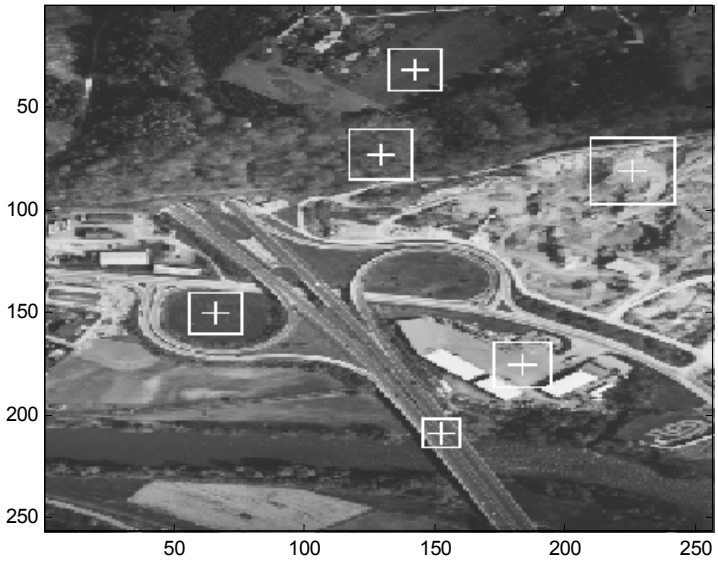


Figure 6.20. Examples of window size selection with real image using the new approach. Four targets are shown, along with the resulting windows. A considerable degree of texture is present and the resulting windows are good matches to the image textures.

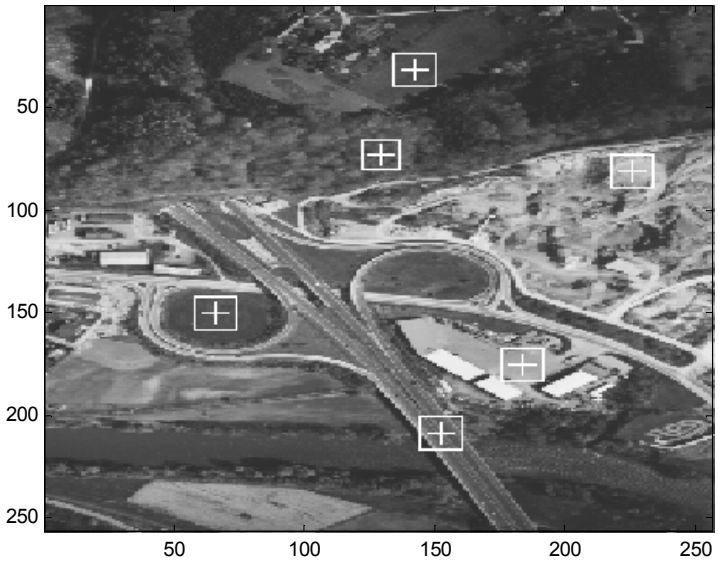


Figure 6.21. Examples of window size selection with real image using Kanade approach. The same targets in Figure 6.20 were used. Four targets are shown, along with the resulting windows. A considerable degree of texture is present and several of these windows seem too small for optimum matching.

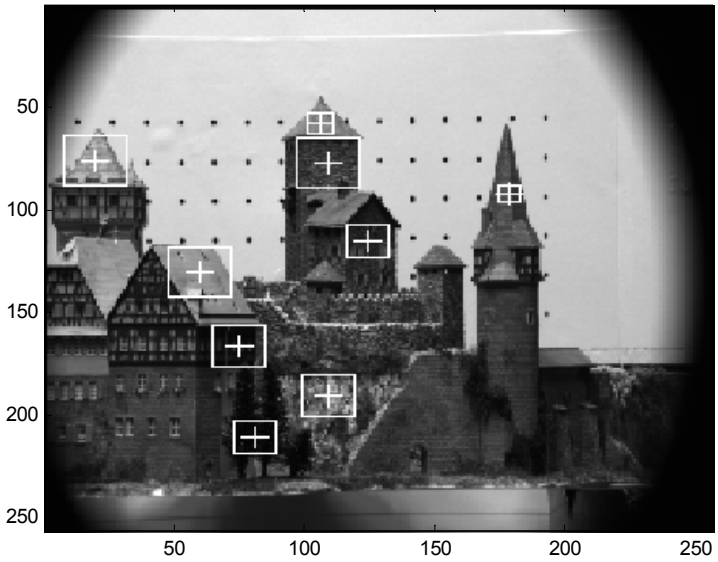


Figure 6.22. Examples of window size selection with real image using the new approach. Nine targets are shown, along with the resulting windows. A considerable degree of texture is present and the resulting windows are good matches to the image textures.

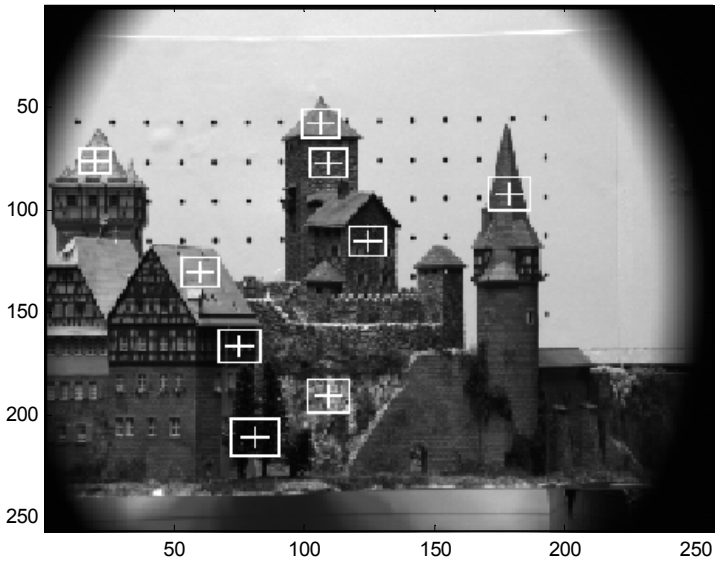


Figure 6.23. Examples of window size selection with real image using Kanade approach. The same targets in Figure 6.22 were used. Nine targets are shown, along with the resulting windows. A considerable degree of texture is present and several of these windows seem too small for optimum matching.

6.5 Moment Invariants and Quantization Effects

6.5.1 Introduction

To what extent are image invariants actually invariant? This question is important because of the growing popularity of invariants for such tasks as image reconstruction [Liao96], registration [Flus94a], pattern recognition [Duda77, Flus93, Flus94b, Gupt87, and Prok92], and tracking [Mata94]. Invariants are often used for these tasks because the analyses involve imaged entities that undergo changes in size, position, and orientation.

Moments are among the most common means of deriving invariants. However, moment invariants are truly invariant only for functions defined in the continuous domain. Quantization of the image plane is necessary, because otherwise the image cannot be processed digitally. Image acquisition by a digital system imposes spatial and intensity quantization that, in turn, introduce errors into moment and invariant computations.

Considerable insight may be gained by considering a simple rectangular region. Using assumptions similar to those given in [Teh86], let *f* represent a binary image (i.e., $f(x,y) \in \{0,1\}$), with $f(x,y) = 1$ if and only if $-a/2 \le x \le a/2$ and $-b/2 \le y \le b/2$. The moment equations then simplify to the following:

$$m_{pq} = \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} x^p y^q dx dy \tag{6.26}$$

$$\mu_{pq} = \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} (x - \bar{x})^p (y - \bar{y})^q dx dy \ . \tag{6.27}$$

In general, the transformation of image *f* into its digitized version $\tilde{f}$ involves a quantization of both the domain and range of the original image. For simplicity, we consider only the former, which typically consists of sampling the image function at a rectangular grid of (*x, y*) locations.

Letting Δx and Δy represent the sampling intervals in the horizontal and vertical directions, respectively, our quantization model is $\tilde{f}(i,j) = f(i\Delta x, j\Delta y)$, for all integers *i* and *j*, as illustrated in Figure 6.24.

Referring to equations (6.26, 6.27), moments and central moments for the rectangular digital image $\tilde{f}$ may be defined as follows,

$$\tilde{m}_{pq} = \sum_{i=-\lfloor a/(2\Delta x)\rfloor}^{\lfloor a/(2\Delta x)\rfloor} \sum_{j=-\lfloor (b/2\Delta y)\rfloor}^{\lfloor b/(2\Delta y)\rfloor} (i\Delta x)^p (j\Delta y)^q \tag{6.28}$$

$$\tilde{\mu}_{pq} = \sum_{i=-\lfloor a/(2\Delta x)\rfloor}^{\lfloor a/(2\Delta x)\rfloor} \sum_{j=-\lfloor b/(2\Delta y)\rfloor}^{\lfloor b/(2\Delta y)\rfloor} ((i\Delta x)-\bar{i})^p ((j\Delta y)-\bar{j})^q \tag{6.29}$$

where $\bar{i} \equiv \tilde{m}_{10}/\tilde{m}_{00}$ and $\bar{j} \equiv \tilde{m}_{01}/\tilde{m}_{00}$.

6.5.2 Hu Moment Invariants and Spatial Quantization

For simplicity, we continue to consider the case that f represents a rectangular region of size $b \times a$. It is not difficult to show, using equations (6.3, 6.26, 6.27), that in this case the first Hu invariant is

$$\phi_1 = \frac{a^2+b^2}{12ab}. \tag{6.30}$$

Similarly, the first Hu moment invariant $\tilde{\phi}_1$ for the quantized image $\tilde{f}$ can be computed using (6.28, 6.29) and $\tilde{\eta}_{pq} = \tilde{\mu}_{pq} / \tilde{\mu}_{00}^{\ \gamma}$:

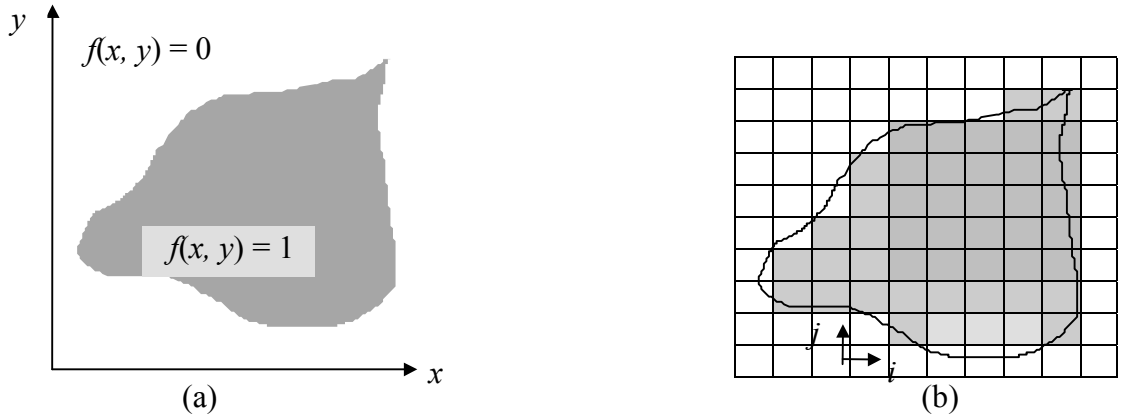


Figure 6.24. Spatial quantization of a binary image. An ideal image f in the continuous domain (a) is converted to discrete version $\tilde{f}$ (b) by sampling at locations $(x, y) = (i\Delta x, j\Delta y)$. In the text, the region defined by $f = 1$ is assumed to be rectangular in shape.

$$\tilde{\phi}_1 = \frac{\Delta x^2 \left\lfloor \frac{a}{2\Delta x} \right\rfloor \left(1 + \left\lfloor \frac{a}{2\Delta x} \right\rfloor \right) + \Delta y^2 \left\lfloor \frac{b}{2\Delta y} \right\rfloor \left(1 + \left\lfloor \frac{b}{2\Delta y} \right\rfloor \right)}{3\Delta x \Delta y \left(1 + 2\left\lfloor \frac{a}{2\Delta x} \right\rfloor \right) \left(1 + 2\left\lfloor \frac{b}{2\Delta y} \right\rfloor \right)} \tag{6.31}$$

The quantization error for ϕ_1 is given by

$$e_{h1} = \phi_1 - \tilde{\phi}_1 . \tag{6.32}$$

As expected, it can be seen from (6.30-6.31) that the error tends toward 0 as the image size (given by *a* and *b*) increases, or as the sampling intervals (given by Δx and Δy) decrease. However, it is interesting to note that e_{h1} does *not* decrease monotonically in general. To illustrate this, three plots of e_{h1} are shown in Figure 6.25.

In general, for the rectangular case, discontinuities in e_{h1} exist at image sizes $a = m(2\Delta x)$ and $b = n(2\Delta y)$, for *m, n* = 0, 1, 2, ... When *a* = *b*, as shown in Figure 6.25(a), e_{h1} decreases monotonically for the case of square pixels, $\Delta x = \Delta y$. However, when $\Delta x > \Delta y$, e_{h1} may be characterized by a sequence of intervals, $m(2\Delta x) \le a < (m+1)(2\Delta x)$, with each specified by a given value of *m*. (A similar analysis holds for $\Delta y > \Delta x$.) Within each of these intervals, there exist $\left\lfloor \frac{\Delta x}{\Delta y} \right\rfloor$ "steps" determined by discontinuities in e_{h1}; careful examination shows that e_{h1} *increases* monotonically over the first half of this interval, and *decreases* monotonically over the second half of this interval. (We state this without formal proof, although we have confirmed this numerically for a large number of cases.) The maximum value of e_{h1} over interval *m* therefore occurs over a range of *a* that includes the midpoint of the interval, given by $a = \Delta x(2m+1)$.

For the simple case $\Delta x = \Delta y$ and *a* = *b* (square pixels and a square image region), the maximum error over interval *m* is given by

$$\max(e_{h1}) = \frac{1}{6(1+2m)^2} . \tag{6.33}$$

A useful upper bound for the error in this case is given by

$$e_{h1} \le \frac{1}{6\left(\frac{a}{\Delta x} - 1\right)^2}. \tag{6.34}$$

For the case $a = b$ but the pixels are not necessarily square, $\Delta x = 2\Delta y$, then it can be shown that

$$e_{h1} \le \frac{\Delta x \Delta y}{6\left(a^2 - (\Delta x + \Delta y)a + \Delta x \Delta y\right)}. \tag{6.35}$$

More generally, for the case $a = b$ but the pixels are not necessarily square, $\Delta x \ge 4\Delta y$, then it can be shown that

$$e_{h1} \le \frac{9\Delta x \Delta y - 4\Delta x^2 - 4\Delta y^2}{6\Delta x \Delta y\left(\frac{2a}{\Delta x} - 1\right)\left(\frac{2a}{\Delta y} - 1\right)}. \tag{6.36}$$

6.5.3 Hu Moment Invariants and Rotation

The previous section presented analytical expressions of error for a rectangular image region with sides that are aligned with the coordinate axes of the image. To investigate the quantization error further, we computed e_{h1} for synthetic images at several orientations. A square image region is used again to facilitate comparison with Section 6.4.2 and with other studies (e.g., [Teh86]).

Table 6.1 reports the minimum and maximum error values obtained for several image sizes, along with average absolute errors, for the case of rotation. To obtain these values, square synthetic regions were rotated in 2-degree increments from 0° to 45° and e_{h1} was computed using (6.18). As expected, the error is reduced for larger image sizes. Figure 6.26 contains a plot of e_{h1} for $a = 30\Delta x$ as a function of rotation angle.

6.5.4 AMIs and Spatial Quantization

We consider again the case that f represents a rectangular image region. In this case it can be shown that the first AMI, in the continuous domain, is

$$I_1 = \frac{1}{144}, \tag{6.37}$$

and the first AMI after quantization is

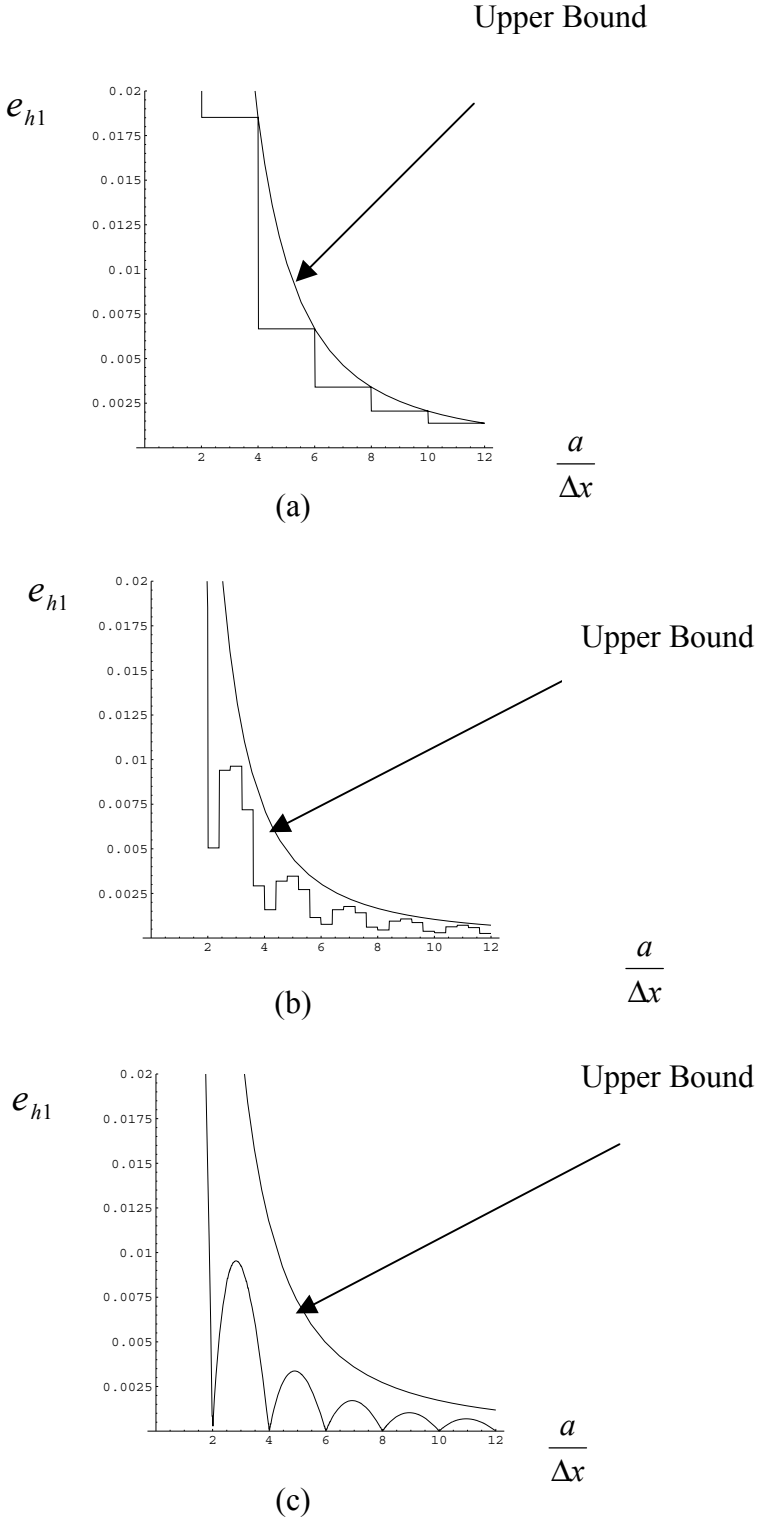


Figure 6.25. Quantization error for the first Hu moment invariant as a function of image size. In general, the error does not decrease monotonically. For these plots, the region is square (a = b). (a) The error e_{h1} is shown for square image pixels, $\Delta x = \Delta y$. (b) The error is shown for the case $\Delta x = 5\,\Delta y$. (c) The error is shown for the case $\Delta x = 100\,\Delta y$. This latter case represents an unusual sampling grid, but illustrates dramatically the effect of image quantization.

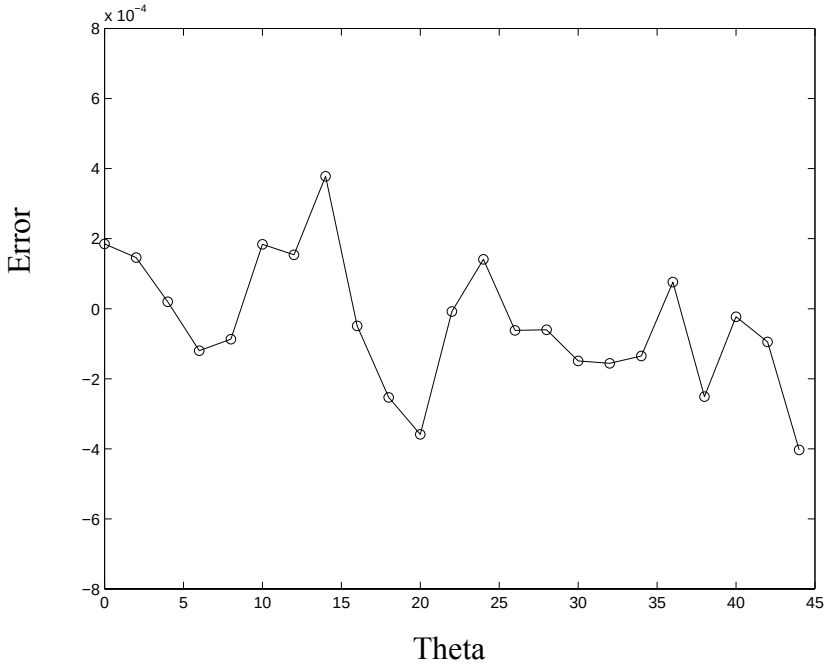


Figure 6.26. Quantization error for the first Hu moment invariant of a square image (a=30 Δx) as a function of rotation.

Table 6.1. Hu moment invariants and rotation. The quantization error e_{h1} for the first Hu moment invariant ϕ_1 is computed for square images.

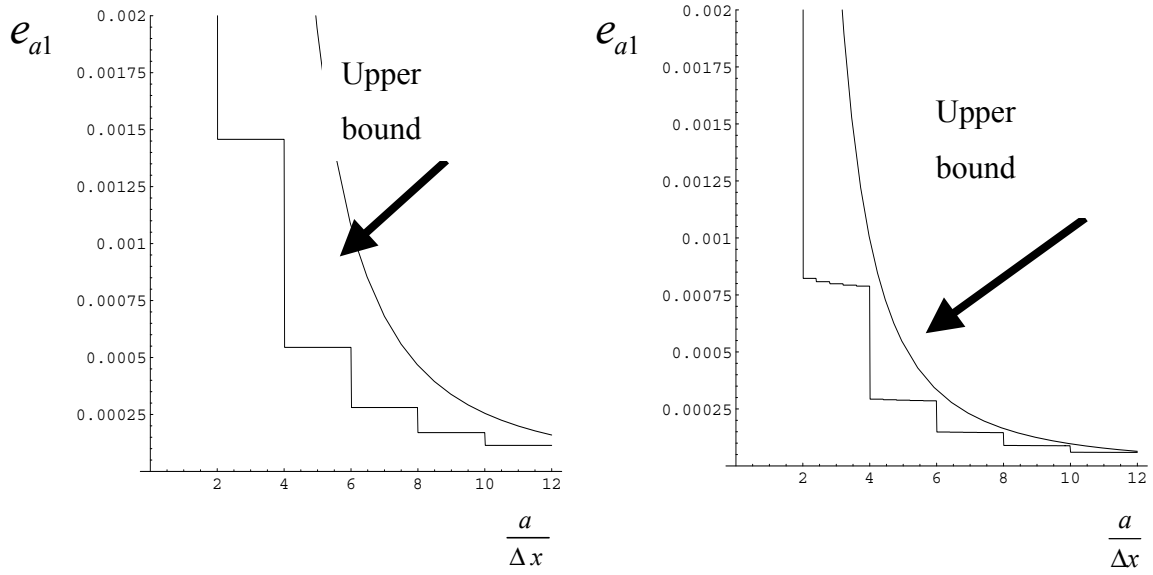
Image size ($a/\Delta x$)	min(e_{h1})	max(e_{h1})	mean of $\lvert e_{h1} \rvert$
10	-0.002120	0.001667	0.000757
20	-0.001289	0.001193	0.000401
30	-0.000403	0.000378	0.000151
40	-0.000205	0.000374	0.000119
50	-0.000212	0.000248	0.000106
60	-0.000113	0.000080	0.000063

$$\tilde{I}_1 = \frac{\left\lfloor \frac{a}{2\Delta x} \right\rfloor \left(1 + \left\lfloor \frac{a}{2\Delta x} \right\rfloor \right) \left\lfloor \frac{b}{2\Delta y} \right\rfloor \left(1 + \left\lfloor \frac{b}{2\Delta y} \right\rfloor \right)}{9 \left(1 + 2 \left\lfloor \frac{a}{2\Delta x} \right\rfloor \right)^2 \left(1 + 2 \left\lfloor \frac{b}{2\Delta y} \right\rfloor \right)^2}. \tag{6.38}$$

The quantization error of I_1 is given by

$$\begin{aligned} e_{a1} &= I_1 - \tilde{I}_1 \\ &= \frac{4 \left\lfloor \frac{a}{2\Delta x} \right\rfloor \left(1 + \left\lfloor \frac{a}{2\Delta x} \right\rfloor \right) + \left(1 + 2 \left\lfloor \frac{b}{2\Delta y} \right\rfloor \right)^2}{144 \left(1 + 2 \left\lfloor \frac{a}{2\Delta x} \right\rfloor \right)^2 \left(1 + 2 \left\lfloor \frac{b}{2\Delta y} \right\rfloor \right)^2} \end{aligned} \tag{6.39}$$

Unlike the case of the Hu error e_{h1}, the AMI quantization error e_{a1} does in fact decrease monotonically for rectangular images as the image size increases. This is illustrated in Figure 6.27 for two cases.



(a) (b)

Figure 6.27. Quantization error for the first affine moment invariant as a function of image size. The error decreases monotonically. For these plots, the image is

square ($a = b$). (a) The error e_{a1} is shown for square image pixels, $\Delta x = \Delta y$. (b) The error is shown for the case $\Delta x = 5\,\Delta y$.

It can be seen from (6.25) that, for a rectangular image, discontinuities in e_{a1} exist at image sizes $a = m(2\Delta x)$ and $b = n(2\Delta y)$, for m, n = 0, 1, 2, ... When $\Delta x \geq \Delta y$, e_{a1} may be characterized by a sequence of intervals, $m(2\Delta x) \leq a < (m+1)(2\Delta x)$, for given values of m. Within each of these intervals, there exist $\left|\frac{\Delta x}{\Delta y}\right|$ steps. Unlike the case for Hu moment invariants, however, it can be shown that e_{a1} decreases monotonically over the entire interval. The maximum value of e_{a1} over interval m therefore occurs at the start of the interval, i.e. for

$$a = m(2\Delta x) \tag{6.40}$$

for a rectangular image. For the case of a square image, $a = b$, the maximum error within this interval is

$$\max(e_{a1}) = \frac{4m(1+m)+(1+2m)^2}{144(1+2m)^4}. \tag{6.41}$$

An upper bound for the case of a square image and square image pixels can also be derived:

$$e_{a1} \leq \frac{2\frac{a}{\Delta x}\left(1+\frac{a}{2\Delta x}\right)+\left(1+\frac{a}{\Delta x}\right)^2}{144\left(\frac{a}{\Delta x}-1\right)^4}. \tag{6.42}$$

For the case that the pixels are not necessarily square, $\Delta x \neq \Delta y$, then it can be shown that

$$e_{a1} \leq \frac{2\frac{a}{\Delta x}\left(1+\frac{a}{2\Delta x}\right)+\left(1+\frac{a}{\Delta y}\right)^2}{144\left(\frac{a}{\Delta x}-1\right)^2\left(\frac{a}{\Delta y}-1\right)^2}. \tag{6.43}$$

6.5.5 The Effect of Rotation and Skew on AMIs with Quantization

Table 6.2 reports error values obtained for several image sizes, for the case of rotation. To obtain these values, square synthetic images were rotated in 2-degree

increments from 0° to 45° and e_{a1} were computed using (6.25). Figure 6.28 contains a plot of e_{a1} for $a = 30\Delta x$ as a function of rotation angle. Table 6.3 reports error values obtained for several image sizes, for the case of changes in skew, a_2 in (6.10). To obtain these values, square synthetic images were deformed by affine transform (using $a_1 = a_4 = 1$ and $a_3 = d_1 = d_2 = 0$) and e_{a1} was then computed using (6.25). Figure 6.29 contains a plot of e_{a1} for $a = 25\Delta x$ as a function of a_2.

Table 6.2. AMI and rotation. The quantization error e_{a1} for the first AMI (I_1) is computed for a square image.

Image size ($a/\Delta x$)	$\min(e_{a1})$	$\max(e_{a1})$	mean of $\lvert e_{a1} \rvert$
10	-0.000178	0.000138	0.000065
20	-0.000108	0.000099	0.000033
30	-0.000034	0.000032	0.000012
40	-0.000017	0.000032	0.000010
50	-0.000017	0.000021	0.000008
60	-0.000011	0.000007	0.000005

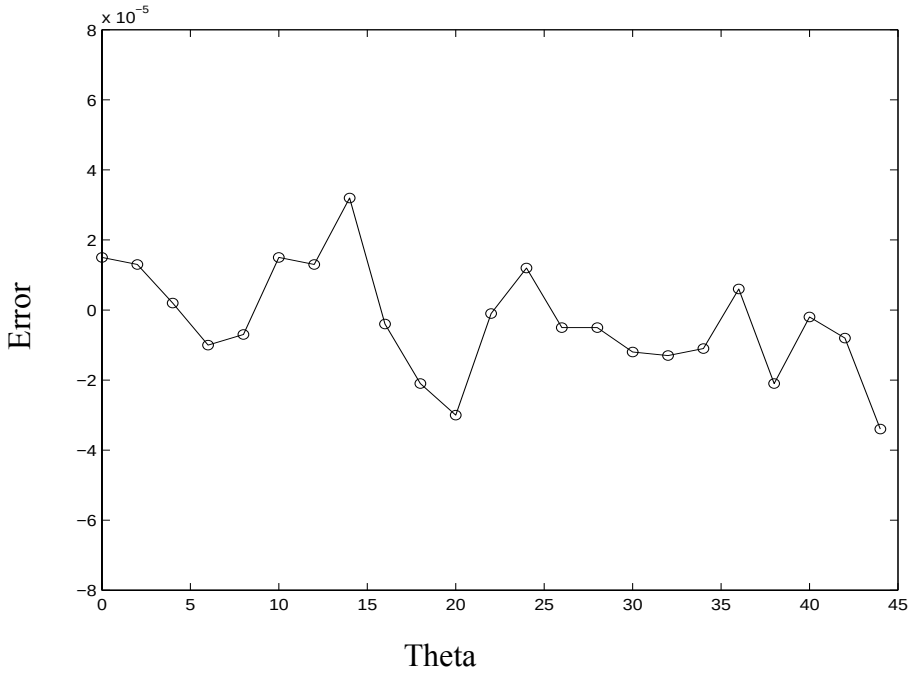


Figure 6.28. Quantization error for the first AMI of a square image (a=30 Δx) as function of rotation.

Table 6.3. Affine moment invariants and skew. The quantization error e_{a1} for the first AMI (I_1) is computed for a square image.

Image size ($a/\Delta x$)	min(e_{a1})	max(e_{a1})	mean of $\lvert e_{a1} \rvert$
10	0.000072	0.000138	0.000095
15	0.000027	0.000062	0.000036
20	0.000018	0.000035	0.000021
25	0.000010	0.000022	0.000013
30	0.000008	0.000015	0.000009
35	-0.000007	0.000011	0.000006
40	-0.000050	0.000009	0.000008
45	-0.000081	0.000007	0.000009
50	-0.000110	0.000006	0.000010

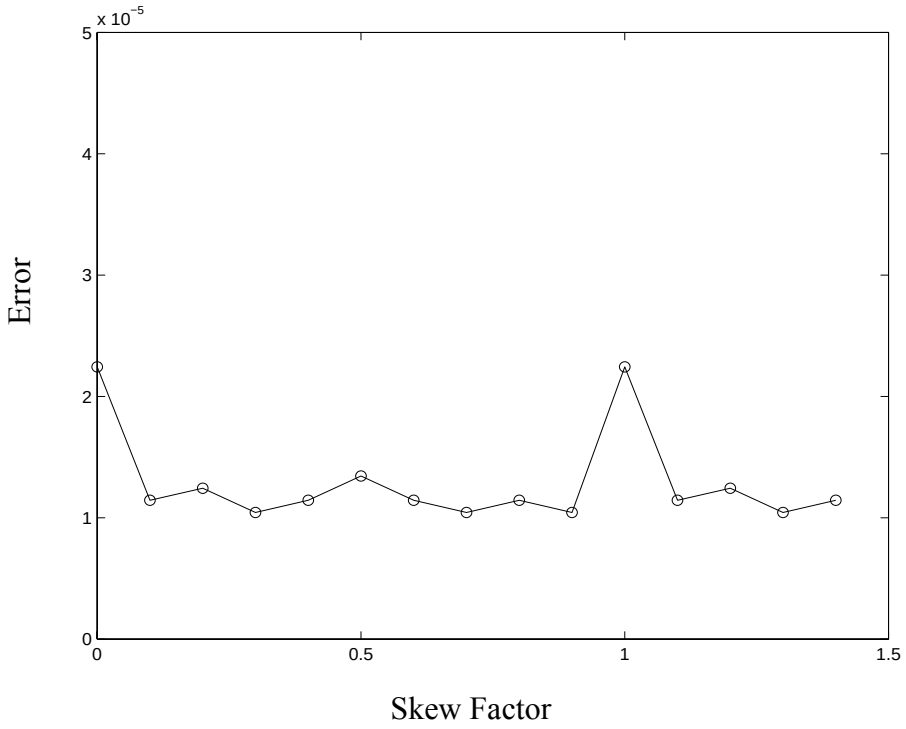


Figure 6.29. Quantization error for the first AMI of a square image (a=25 Δx) as function of skew (a_2).

Chapter 7

Experimental Results

7.1 Introduction

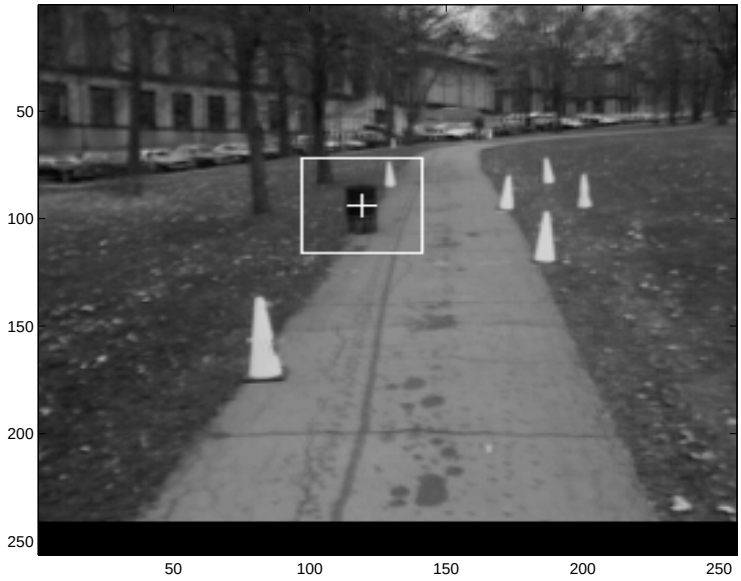
This chapter illustrates the effectiveness of the tracking system for both monocular and binocular image sequences in several situations. First, the effect of window size on object tracking is shown, and then the results for tracking occluded objects.

7.2 The Effect of Window Size on Object Tracking

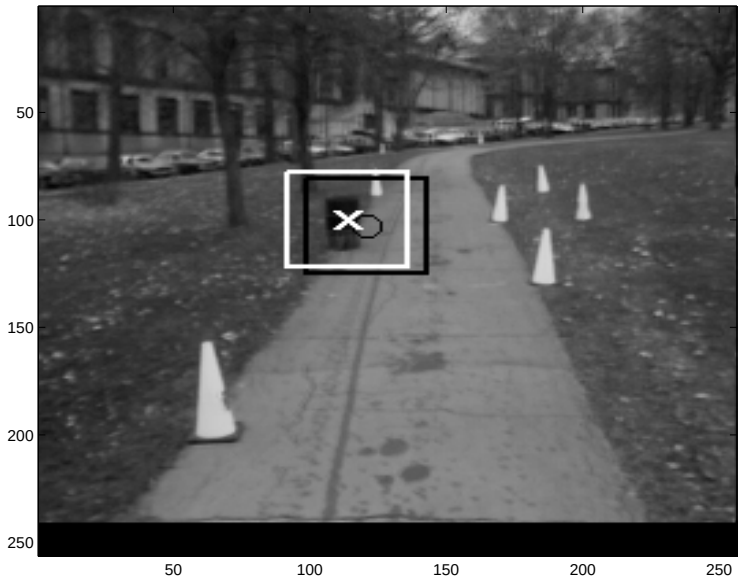
As described in Chapter 6, the search technique used here is very sensitive to the choice of window size, and we have developed a new approach based on the moment invariants to select that size. Intuitively, a window that is too small will tend to be distracted by texture primitives, or it may enclose a featureless region that is unsuitable for area-based comparisons. For a window that is too large, the similarity measurements may give incorrect matches due to repeating patterns, occlusion, and perspective differences.

To demonstrate the effect of window size selection on the tracking system in the monocular case, the example discussed earlier in Chapter 4 (see Figure 4.6) is re-analyzed using a user-defined window size. Previously, with the adaptive window size selection, the system successfully tracked the object until the last frame. In Figure 7.1, the user has specified a fixed window size. In this case, however, in frame 14 (Figure 7.1(c)) it can be noticed that the tracking window has been pulled toward the background, to the extent that the estimated target location no longer lies on the object of interest.

In another binocular image sequence, illustrated in Figure 7.2, the system was initialized by manual selection of corresponding points for a target in the initial left and right images, and in the subsequent left image. After that, the Kalman filter provided estimates of new image locations, and each was used as the starting point for a correspondence search. In this example, the user has specified a fixed window size in the first frame. In this case, however, in image pair 3 (Figure 7.2(c)), we can see that the tracking window has been drifted because the window size does not contain enough textures.

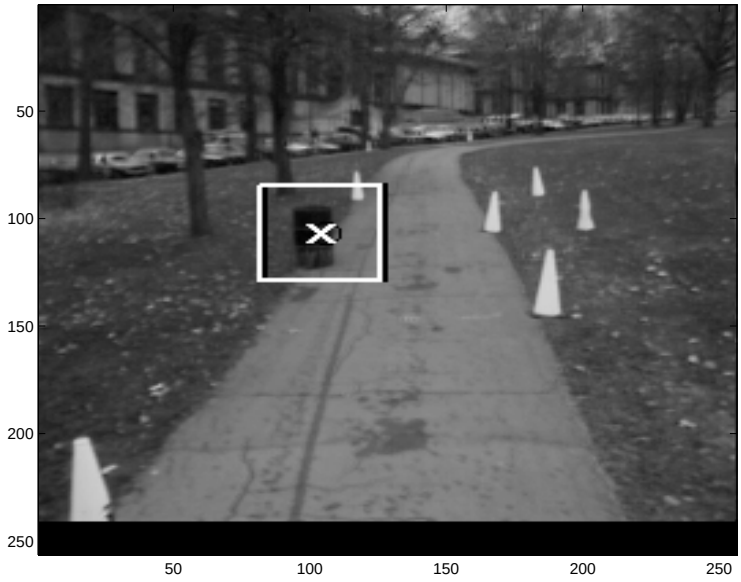


(a)

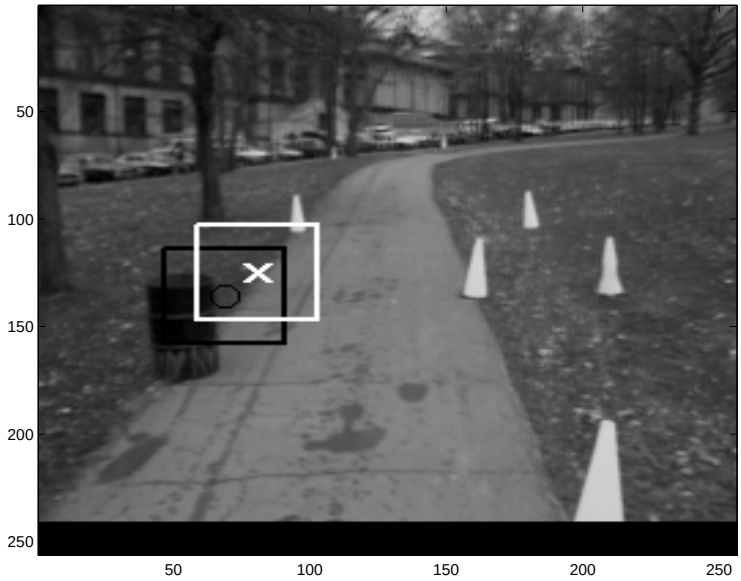


(b)

Figure 7.1. The effect of window sizes on the tracking method. Example of tracking a cone over a monocular sequence of 18 frames with a user defined window size. (a) Frame 1 of image sequence. (b) Frame 3. The white "+" indicates the desired target at the first frame. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.

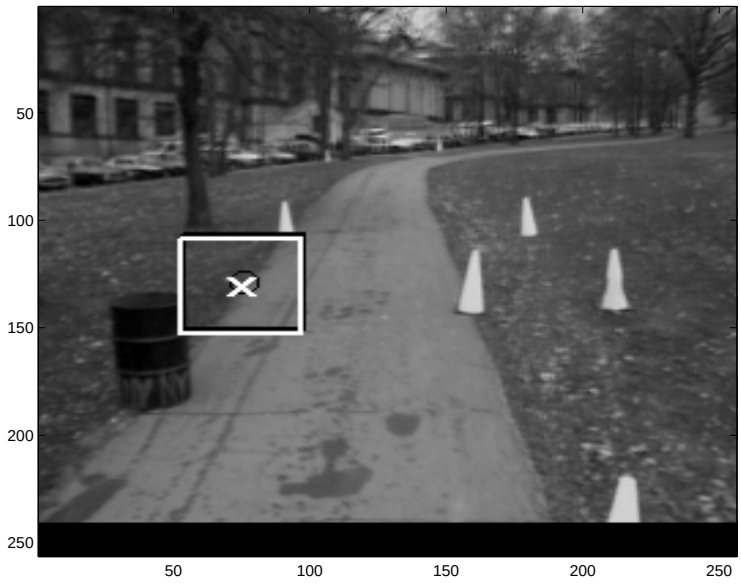


(c)



(d)

Figure 7.1, continued. (c) Frame 6. (d) Frame 14. However, in the middle of the sequence confuses the tracker and results in the target location leaving the cone of interest.



(e)

(f)

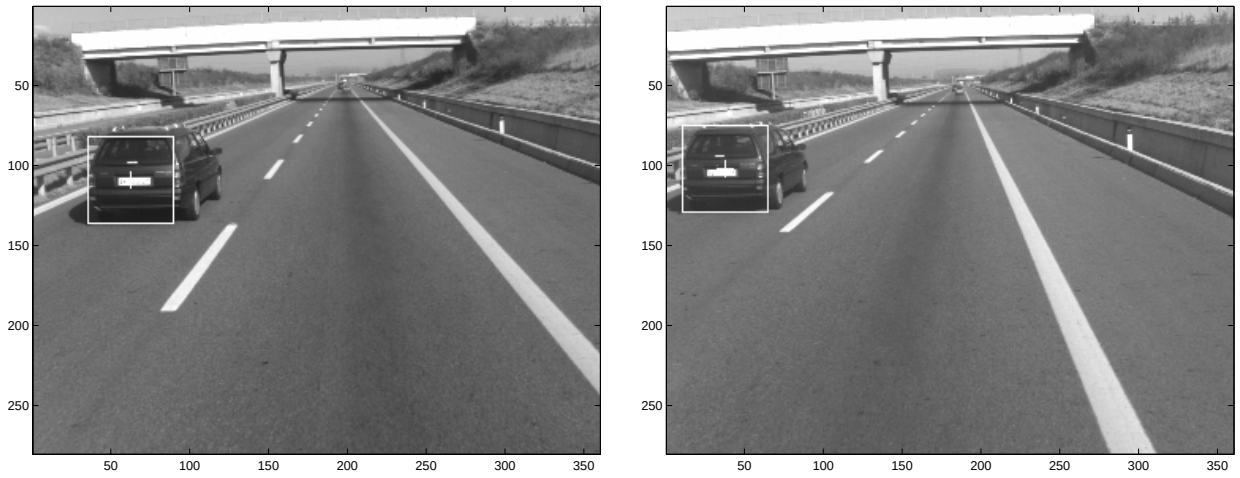
Figure 7.1, continued. (e) Frame 15. (f) Frame 18.

Meanwhile, in image pair 7 (Figure 7.2(d)) we can see that the tracking window has been pulled toward the background, to the extent that the estimated target locations no longer lies on the object of interest.

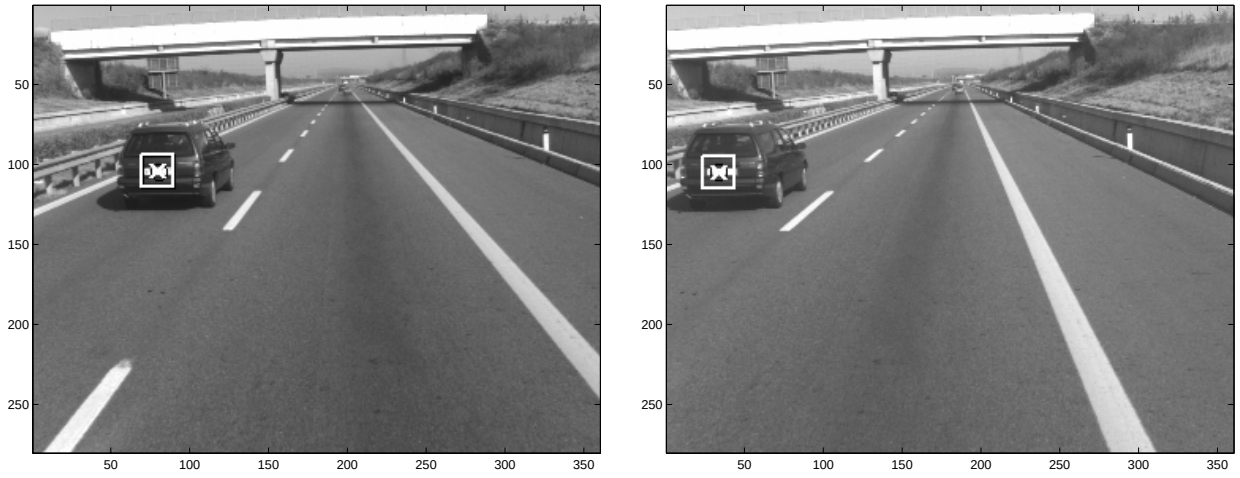
In summary, the area-based tracking system developed in this dissertation, along with approaches developed by other researchers, is highly sensitive to window size selection. The results presented in this section clearly show that the adaptive window size selection discussed in Chapter 4 has a superior performance over the user-specified selection.

7.3 Tests using Monocular Image Sequences

We have performed different experiments with different targets using our tracking system. The first experiment (Figure 7.3) was performed with the system tracking a person walking through the laboratory in a straight line. In this sequence, the desired target was selected manually in the first frame. After that, the Kalman filter provided estimates of new image locations, and these were used as the starting points for correspondence search. We use an adaptive search window for the desired target in the first frame, as described in Chapter 6. This sequence demonstrates the robustness of the tracking despite the presence, at times, of an area of background similar to the tracked object. In general, the system succeeds in keeping the target close to the image center. Figure 7.4 plots the residuals of matching against the frame number (Figure 7.4(a)), and pan steps against the frame number (Figure 7.4(b)). This system attempts to detect whether the tracked object is occluded or unoccluded according to the residual of the match between the associated region in the first and current frames; if the residual exceeds a user-defined threshold, the object assumed to be occluded. In Figure 7.4(a), it can be seen the residuals are all quite small, and this is because the object is not occluded.

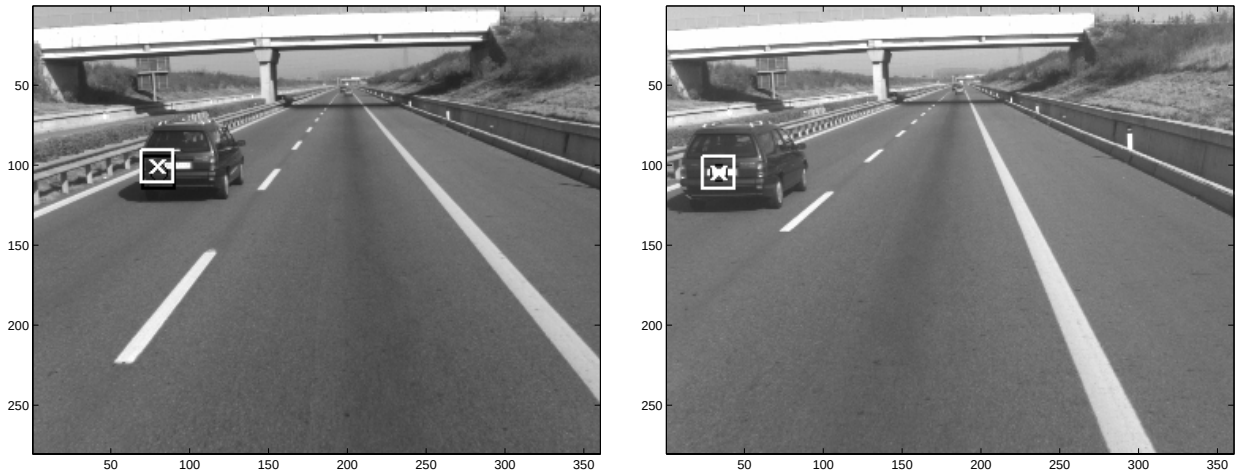


(a)

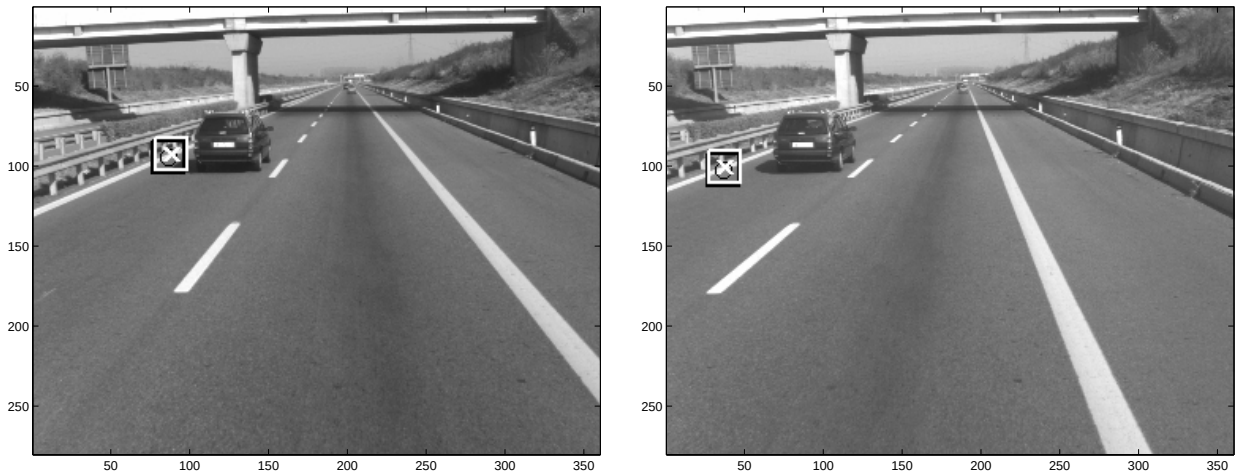


(b)

Figure 7.2. The effect of window sizes on the tracking method. Example of tracking a car over a binocular sequence of 18 frames with a user defined window size. (a) First image pair in sequence. (b) Image pair 2. The center of each black square denotes a point predicted by the Kalman filter. Each white "×" denotes the target detected by the correspondence search method.



(c)



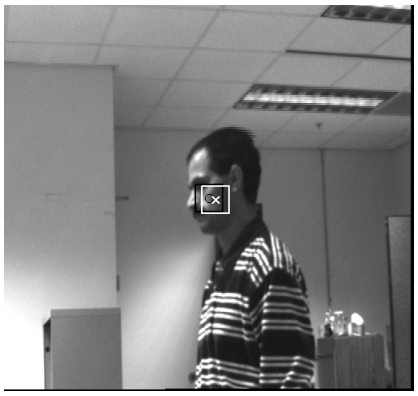
(d)

Figure 7.2, continued. Selected images from stereo "road" image sequence after applying the tracking algorithm. (c) Image pair 3. (d) Image pair 7.

(a)

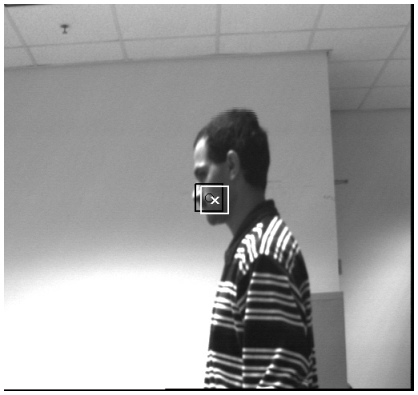
(b)

Figure 7.3. Selected images from "walking person" image sequence after applying the tracking algorithm. (a) First frame in sequence. The white "+" indicates the desired target and the white square represents the window size that is automaticaly selected well suited for the desired target in the first frame (b) Frame 2. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



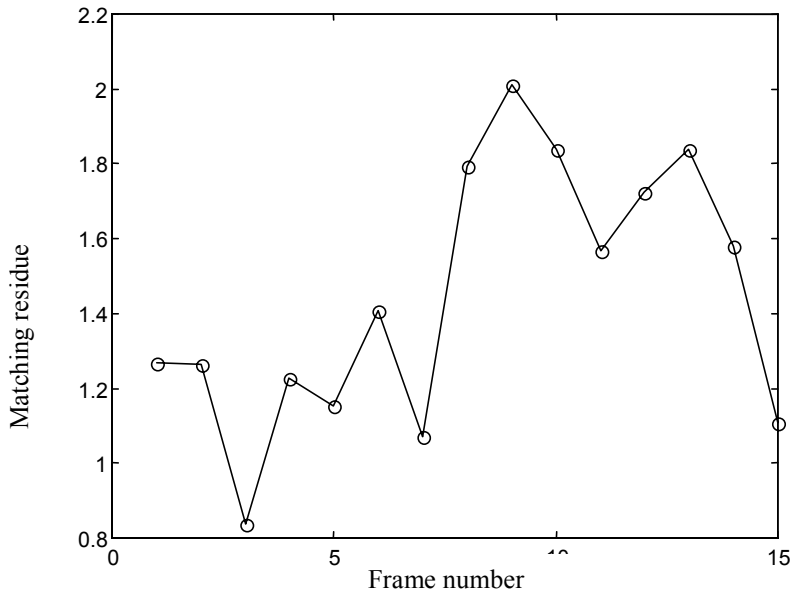
(c)

(d)

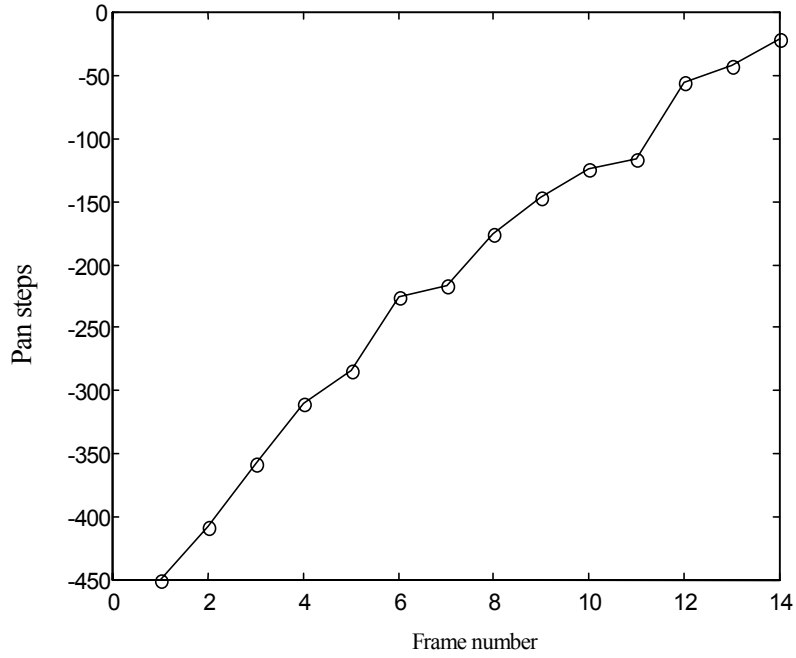


(e)

Figure 7.3, continued. (c) Frame 5. (d) Frame 8. (e) Frame 15.



(a)



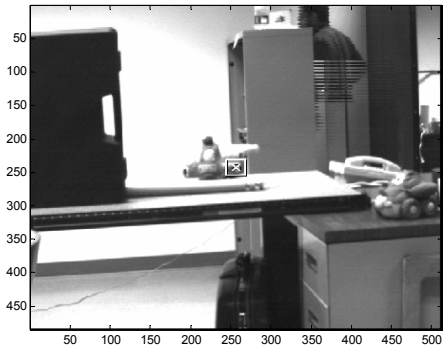
(b)

Figure 7.4. (a) Residual magnitudes versus frame number for the tracked person in Figure 7.3. (b) Pan camera position in steps versus frame number for the tracked person in Figure 7.3.

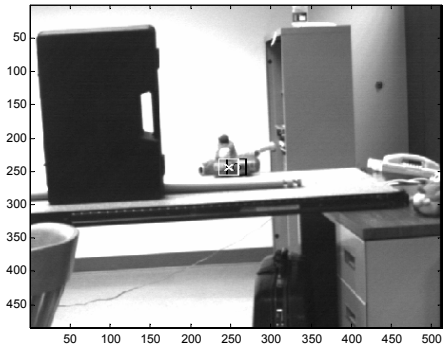
Another experiment was performed with the system tracking a moving car toy. The results are shown in the image sequence of Figure 7.5. We notice, as we move from one frame to the other, that the car toy becomes partly then completely hidden behind a barrier. Despite the occlusion, the system tracks the object successfully.

The important feature to note in this sequence, is that the desired target remains consistently on the back left section of the car toy through most of the sequence, although it has drifted slightly forward in the last frame. Some factors are contributing to this drift. First, if a part of the tracked object is occluded with another object so the result of the matching is inaccurate and this will affect on the measurement vector of Kalman filter that used to provide an initial estimate of the match location in the next frame. Secondly, if the desired target region matched to a similar region then the result of the matching may be accurate but this matching is related to another object. Despite of this drift, the matching compensates this drift and the system keeps the target close to the image center.

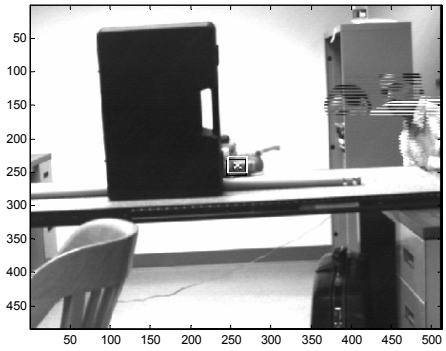
Figure 7.6, plots the residuals of matching versus the frame number (Figure 7.6(a)), and pan steps versus the frame number (Figure 7.6(b)). In Figure 7.6(a), it can be seen the residual of the match fluctuates less than the user-defined threshold until the object is occluded, the residual of the match becomes greater than the user-defined threshold (see Figures 7.6(a)).



(a)

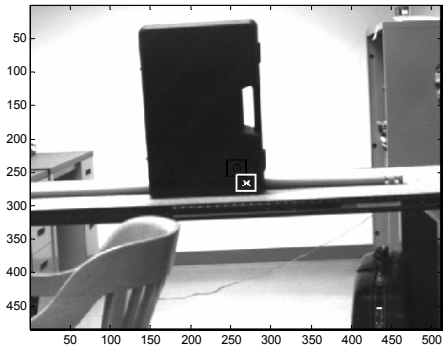


(b)

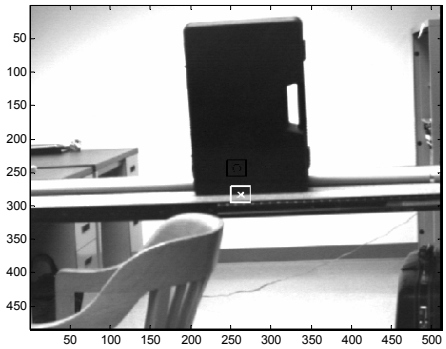


(c)

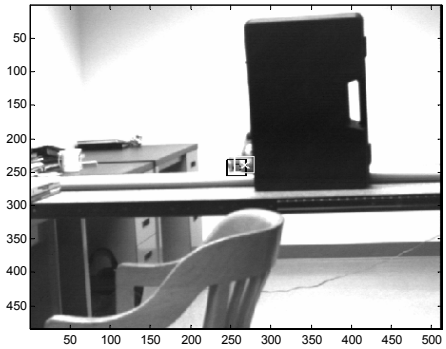
Figure 7.5. Selected images from "car toy" image sequence after applying the tracking algorithm. (a) Second frame in sequence. (b) Frame 4. (c) Frame 6.The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



(d)

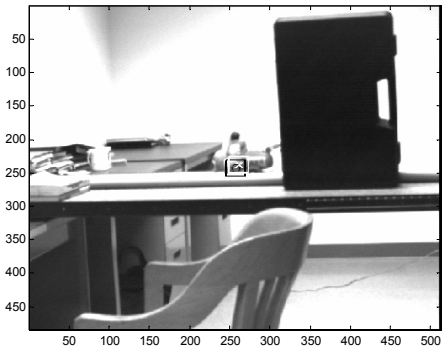


(e)

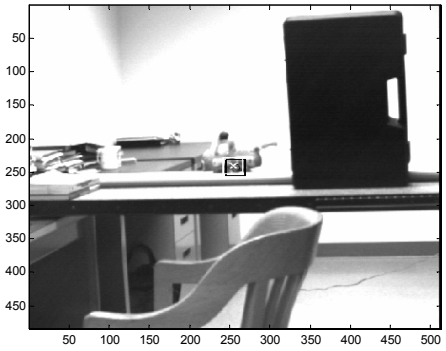


(f)

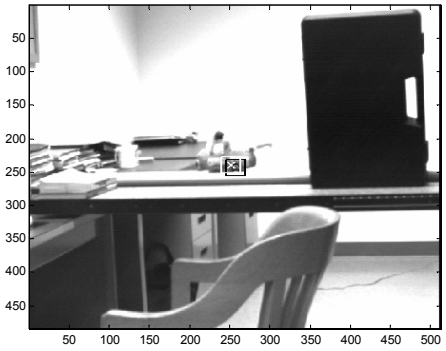
Figure 7.5, continued. (c) Frame 9. (d) Frame 10. (e) Frame 11.



(g)

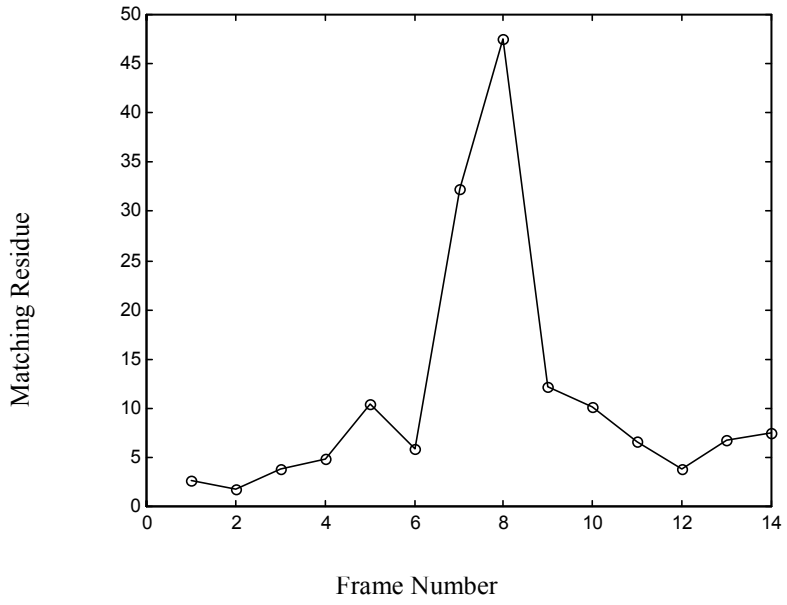


(h)

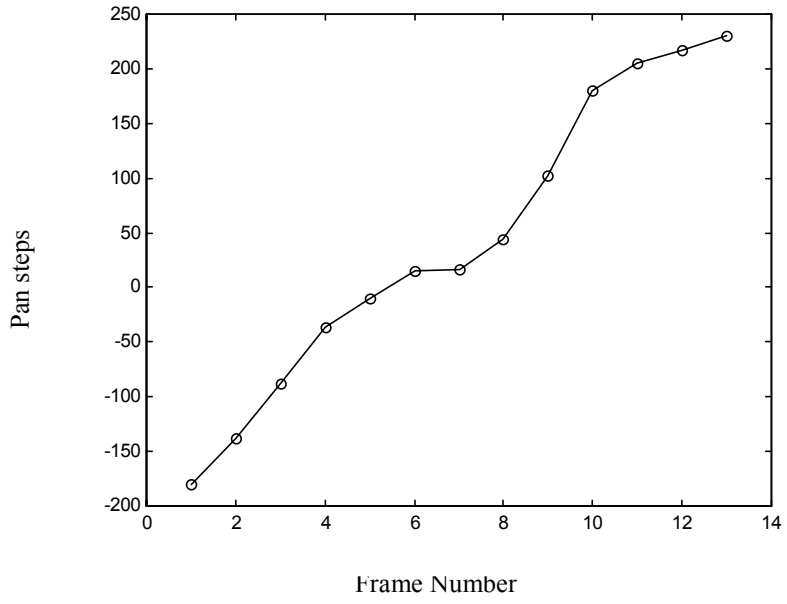


(i)

Figure 7.5, continued. (g) Frame 12. (h) Frame 13. (i) Frame 15.



(a)

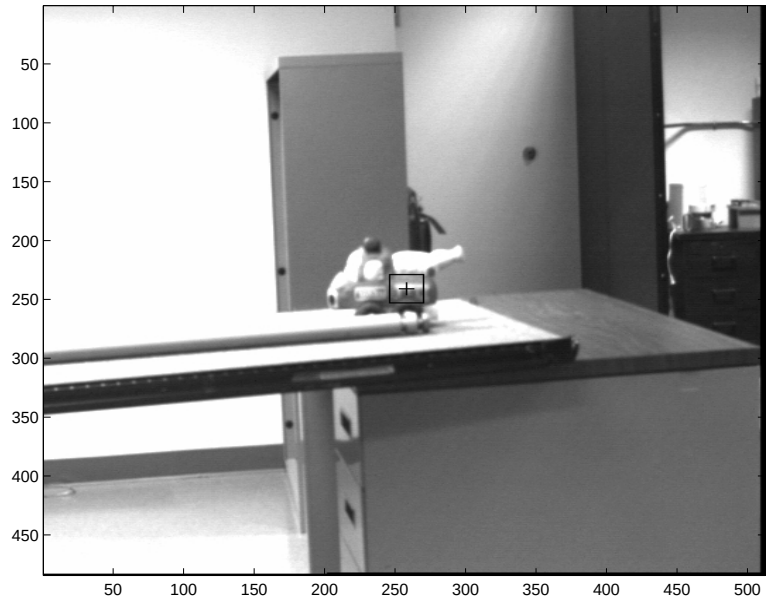


(b)

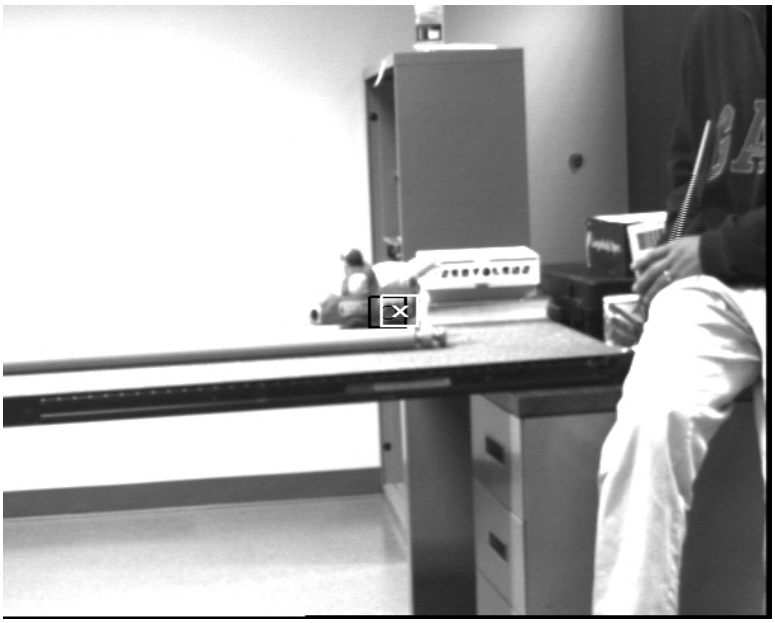
Figure 7.6. (a) Residual magnitude versus frame number for the tracked car toy in Figure 7.3. (b) Pan camera position in steps versus frame number for the tracked car toy in Figure 7.5.

Another experiment was performed with the system tracking a moving car toy. While the car toy moves toward a certain direction, it becomes occluded by another car toy coming from the opposite direction. The results are shown in the image sequence of Figure 7.7. A strong proof for the efficiency of the system performance is that despite the occlusion, the system tracked the object successfully.

Figure 7.8, plots the residuals of matching versus the frame number (Figure 7.8(a)), and pan steps against the frame number (Figure 7.8(b)).

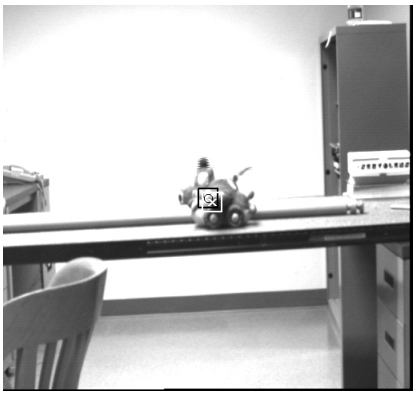


(a)



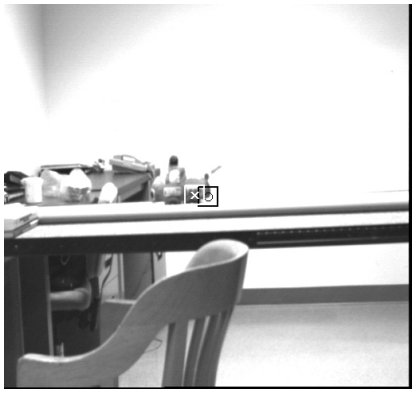
(b)

Figure 7.7. Selected images from "two car toys" image sequence after applying the tracking algorithm. (a) First frame in sequence. (b) Frame 3. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



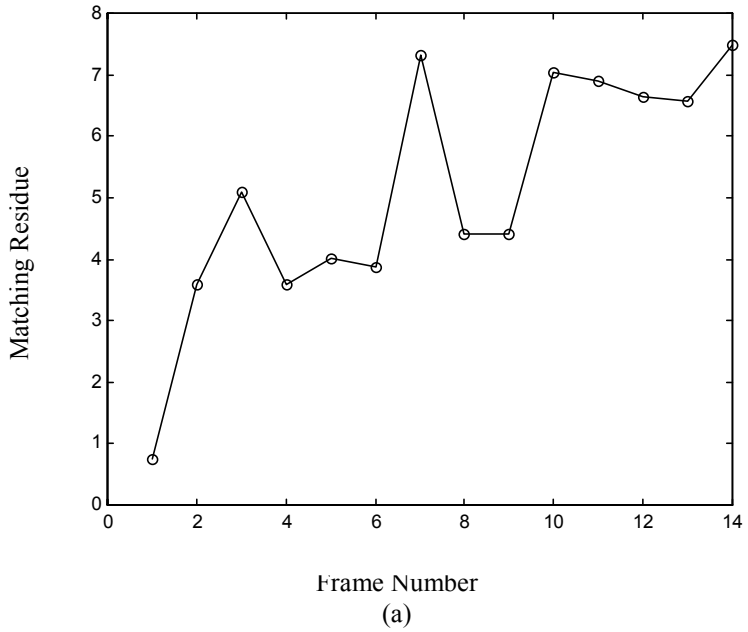
(c)

(d)

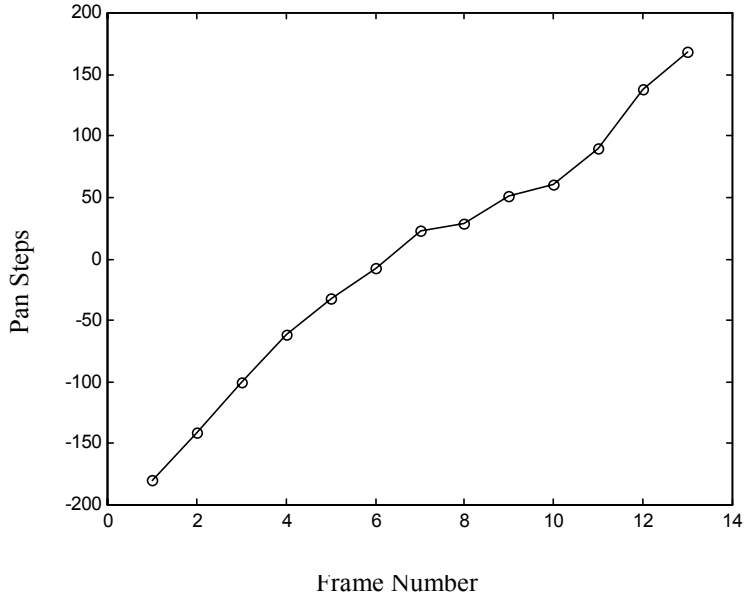


(e)

Figure 7.7, continued. (c) Frame 8. (d) Frame 9. (e) Frame 13.



(a)

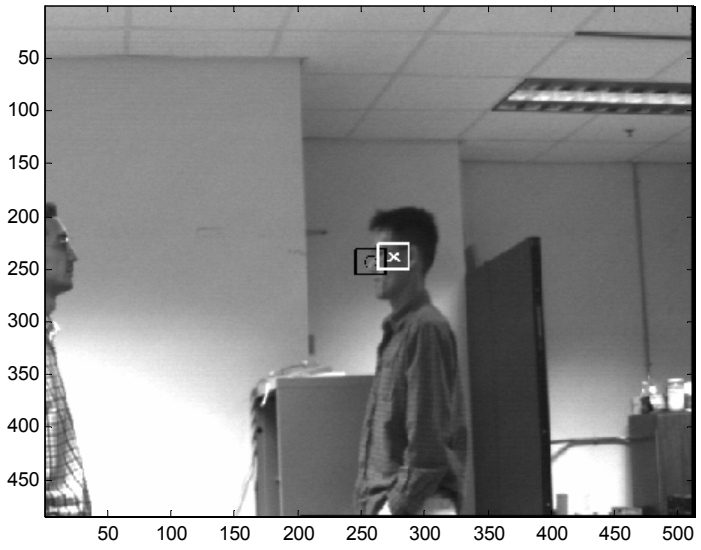


(b)

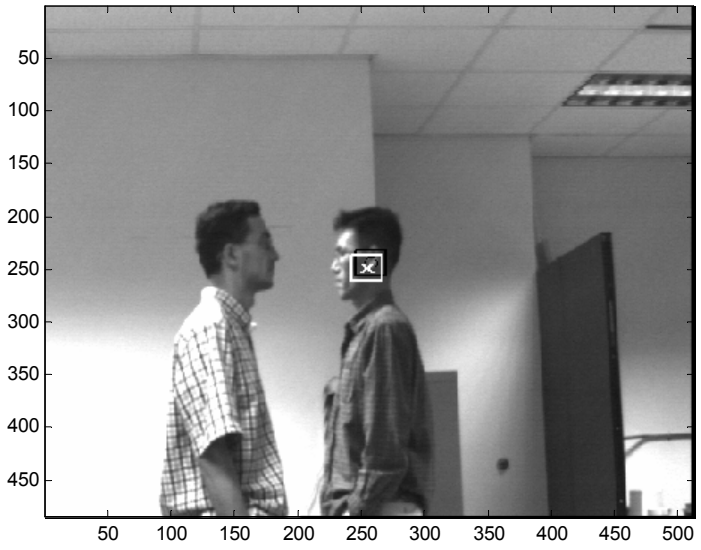
Figure 7.8. (a) Residual magnitude versus frame number for the tracked car toy in Figure 7.7. (b) Pan camera position in steps versus frame number for the tracked car toy in Figure 7.7.

Also, to show how the system performs in case of another moving target, an experiment has been conducted, tracking a walking person through the laboratory (see Figure 7.9.). In this experiment, another person moves in the opposite direction of the desired target, and he partially occludes the target as he moves. It can be noticed that in frame 9, Sang-Mook slowed down so the prediction is bad but the matching is able to compensate. Meanwhile in frame 9 Sang-Mook goes faster so the prediction is bad but the matching is able to compensate. Also, It can be noticed that while the occluding person passes by the target, the focus still remains on the desired target. This proves that without a good initial point of search, the system may loose the tracked target. Figure 7.10, plots the residuals of matching versus the frame number (Figure 7.10(a)), and pan steps versus the frame number (Figure 7.10(b)).

This sequence demonstrates the robustness of the tracking despite the presence, at times, of a partial occlusion of the tracked object. In general, the system succeeds in keeping the target close to the image center.

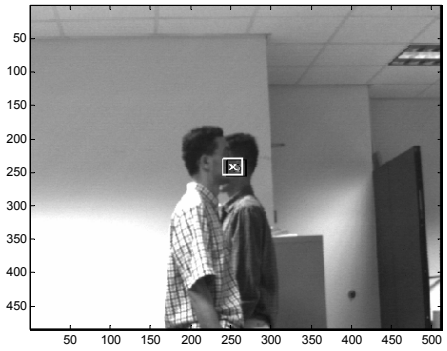


(a)

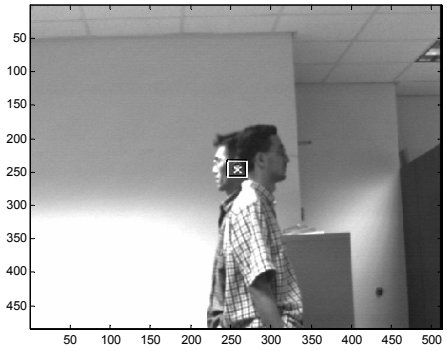


(b)

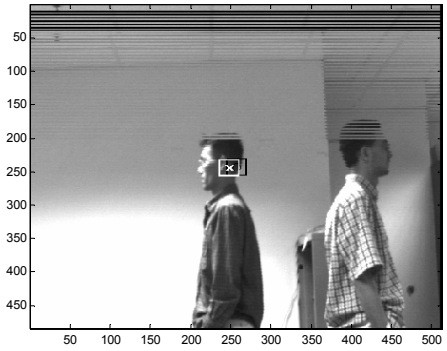
Figure 7.9. Selected images from "two walking persons" image sequence after applying the tracking algorithm. (a) Frame 3 in sequence. (b) Frame 5. The center of each black rectangle denotes a starting point predicted by the Kalman filter. Each white "×" indicates the target detected by the correspondence search method.



(c)

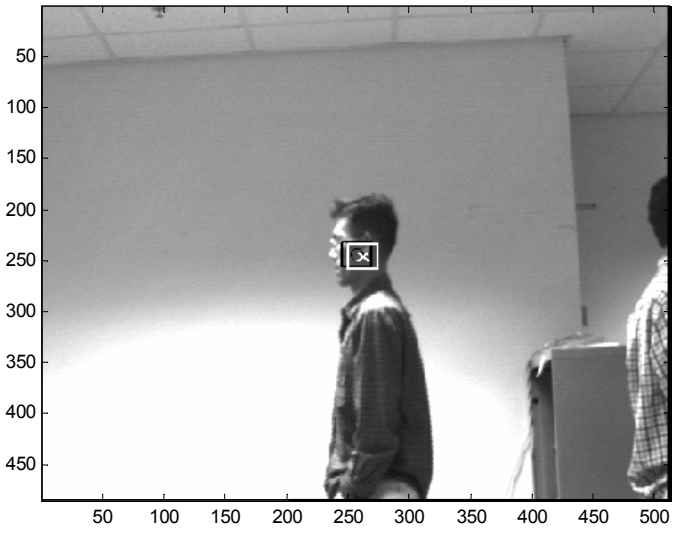


(d)

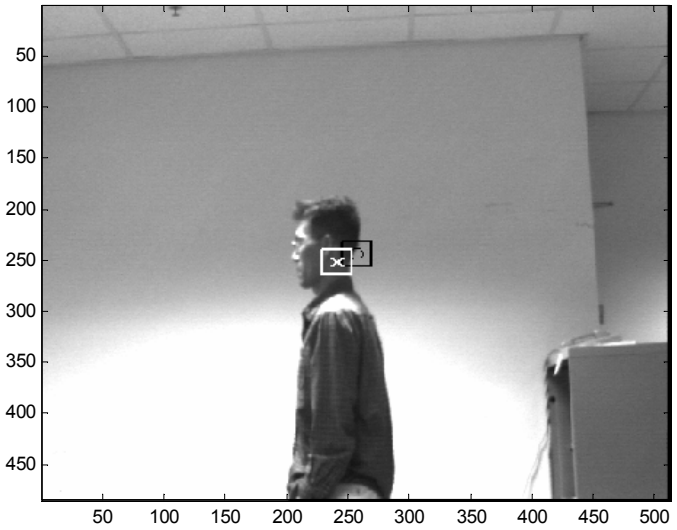


(e)

Figure 7.9, continued. (c) Frame 6. (d) Frame 7. (e) Frame 9. The occlusion occurs at frame 6 and 7.

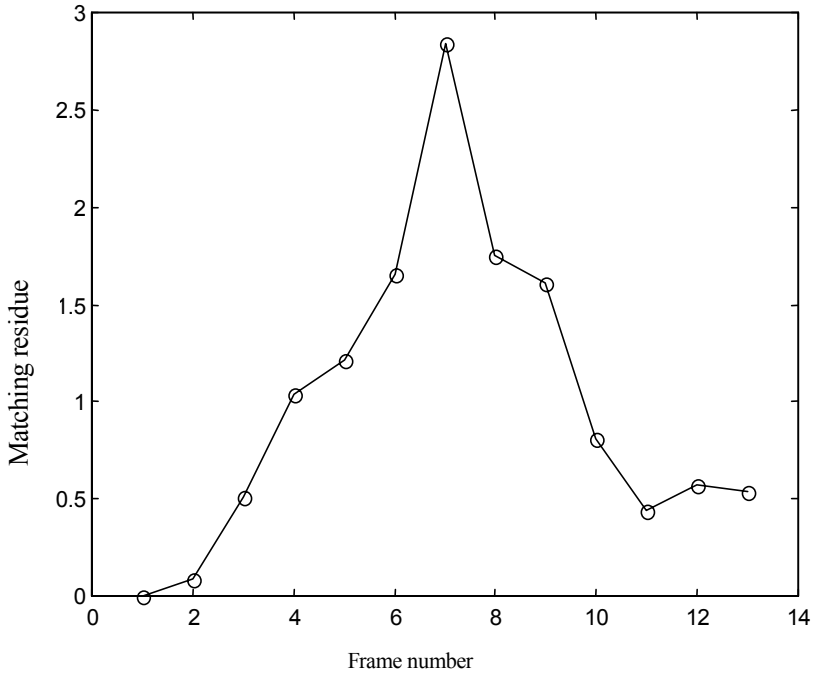


(f)

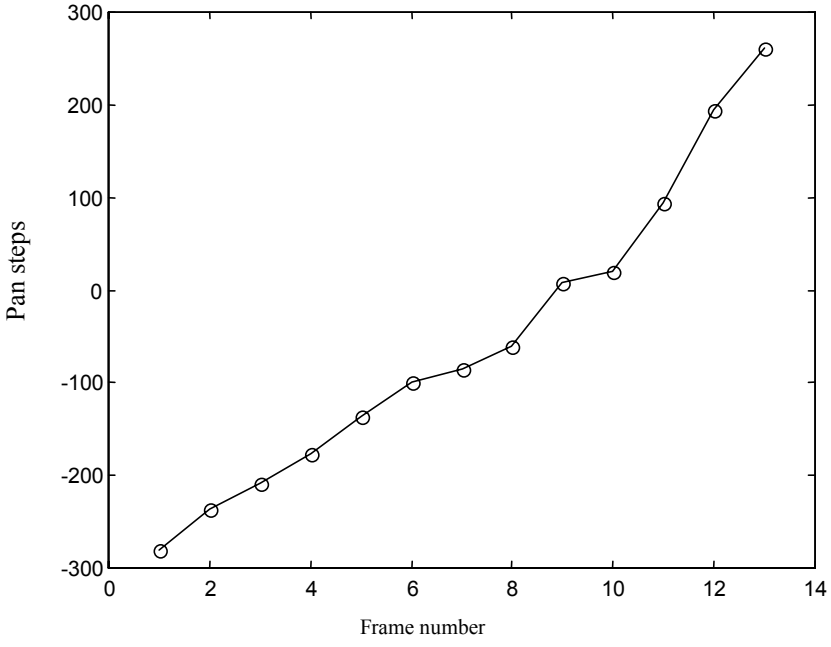


(g)

Figure 7.9, continued. (f) Frame 10. (g) Frame 12.



(a)



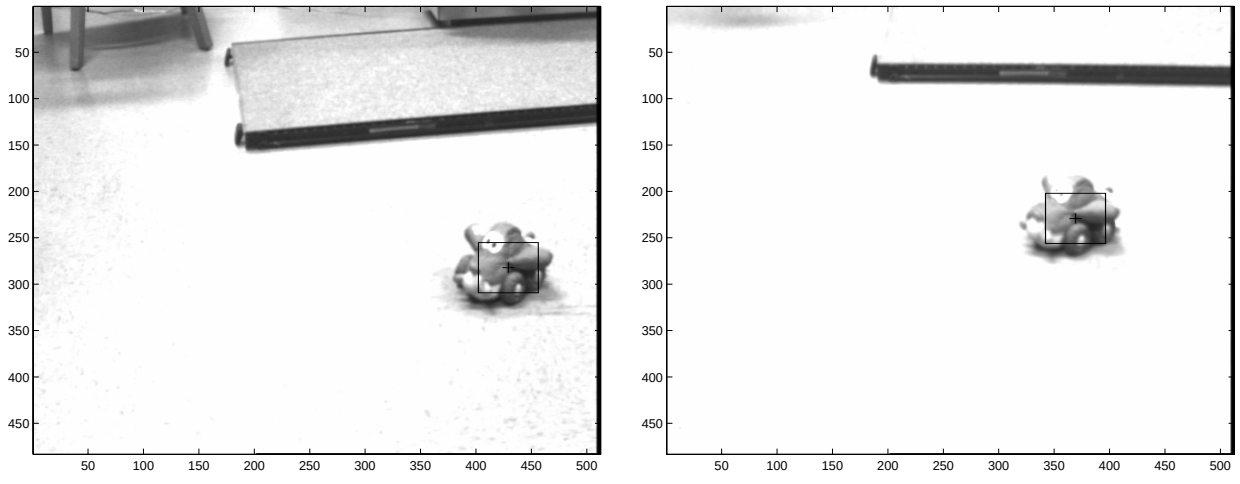
(b)

Figure 7.10. (a) Residual magnitude versus frame number for the tracked person in Figure 7.9. (b) Pan camera position in steps versus frame number for the tracked person in Figure 7.9.

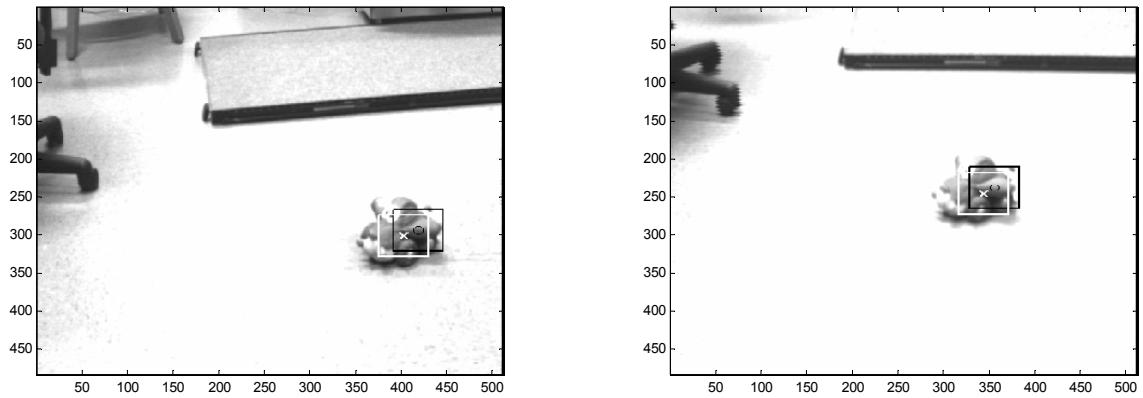
7.4 Tests using Binocular Image Sequences

Figure 7.11 presents an example of successful tracking. The system tracks a car toy in a stereo image sequence over 15 frames. The system was initialized by manual selection of corresponding points for a target in the initial left and right images, and in the subsequent left image.

In another binocular image sequence, the system tracks a car toy (see Figure 7.12). The system was initialized by manual selection of corresponding points for a target in the initial left and right images, and in the subsequent left image. Once the car toy moves behind a barrier, the system detects an occlusion. The matching residue increases as shown in Figure 7.13. In this case, the system continues to track the car based upon the predicted locations by Kalman filter not by the detected locations by matching. Once the car passes this barrier, the matching residue decreases to be quiet small again and the system tracks the car based upon the detected locations by matching. We can notice that the system detected an occlusion, starting from frame 5, using the left camera; meanwhile the system detects also an occlusion, starting from frame 8, using the right camera.

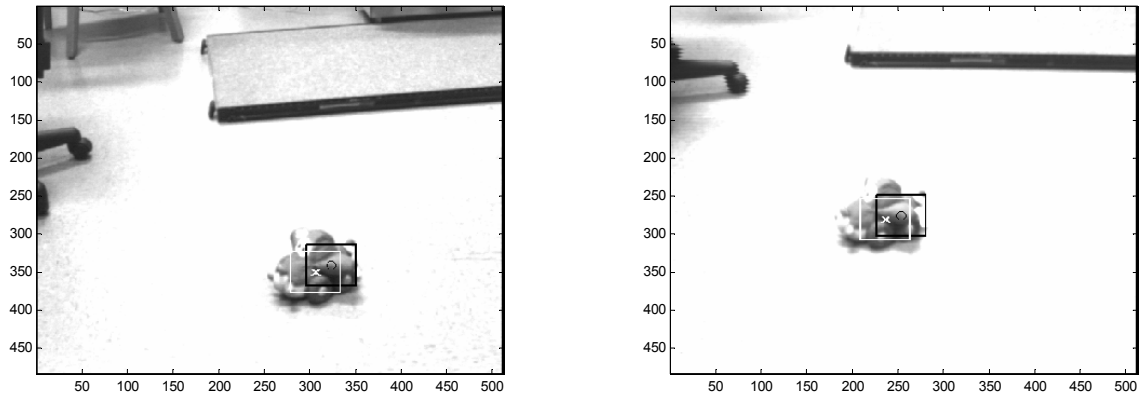


(a)

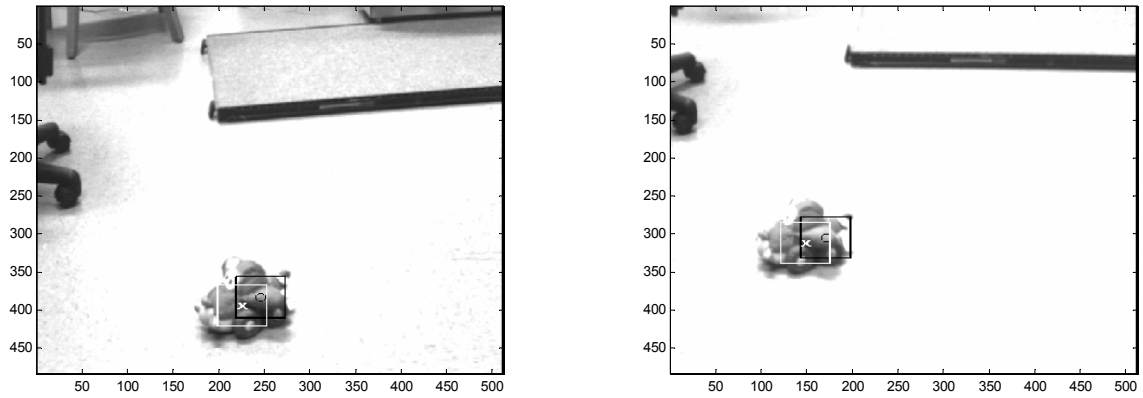


(b)

Figure 7.11. Selected images from stereo "car toy" image sequence after applying the tracking algorithm. (a) First image pair in sequence. (b) Image pair 3. The center of each black square denotes a point predicted by the Kalman filter. Each white "×" denotes the target detected by the correspondence search method.

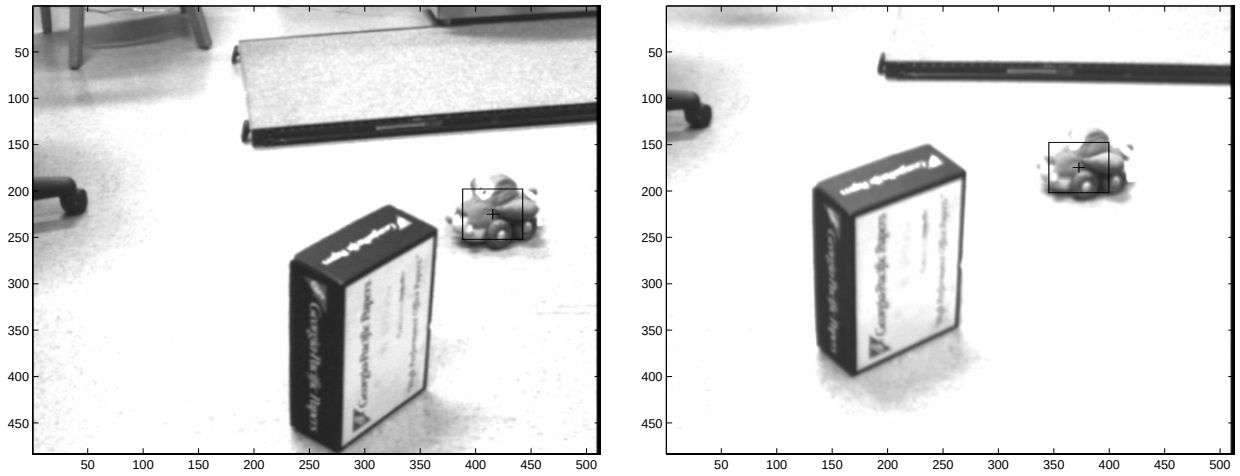


(c)

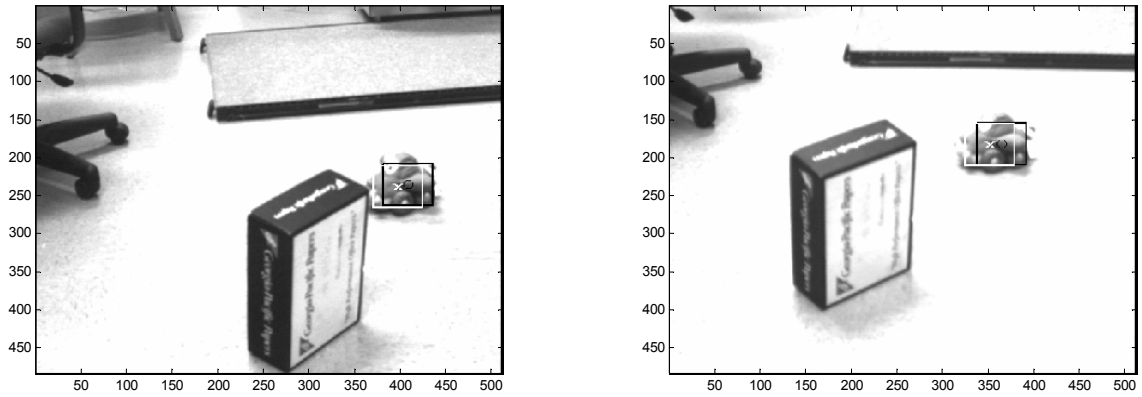


(d)

Figure 7.11, continued. (c) Image pair 10. (d) Image pair 14.

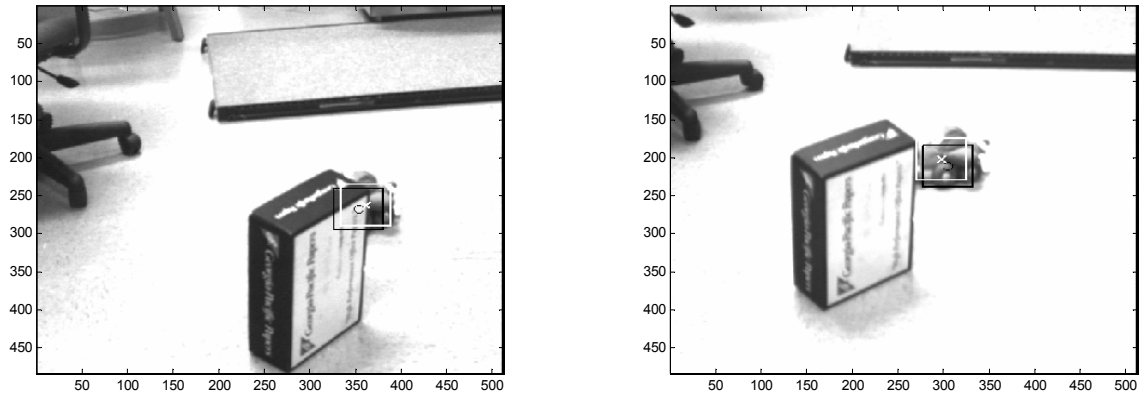


(a)

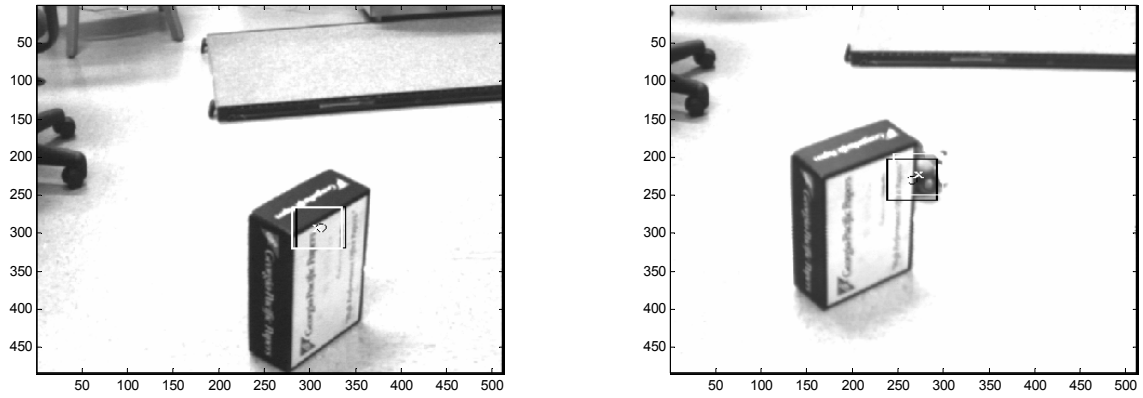


(b)

Figure 7.12. Selected images from stereo "car toy" image sequence after applying the tracking algorithm. (a) First image pair in sequence. (b) Image pair 3. The center of each black square denotes a point predicted by the Kalman filter. Each white "×" denotes the target detected by the correspondence search method.

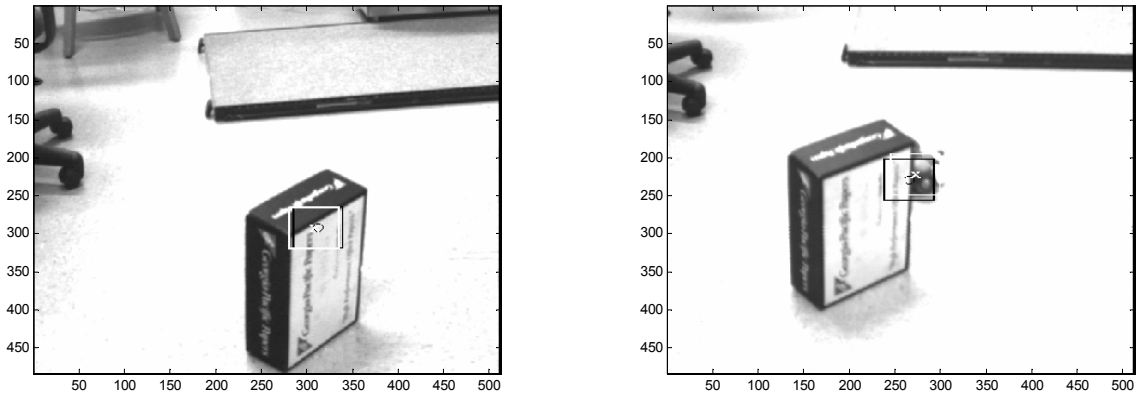


(c)

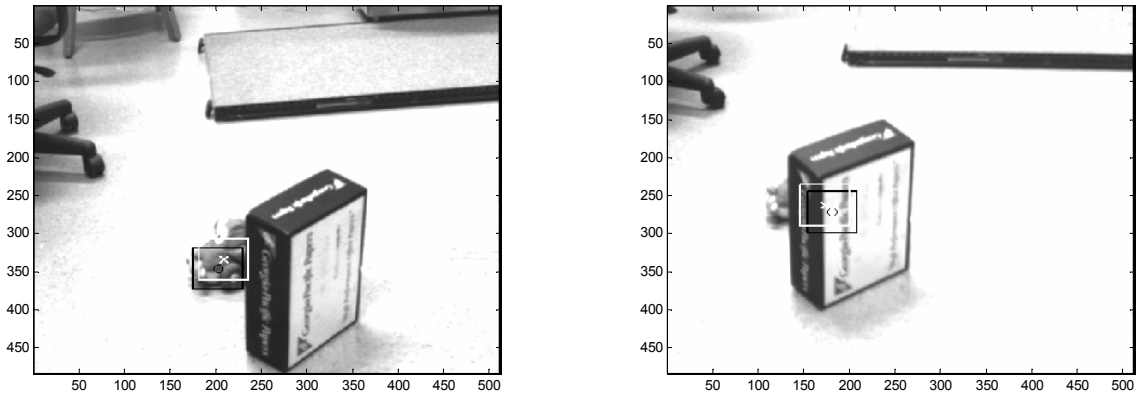


(d)

Figure 7.12, continued. (c) Image pair 6. (d) Image pair 7.

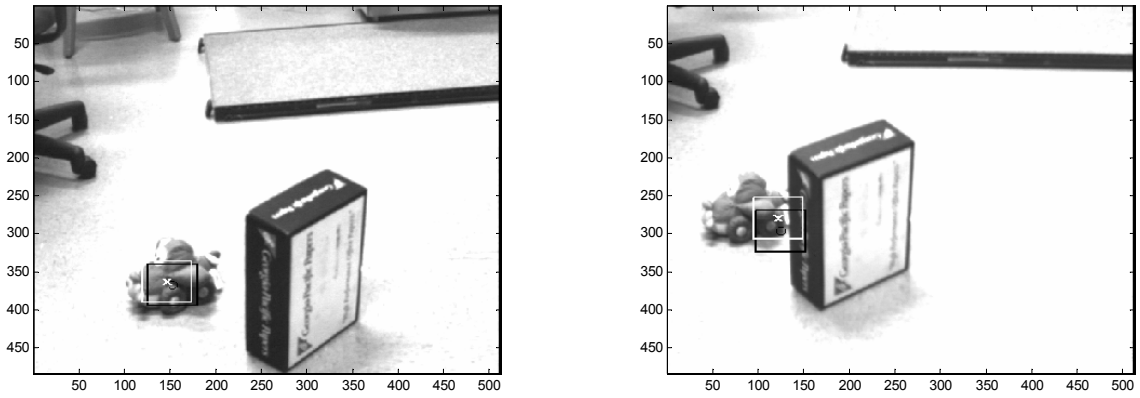


(e)

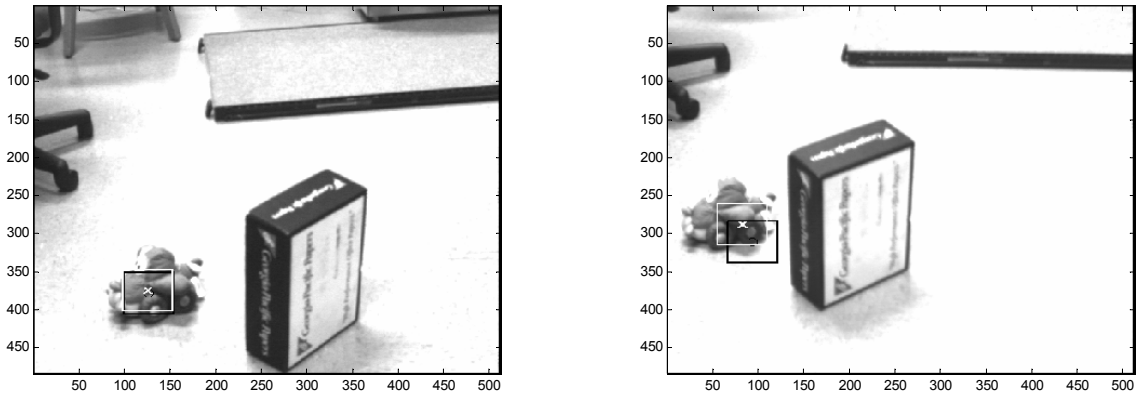


(f)

Figure 7.12, continued. (e) Image pair 8. (f) Image pair 12.



(g)



(h)

Figure 7.12, continued. (g) Image pair 14. (h) Image pair 15.

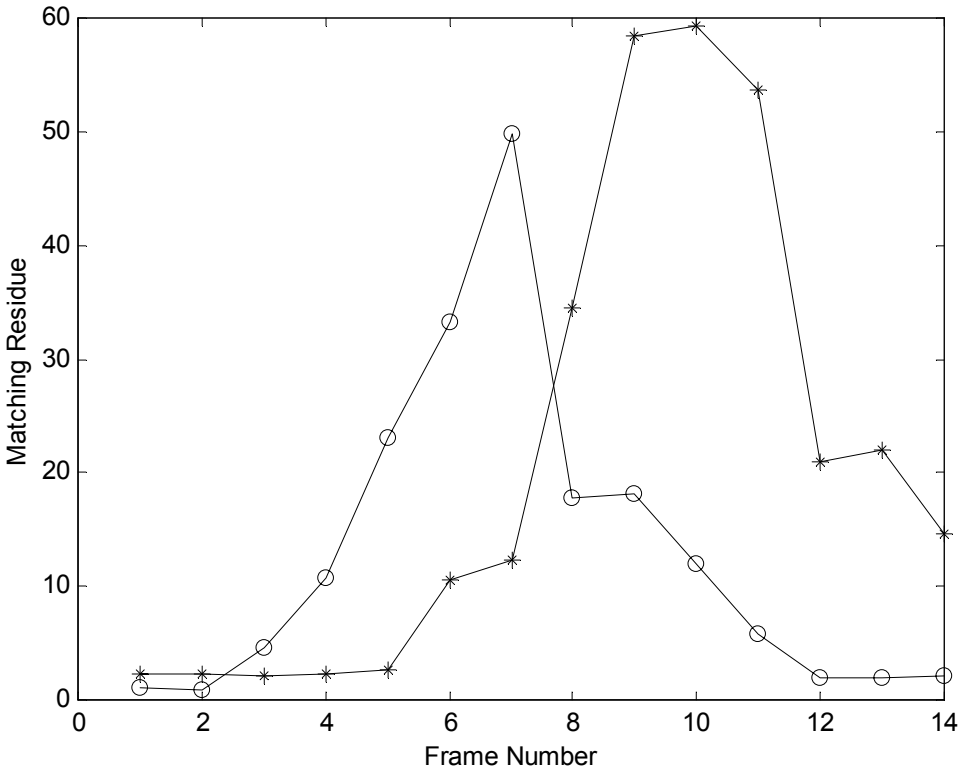


Figure 7.13. Matching residues for the tracked target in Figure 7.12. Each "o" denotes the left to left matching residue for the tracked target. Each "∗" denotes the left to right matching residue for the tracked target.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The primary goal of motion tracking is to keep an object of interest, generally known as a visual target, in the view of the observer at all times. In this dissertation, we presented a novel tracking approach that integrates Kalman-based prediction with fast, area-based search for correspondences. The system described here differs from most earlier approaches in that 3D structure is not considered. Area-based matching is performed using a gradient-descent technique that optimizes 2D affine transformation parameters. In 1994, Shi and Tomasi [Shi94] published a technique for seeking area-based image correspondences using a Newton-Raphson style search for 2D affine deformation parameters. They demonstrated its effectiveness for monocular tracking, in which only small interframe motion is present. We have extended it to accommodate binocular image sequences, as for successive images in a monocular sequence. This search method requires a good initial estimate of the match location, and our technique provides that by using a Kalman filter.

The tracking approach described here can track a moving object in an image sequence where the camera is passive, and when the camera is actively controlled. The algorithm used here is computationally cheap and suitable for real-time implementation. We implemented the proposed method on an active vision system, and carried out experiments of monocular and binocular tracking for various kinds of objects in different environments. These experiments demonstrated that very good performance using real images for fairly complicated situations.

Image matching between two images is a simple one to one mapping, if there is no occlusion. In the presence of occlusion wrong matching is inevitable. This dissertation considers the effect of occlusion on tracking a moving object for both monocular and binocular image sequences. The visual tracking system described here attempts to detect occlusion based on the residual error computed by the matching method. If the residual

matching error exceeds a user-defined threshold, this means that the tracked object may be occluded by another object. When occlusion is detected, tracking continues with the predicted locations based on Kalman filtering. This serves as a predictor of the target position until it reemerges from the occlusion again. Although the method uses a constant image velocity Kalman filtering, it has been shown to function reasonably well in a non-constant velocity situation. Experimental results show that tracking can be maintained during periods of substantial occlusion.

For area-based image matching applications, the choice of window size can have a profound effect on the results that are obtained. This dissertation has also presented a novel approach to the automatic selection of window size, based on the use of moment invariants. We have examined a particular moment invariant that is insensitive to changes in (spatial) scale, translation, rotation, and (intensity) scale, and we have developed a technique for choosing window size based on this invariant. This work is the first, to our knowledge, to suggest the use of moment invariants for window-size selection.

This dissertation has also presented an analysis of quantization-induced error for several important cases of moment invariants. Analytical expressions and experimental results were computed for rectangular images undergoing changes in scale, rotation, and skew. As expected, the error due to quantization decreases as the image size increases relative to pixel size. The quantization error of affine moment invariants tends to be much smaller than Hu moment invariants for square images undergoing rotation. In general, quantization errors are not necessarily periodic with respect to rotation or skew changes. This work is the first to consider quantization error for affine moment invariants, and represents a significant extension of previous work for the case of Hu moment invariants.

The system we have described in this dissertation can assist other vision tasks, such as object recognition, by always keeping the object of interest in view for studying. The motion path of the object being tracked can easily be recorded, and may allow a motion path based object recognition system to be developed.

8.2 Future Work

This section investigates some directions that these extensions could take, as well as novel situations that could benefit from this work.

The quantization error analysis will aid future work by quantifying the extent to which spatial quantization impacts the moment-invariant calculations. Such analysis provides new insights into the utility of these invariants when used as features for recognition, image reconstruction, and matching.

Some operations described in this dissertation could benefit from hardware acceleration. The matching process described in Chapter 4 is quite computationally expensive, but can be done in hardware in real time. In this dissertation, all operations were performed sequentially on a Sun Sparc station 20 , resulting in a processing rate of approximately 5 seconds per frame.

The tracking system described in this dissertation is suited for low to medium moving object speeds. A future extension would be to integrate this algorithm to handle higher velocities for pursuit.

An object moving away from the camera produces a contracting image, meanwhile an object moving towards a camera produces an expanding image. Zooming while tracking compensates this expansion or contraction through focal length adjustments. A future extension would be to experiment with the third degree of freedom of the pan-tilt unit, zooming.

In this research, a very basic calibration technique was developed for the camera positioning sub-system. A future extension would be to develop an automatic camera calibration technique.

There are many applications for the visual tracking system presented in this dissertation. Its ability to keep the moving object near the center of an image frame is beneficial in several fields. The tracking system using active camera has advantages in indoor or outdoor surveillance, in automatic video recording and video teleconferencing, and in manufacturing environment.

References

[Abbo92] A. L. Abbott, "A Survey of Selective Fixation Control for Machine Vision," *IEEE Control Systems Magazine,* pp. 25-31, Aug. 1992.

[Abbo95] A. L. Abbott and B. Zheng, "Active Fixation using Attentional Shifts, Affine Resampling, and Multiresolution Search," *Proceedings: Fifth International Conference on Computer Vision*, pp. 1002-1008, June 1995.

[Ahuj93] N. Ahuja and A. L. Abbott, "Active Stereo: Integrating Disparity, Vergence, Focus, Aperture, and Calibration for Surface Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no.10, pp. 1007-1029, Oct. 1993.

[Aloi91] Y. Aloimonos and D. Tsakiris, "On the Mathematics of Visual Tracking," *Image and Vision Computing*, vol. 9, no. 4, pp. 235-251, 1991.

[Ayac89] N. Ayache and O. Faugeras, "Maintaining Representations of the Environment of a Mobile Robot," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 6, pp. 804-819, 1989.

[Bade97] J. Badenas, M. Bober and F. Pla, "Motion and Intensity-based Segmentation and its Application to Traffic Monitoring," *Proceedings: International Conference on Image Analysis and Processing*, vol. 1, pp. 502-509, 1997.

[Barr95] J. L. Barron and R. Eagleson, "Binocular Estimation of Motion and Structure from Long Sequences Using Optical Flow without Correspondence," *Proceedings: International Conference on Image Processing*, vol. 2, pp. 193-196, Oct. 1995.

[Basc95] B. Bascle and R. Deriche, "Region Tracking through Image Sequences," *Proceedings: 5th International Conference on Computer Vision*, pp. 302-307, 1995.

[Bhat97] D. Bhattacharya and S. Sinha, "Invariance of Stereo Images via the Theory of Complex Moments," *Pattern Recognition*, vol. 30, no. 9, pp. 1373-1386, 1997.

[Blak92] Andrew Blake and Alan Yuille, “Active vision,” *Cambridge, Mass.: MIT Press*, 1992.

[Boyk98] Yuri Boykov, Olga Veksler, and Ramin Zabih, “A Variable Window Approach to Early Vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1283-1294, December 1998.

[Brad94] K. J. Bradshaw, P.F. McLauchlan, I.D. Reid, and D.W. Murray, “Saccade and Pursuit on an Active Head-eye Platform,” *Image and Vision Computing*, vol. 12, pp. 155-163, 1994.

[Bray90] A. J. Bray, “Tracking Objects using Image Disparities,” *Image Vision Computing*, vol. 8, no. 1, pp 4-9, 1990.

[Broi90] T. J. Broida, S. Chandrashekhar, and R. Chellappa, “Recursive 3-D Motion Estimation from Monocular Image Sequence,” *IEEE Transactions on Airospace and Electronic Systems*, vol. 26, no. 4, July 1990.

[Brun94] K. Brunnström, J.-O. Eklundh and T. Uhlin, “Active Fixation for Scene Exploration,” *International Journal of Computer Vision*, 1994.

[Cai95] Q. Cai, A. Mitiche, and J. K. Aggarwal, “Tracking Human Motion in an Indoor Environment,” *Proceedings International Conference on Image Processing*, vol.1, pp. 215-18, Washington, DC, USA; 23-26 Oct. 1995

[Clar88] J. J. Clark and N. J. Ferrier, "Modal Control of an Attentive Vision System," *Proceedings: 2nd International Conference on Computer Vision*, pp. 514-523, Dec. 1988.

[Coch92] S.D. Cochran and G. Medioni, “3D Surface Description from Binocular Stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14, pp. 981-994, 1992

[Coom92] D. Coombs and C. Brown, “Real-time Smooth Pursuit Tracking for a Moving Binocular Robot,” *Proceedings: Conference on Computer Vision and Pattern Recognition*, pp. 23-38, June 1992.

[Cox96] I. J. Cox and S. L. Higorani, “An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual

Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138-150, Feb. 1996.

[Cube99] Datacube Inc. Home Page, "http://www.datacube.com/," 1999.

[Dell97] F. Dellaert and C. Thorpe, "Robust Car Tracking using Kalman filtering and Bayesian templates," *Proceedings: SPIE, The International Society for optical Engineering*, vol. 3207, pp. 72-83, Pittsburgh, PA, USA, 15-17 Oct, 1997.

[Denz97] J. Denzler, H. Niemann, "Real-time Pedestrian Tracking in Natural Scenes," *Computer Analysis of Images and Patterns. 7th International Conference, CAIP '97 Proceedings*, pp. 42-49, Kiel, Germany; 10-12 Sept. 1997

[Duda77] S. A. Dudani, K. J. Breeding and R. B. McGhee, "Aircraft Identification by Moment Invariants," *IEEE Transactions on Computers*, vol. C-26, no. 1, pp. 39-46, Jan. 1977.

[Dufa95] F. Dufaux and F. Moscheni, "Motion Estimation Techniques for Digital TV," *Proceeding of the IEEE*, vol. 83, no.6, pp. 858-876, 1995

[Fair95] S. M. Fairly, I. D. Reid, and D. W. Murray, "Transfer of Fixation for an Active Stereo Platform Via Affine Structure Recovery," *Proceedings: Fifth International Conference on Computer Vision*, pp. 1100-1105, Cambridge, MA, June 1995.

[Falk95] L. Falkenhagen,"3D Object-Based Depth Estimation from Stereoscopic Image Sequences", *International Workshop on stereoscopic and three-dimensional imaging*, September 6-8 1995, Fera Congress Center, Santorini, Greece, 1995.

[Faug93] Olivier Faugeras, "Three-Dimensional Computer Vision–Ageometric Viewpoint," *MIT Press*, Cambridge, MA, 1993.

[Flus93] J. Flusser and T. Suk, "Pattern Recognition by Affine Moment Invariants," *Pattern Recognition*, vol. 26, no. 1, pp. 167-174, 1993.

[Flus94a] J. Flusser and T. Suk, "A Moment-based Approach to Registration of Images with Affine Geometric Distortion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 2, pp. 382-387, March 1994.

[Flus94b] J. Flusser and T. Suk, “Affine Moment Invariants: A New Tool for Character Recognition,” *Pattern Recognition Letters*, vol. 15, pp. 433-436, 1994.

[Frau90] Juan Frau, V. Liario, and G. Oliver, “Polynomial Regression Analysis for Estimating Motion from Image Sequences,” Proceedings: *The International Society for Mobile Robotics (SPIE),* vol. 1388, pp. 329-340, 1990.

[Fuh93] C. S. Fuh, P. Maragos, L. Vincent, “Visual Motion Correspondence by Region-based Approaches,” *Proceedings: Asian Conference on Computer Vision*, pp. 784-789, 1993.

[Fusi97] A. Fusiello, V. Roberto, and E. Trucco, “Experiments with a New Area-based Stereo Algorithm,” *Proceedings: International Conference on Image Analysis and Processing,* vol. I, pp. 669-676, Florence, Italy, Sept. 17-19, 1997.

[Geig92] D. Geiger, B. Ladendorf and A. Yuille, "Occlusions and Binocular Stereo." In G. Sandini (ed.): *Proceedings of the 3rd European Conference on Computer Vision*, Springer Verlag, Berlin, Heidelberg, New York, pp. 425-433, 1992.

[Genn82] D. B. Gennery, “Tracking known Three-dimensional Objects,” *Proceedings: AAAI 2nd National Conference On AI,* Pittsburgh, pp 13-17, 1982.

[Gj98] Castro GJ, Nieto J, Gallego LM, Pastor L, Cabello E., "An Effective Camera Calibration Method," *Proceedings*: *AMC'98 5th International Workshop on Advanced Motion Control*, pp.171-4. Piscataway, NJ, USA.

[Gupt87] L. Gupta and M. D. Srinath, “Contour Sequence Moments for the Classification of Closed Planar Shapes,” *Pattern Recognition,* vol. 20, no. 3, pp. 267-272, 1987.

[Hobr95] Hobrough G. L., “Automatic Stereo Plotting,” *Photogrammetric Engineering & Remote Sensing* (*PE&RS),* vol. 25, no. 5, pp. 763-769, 1995.

[Hoff89] W. Hoff and N. Ahuja, "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, February 1989.

[Homa95] A. S. Homainejad and M. R. Shortis, "A stereo Vision System for Tracking a Dynamic Object," *Proceedings of International Society of Optical Engineering (SPIE),* vol. 2958, pp. 264-271, 1995.

[Horn81] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, pp. 185-203, 1981.

[Hu62] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Transactions on Information Theory,* vol. IT-8, pp. 179-187, Feb. 1962.

[Hube95] E. Huber and D. Kortencamp, "Using Stereo Vision to Pursue Moving agents with a Mobile Robot," *Proceedings: IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2340-2346, Nagoya, Japan, 21-27 May, 1995.

[Hube81] Peter. J. Huber, "*Robust Statistics*," *volume IX of Wiley,* New York, 1981.

[Hung95] Y.-P. Hung, C.-Y. Tang, S.-W. Shih, Z. Chen, W.-S. Lin, "A 3D Predictive Visual Tracker for Tracking Multiple Moving Objects with a Stereo Vision System," *Proceedings of 3rd International Computer Science Conference Image Analysis Applications and Computer Graphics (ICSC95),* pp. 25-32, Hong Kong; 11-13 Dec. 1995.

[Hutt94] D. P. Huttenlocher and E. W. Jaquith, "Detecting Moving Objects with a Moving Camera by Comparing Edge Contours," TR94-*1405,* Institution Cornell University, Computer Science, 1994.

[Ibra93] M. Ibrahim Sezan, Reginald L. Lagendijk, "*Motion analysis and Image Sequence Processing*," Boston, Kluwer Academic Publishers, 1993.

[Inou92] H. Inoue, T. Tachikawa and M. Inaba, "Robot Vision System with a Correlation Chip for Real-time Tracking, Optical Flow and Depth Map Generation," *Proceedings: IEEE International Conference on Robotics and Automation*, pp. 1621-1626, 1992.

[Jain95] Ramesh Jain, Rangachar Kasturi, "*Machine Vision*," 1995.

[Jang97] D.-S. Jang, G.-Y. Kim and H.-L. Choi, "Model-Based Tracking of Moving Object," *Pattern Recognition*, vol. 30, no. 6, pp. 999-1008, 1997.

[Jone92] D. Jones and J. Malik, "A Computational framework for determining stereo correspondence from a set of linear spatial filters," *European Conference on Computer Vision*, Santa Margherita Ligure, Italy, pp. 395-410, 1992.

[Kalm60] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME-Journal of Basic Engineering*, vol. 82, pp. 35-45, March 1960.

[Kam93] J. W. Y. Kam, "A Real-time 3D Motion Tracking System," *M. S. Technical Report 93-16*, Laboratory for Computational Intelligence, Department of Computer Science, The University of British Columbia, April 1993.

[Kana94] T. Kanade, and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920-932, Sept. 1994.

[Koos93] A. Kooschan, "What is New in Computational Stereo Since 1989: A Survey on Current Stereo Papers," TR 93-22, University, Aug. 1993.

[Krot89] E. P. Krotkov, "*Active Computer Vision by Cooperative Focus and Stereo*," Springer-Verlag, 1989.

[Lan95] Z.-D. Lan and R. Mohr, "Robust Matching by Partial Correlation," *INRIA Research Report* no. 2643, 1995.

[Lan97] Z.-D. Lan and R. Mohr, "Robust Location Based Partial Correlation," *INRIA Research Report* no. 3186, 1997.

[Lee91] S. Lee and Y. Kay, "A Kalman Filter Approach for Accurate 3-D Motion Estimation from Sequences of Stereo Images," *CVGIP: Image Understanding*, vol. 54, no. 2, pp. 244-258, Sept. 1991.

[Lee95] J. W. Lee, M. S. Kim, and I. S. Kweon, "A Kalman Filter Based visual Tracking Algorithm for an object Moving in 3D," *International Conference on Intelligent Robots and systems*, pp. 342-347, 1995.

[Levi73] M.D. Levine, D. A. O'Handley and G. M. Yagi, "Computer Determination of Depth Maps," *Computer Graphics and Image Processing*, vol. 2, pp.131-150, 1973.

[Li93] H. Li, P. Roivainen, and R. Forchheimer, “3-D Motion Estimation in Model-based Facial Image Coding,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 545-555, 1993.

[Liao96] S. X. Liao and M. Pawlak, “On Image Analysis by Moments,” *IEEE Transactions on Pattern ansalysis and Machine Intelligence*, vol. 18, no. 3, pp. 254-266, March 1996.

[Lin96] Weiqing Lin, “Multiresolution Stereo Matching Using Two-Dimensional Affine Warping,” *M. S. Research Report*, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, April 1996.

[Lott93] Jean-Luc Lotti, “Adaptive Window Algorithm for Aerial Image Stereo,” *INRIA Research Report* no. 2121, 1993.

[Lott94] J.-L. Lotti, and G. Giraudon, “Correlation Algorithm with Adaptive Window for Aerial Image in Stereo Vision,” *Proceedings of International Society of Optical Engineering (SPIE)*, vol. 2315, pp. 76-87, 1994

[Lync89] P.M. Lynch and R. Vangal, "Tracking Partially occluded Two Dimensional Shapes," *Proceedings of International Society of Optical Engineering (SPIE)*, vol. 1193, *Intelligent Robots and Computer Vision*, pp. 303-314, 1989.

[Mae96] Y. Mae, Y. Shirai, J. Miura, and Y. Kuno, “Object Tracking in Cluttered Background Based on Optical Flow and Edges,” *Proceedings: 13th International Conference on Pattern Recognition*, vol.1, pp.196-200, (1996).

[Mait79] S. Maitra, “Moment Invariants,” *Proceedings of IEEE*, vol. 67, no. 4, pp. 697-699, April 1979.

[Maki93] A. Maki, T. Uhlin and J.-O. Eklundh, “Phase-Based Disparity Estimation in Binocular Tracking,” *Proceedings: 8th Scandinavian Conference on Image Analysis*, pp. 1145—1152, May 1993.

[Mank97] G. S. Manku, P. Jain, A. Aggarwal, L. Kumar, and S. Banerjee, “Object Tracking Using Affine Structure for Point Correspondences,” *Proceedings: Conference on Computer Vision and Pattern Recognition*, pp. 704-709, San Juan, Puerto Rico, June 1997.

[Mata94] M. A. Matar, M. M. Kouta, M. H. Assal, and G. I. Salama, “Moment-based Object Identification and Tracking,” *Proceedings: Second International Conference on Artificial Intelligence Applications*, Cairo, Egypt, pp. 284-294, Jan. 1994.

[Mayb79] Peters S. Maybeck, "*Stochastic Models Estimation, and Control*," Vol. 1, Academic Press, New York, 1979.

[Mena97] C. Menard and W. G. Kropatsch, “Adaptive Stereo Matching in Correlation Scale-space,” *Proceedings: International Conference on Image Analysis and Processing*, vol. I, pp. 677-684, Florence, Italy, Sept. 17-19, 1997.

[Miti94] A. Mitiche, “*Computational Analysis of Visual Motion*,” Plenum Press, New York and London, 1994.

[Mont94] D. A. Montera, S.K. Rogers, D.W. Ruck, and M. E. Oxley, “Object Tracking through Adaptive Correlation,” *Optical Engineering*, vol. 33, no. 1, pp. 294-302, Jan. 1994.

[Murr94] D. Murray and A. Basu, “Motion Tracking with an Active Camera,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, o. 5, pp. 449-459, May 1994.

[Okad96] R. Okada, Y. Shirai and J. Miura, ``Object Tracking Based on Optical Flow and Depth", *Proceedings: IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 565-571, 1996.

[Okut92] M. Okutomi and T. Kanade, “A Locally Adaptive Window for Signal Matching,” *International Journal of Computer Vision*, vol. 7, no.2, pp.143-162, 1992.

[Olso89] T. J. Olson and R. D. Potter, "Real-time Vergence Control," *Proceedings: IEEE Conference on Computer Vision and Pattern Recognition*, pp. 404-409, 1989.

[Olso91] T. J. Olson and D. J. Coombs, "Real-time Vergence Control for Binocular Robots," *International Journal of Computer Vision*, pp. 67-89, 1991.

[Pan94] J. N. Pan, Y. Q. Shi, and C. Q. Shu, “A Kalman Filter in Motion Analysis from Stereo Image Sequences,” *Proceedings: First International Conference On Image Processing*, vol. 3, pp. 63-67, Austin, TX, November 1994.

[Papa93] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, “Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision,” *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 14-35, February 1993.

[Perc95] Directed Perception, Inc., “*Pan-Tilt Unit (Model PTU),*” User’s Manual, 1.07b edition, June1995.

[Perc99] Directed Perception, Inc. Home Page, “http://www.directedperception.com/," 1999.

[Pilu97] Maurizio Pilu, Filton Road, and Stoke Gifford, “A Direct Method for Stereo Correspondence Based on Singular Value Decomposition,” *Proceedings: Conference on Computer Vision and Pattern Recognition*, pp. 261-266, June 17-19, 1997.

[Prok92] R. J. Prokop and A. P. Reeves, “A Survey of Moment-based Techniques for Unoccluded Object Representation and Recognition,” *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, vol. 54, no. 5, pp. 438-460, September 1992.

[Puln99] Pulnix Inc. Home Page, "http://www.pulnix.com/," 1999.

[Reid93] I. D. Reid and D.W. Murray, “Tracking Foveated Corner Clusters using Affine Structure,” *Proceedings: Fourth International Conference on Computer Vision*, pp. 76-83, Berlin, Germany, May 1993.

[Reid95] S. M. Fairley, I. D. Reid, and D. W. Murray, "Transfer of Fixation for an Active Stereo Platform via Affine Structure Recovery," *Proceedings: 5th International Conference on Computer Vision*, pp. 1100-1105, June 1995.

[Reid96] I. D. Reid and D.W. Murray, “Active Tracking of Foveated Feature Clusters using Affine Structure," International *Journal on Computer Vision*, vol. *18*, no. 1, pp. 41-60, 1996.

[Rema94] P. Remanino, P. Brand, and R. Mohr, “Correlation Techniques in Adaptive Template Matching with Uncalibrated Cameras,” *Proceedings of International Society of Optical Engineering (SPIE)*, vol. 2356, Vision Geometry III, pp. 252-263, 1994.

[R99] Wagner R, Feiyu Liu, and Donner K., "Robust motion estimation for calibrated cameras from monocular image sequences, "*Computer Vision & Image Understanding*," vol.73, no.2, pp.258-68, Feb. 1999.

[Roac79] J. W. Roach, and J. K. Aggarwal, “Computer Tracking of Objects Moving in Space,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI, PAMI-1, 2, 1979.

[Roac80] J. W. Roach, and J. K. Aggarwal, ”Determining the Movement of Objects from a Sequence of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, pp. 554-562, 1980.

[Rous87] P.J. Rousseeuw and A.M. Leroy, “*Robust Regression and Outlier Detection*,” volume XIV of Wiley, J. Wiley and Sans, New York, 1987

[Sala98a] G. I. Salama and A. L. Abbott, “Moment Invariants and Quantization Effects” *Proceedings: Conference on Computer Vision and Pattern Recognition*, pp. 157-163, Santa Barbara, CA, 1998.

[Sala98b] G. I. Salama and A. L. Abbott, “Window-size Selection using Moment Invariants,” *Proceedings: Conference on Computer Vision, Pattern Recognition, and Image Processing*, pp. 374-377, Research Triangle Park, NC, 1998.

[Sala98c] G. I. Salama and A. L. Abbott, “Monocular and Binocular Tracking,” *Proceedings: Conference on Computer Vision, Pattern Recognition, and Image Processing*, pp. 374-377, Research Triangle Park, NC, 1998.

[Sche98] S. Scherer, W. Andexer, and A. Pinz, "Robust Adaptive Window Matching by Homogeneity Constraint and Integration of Descriptions," *Proceedings: 14th International Conference on Pattern Recognition*, pp. 777-779, Brisbane, Australia, 1998.

[Schm95] C. Schmid and R. Mohr, “Matching by Local Invariants,” *RR-2644*, INRIA, August 1995.

[Shi94] J. Shi and C. Tomasi, “Good Features to Track,” *Proceedings: IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 593-600, June 1994.

[Shie92] J.-Y. Shieh, H. Zhuang, and R. Sudhakar, “A Direct Method of Motion Estimation from a Sequence of Stereo Images,” *IEEE International Conference on Systems, Man and Cybernetics,* vol. 2, pp. 1229-1234, Chicago, IL, USA; 18-21 Oct.1992.

[Soat94] Stefano Soato and Pietro Perona, “Recursive Estimation of Camera Motion from Uncalibrated Image Sequences,” *Technical ReportCIT-CDS 94-005*, California Institute of Technology, 1993.

[Sung97] S.-H. Sung, S.-I. Chien, M.-G. Kim, and J.-N. Kim, “Adaptive Window Algorithm with Four-direction Sizing Factors for Robust Correlation-based Tracking,” *Proceedings: Ninth IEEE International Conference on Tools with Artificial Intelligence*, pp. 208-215, Newport Beach, CA, USA; 3-8 Nov. 1997.

[Tayl94] J. Taylor, T. Olson, and W. N. Martin, "Accurate Vergence Control in Complex Scenes," *Proceedings: Conference on Computer Vision and Pattern Recognition*, pp. 540-545, June 1994.

[Teh86] C.-H. Teh and R. T. Chin, “On Digital Approximation of Moment Invariants,” *Computer Vision, Graphics, and Image Processing,* vol. 33, pp. 318-326, 1986.

[Teka95] A.M. Tekale, “Digital Video Processing,” Prentic-Hall, 1995.

[Toma92] C. Tomasi and T. Kanade, “Shape and Motion from Image Streams under Orthography: A Factorization Method,” *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.

[Tsai89] R. Y. Tsai and R. K. Lenz, “A new Technique for fully Autonomous and Efficient Robotic Hand/Eye Calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 3, pp. 345-358, June. 1989.

[Tsui97] H. T. Tsui, Z. Y. Zhang, and S. H. Kong, “Feature Tracking from an Image Sequence using Geometric Invariants,” *Proceedings: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.244-9, 1997, Los Alamitos, CA, USA.

[Tsuy98] Yamane Tsuyashi, Shirai Y, Miura J., “Person Tracking by Integrating Optical Flow and Uniform Brightness Regions**,”** *Proceedings: 1998 IEEE International Conference on Robotics and Automation). IEEE.* Part vol.4, pp.3267-72, 1998, New York, *NY, USA.*

[Visi99] Computer Vision Home Page, Test Images, " http://www.cs.cmu.edu/~cil/vision.html".

[Wang95] H. Wang, W.L. Goh, C.S. Chua, and C.T. Sim, “Real-Time Object Tracking,” *Proceedings: 21st International Conference on Industrial Electronics, Control, and Instrumentation (IECON)*, vol. 2, pp.1366-71, 1995, New York, NY, USA.

[Wang98] H. Wang, C.S. Chua, and C.T. Sim, “Real-Time Object Tracking from Corners,” *Robotica*, vol.16, pt.1, Jan.-Feb. 1998, pp.109-16. Publisher: Cambridge University Press, UK.

[Waxm86] A. M. Waxman and J. H. Duncan, “Binocular Image Flows: Steps Toward Stereo-Motion Fusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, November 1986.

[Welc97] Greg Welch and Gary Bishop, "An Introduction to Kalman Filter," http://www.cs.unc.edu/~welch/media/pdf/kalman.pdf, September 17, 1997.

[Weng92] J. Weng, N. Ahuja, and T. S. Huang, “Matching Two perspective Views,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14, pp. 806-825, 1992.

[Wilc87] B. Wilcox, D. B. Gennery, B. Bon, and T. Litwin “Real-Time Model-based Vision System for Object Acquisition and Tracking,” *Proceedings of International Society of Optical Engineering (SPIE)*, vol 754, 1987.

[Xu87] G. Xu, S. Tsuji and M. Asada, “A Motion Stereo Method Based on Coarse-to-Fine Control Strategy,”*IEEE Transctions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 2, March 1987.

[Yach81] M. Yachida, M. Asada and S. Tsuji, “Automatic Analysis of Moving Image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 1, pp. 12-20, 1981.

[Yama95] S. Yamamoto, Y. Mae, Y. Shirai and J. Miura, ``Realtime Multiple Object Tracking Based on Optical Flows," *Proceedings: IEEE International Conference on Robotics and Automation,* vol. 3, pp. 2328-2333, 1995.

[Yi95] J.-W. Yi, T. S. Yang., and J.-H. Oh, “Estimation of Depth and 3D Motion Parameters of Moving Objects with Multiple Stereo Images by Using Kalman Filter,” *Proceedings: IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 2, pp.1225-30, 1995, New York, NY, USA.

[Zhan92] Z. Zhang and O. Faugeras, “Three-Dimensional Motion Computation and Object Segmentation in a Long Sequence of Stereo Frames,” *International Journal on Computer Vision*, vol. 7, no. 3, pp. 211-241, 1992.

[Zhen94] B. Zheng, “Multiresolution Fixation of Binocular Vision System,” *M.S. Thesis*, Virginia Polytechnic Institute and State University, December 1994.

Appendix A

Two-dimensional Affine Transformation

A transformation that preserves lines and parallelism (maps parallel lines to parallel lines) is an affine transformation. A 2D affine mapping establishes a one-to-one relationship between a point (*x, y*) and a transformed point ($\hat{x}$, $\hat{y}$) as follows:

$$\hat{x}=d_{xx}x+d_{xy}y+d_x \quad \text{(A.1)}$$

$$\hat{y}=d_{yx}x+d_{yy}y+d_y. \quad \text{(A.2)}$$

The constants d_x and d_y represent a displacement, and constants d_{xx}, d_{xy}, d_{yx}, and d_{yy} represent the scale, rotation, and skew of the mapping between the two points (*x, y*) and ($\hat{x}$, $\hat{y}$). Equations (A.1, A.2) can be rewritten in matrix form as follows:

$$\hat{X}=DX+d \quad \text{(A.3)}$$

Where

$$\hat{X}=\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix},\ X=\begin{bmatrix}x\\ y\end{bmatrix}, D=\begin{bmatrix}d_{xx} & d_{xy}\\ d_{yx} & d_{yy}\end{bmatrix}, \text{ and } d=\begin{bmatrix}d_x\\ d_y\end{bmatrix}. \quad \text{(A.4)}$$

Figure A.1 illustrates the 2D affine mapping between a stereo image pair corresponding to the same object in three-dimensional space where the difference between the two images results from the perspective difference of two cameras. It can be noticed that if a target point is given in the left image (Figure A.1 (a)), then it is difficult to find a corresponding point on the right image (Figure A.1 (b)). The 2D affine transformation is able to handle problems that result from the perspective differences in image matching [Lin96].

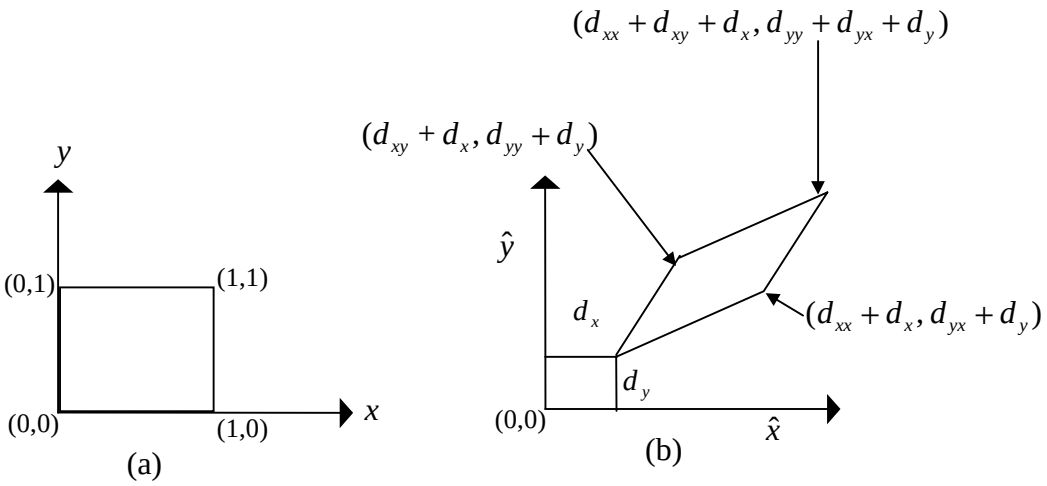


Figure A.1. 2-D affine mapping between a stereo image pair.
(a) Image plane (x, y). (b) Transformed plane ($\hat{x}$, $\hat{y}$)

Appendix B

The Discrete Kalman Filter

The discrete Kalman filter [Kalm60] is a recursive predictive update technique used to determine the correct parameters of a process model. Given some initial estimates, the parameters of a model can be predicted and adjusted with each new measurement, providing an estimate of error at each update. Its ability to incorporate the effects of noise, and its computational structure, has made it popular for use in computer vision tracking applications.

The Kalman filter estimates the state of a linear system modeled by the linear stochastic difference equation

$$X_{k+1} = \varphi_k X_k + BU_k + W_k \tag{B.1}$$

with a measurement

$$Z_k = H_k X_k + V_k \tag{B.2}$$

Where the following definitions are used:

X_k	System state vector at time *k*,
$\hat{X}_k$	Vector containing the current parameters at time *k*,
$\hat{X}_k^-$	Vector containing the current estimate of parameters at time *k*,
$\hat{X}_{k+1}$	Vector containing the parameter estimates at the next time sample,
φ_k	State transition matrix relating X_k to X_{k+1},
W_k	Zero mean white noise, modelling input process noise,
Z_k	Measurement vector at time *k,*
H_k	Matrix giving the noiseless connection between the state vector and measurement vector at time *k,*
V_k	Measurement noise as zero mean white noise,
U_k	The control input vector,
B	Matrix relates the control input to the state X,
K_k	Kalman gain matrix at time k,

Q_k System error covariance matrix ,

R_k Measurement error covariance matrix ,

P_k Matrix containing the error covariance for the current parameters,

P_k^- Matrix containing the error covariance for the estimated parameters,

$\hat{P}_k^-$ Matrix containing the error covariance for the estimated parameters at the next time instance,

Figure B.1 offers a complete picture of the operation of the Kalman filter. The equations of the Kalman filter fall into two groups: the time update equations and measurement update equations [Welc97]. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain a priori estimates for the next time step. So, these also can be thought of as predictor equations.

$$\hat{X}_{k+1}^- = \varphi_k \hat{X}_k \tag{B.3}$$

$$P_{k+1}^- = \varphi_k P_k \varphi_k^T + Q_k \tag{B.4}$$

The measurement update equations are responsible for incorporating a new measurement into a priori estimate to obtain an improved a posteriori estimate. So, these equations can be thought of as correct equations.

The Kalman filter provides a one-step prediction of the state, as follows:

$$\hat{X}_k = \hat{X}_k^- + K_k\left(Z_k - H_k \hat{X}_k^-\right) \tag{B.5}$$

The kalman gain is defined as

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{B.6}$$

The updated error covariance is

$$P_k = \left(I - K_k H_k\right) P_k^- \tag{B.7}$$

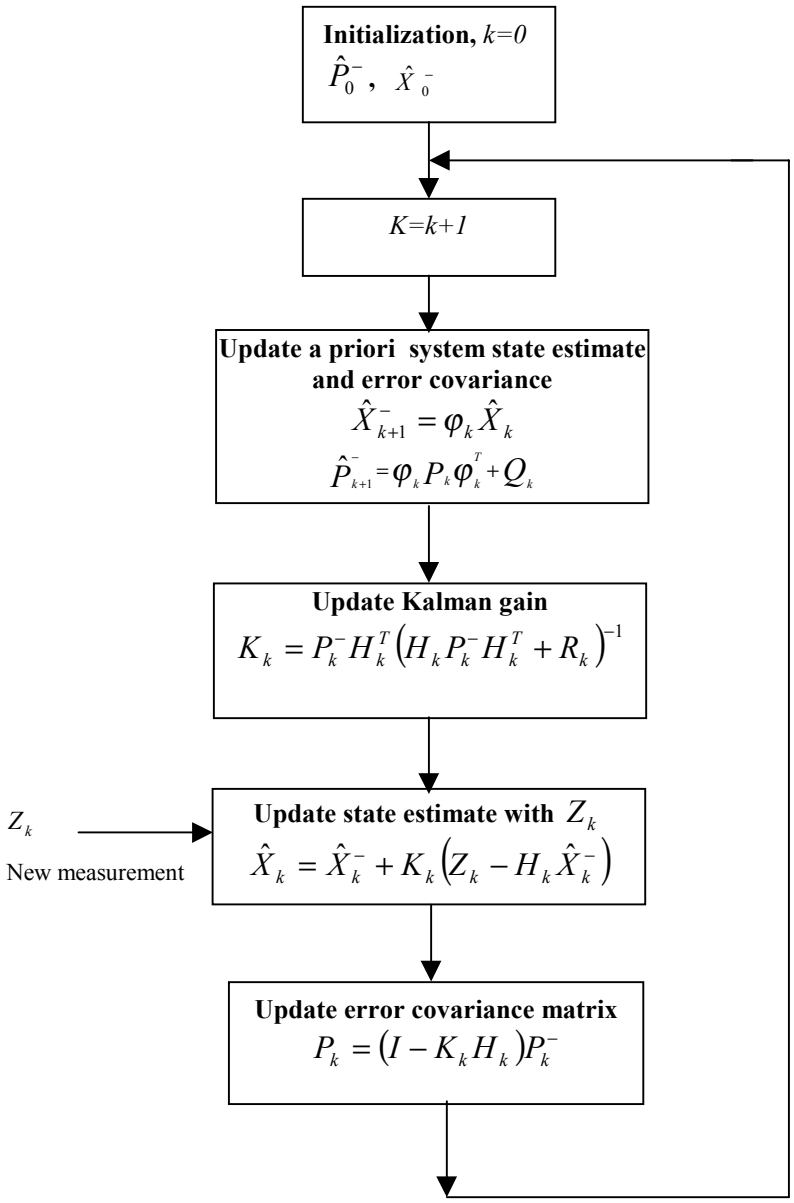


Figure B.1 Kalman filter flowchart.

For example, the matrices for a constant velocity 2D Kalman filter are as follows:

$$X = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}, \quad \varphi = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.8}$$

$$Z=\begin{pmatrix} x \\ y \end{pmatrix}, \qquad H=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \qquad R=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{B.9}$$

Vita

Gouda Salama was born in 1965. He finished his B.Sc., Electrical Engineering from Military Technical College, Cairo, Egypt in 1988. He did his M.S., Computer Engineering from Military Technical College, Cairo, Egypt in 1994. In 1996 he came to U.S.A. on a scholarship awarded by Egyptian Embassy for graduate studies. Currently, he has been pursuing his doctoral degree at the Bradley Department of Electrical engineering at Virginia Polytech. Inst. and State Univ. His Research involves robust algorithms to track monocular and binocular moving objects. His interests are pattern recognition, computer vision, image processing, computer graphics, Artificial intelligence and control system.