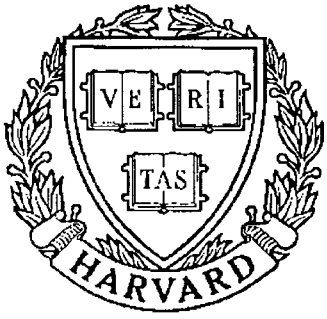
THESIS REPORT
Master's Degree

SYSTEMS
RESEARCH
CENTER



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Visual Tracking Strategies

by: D.P. Tsakiris
Advisor: J. Aloimonos

M.S. 88-8
Formerly TR 88-66

VISUAL TRACKING STRATEGIES

by

Dimitrios P. Tsakiris

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1988

Advisory Committee:

Professor	John Aloimonos
Professor	John S. Baras
Professor	P. S. Krishnaprasad

ABSTRACT

Title of Thesis: Visual Tracking Strategies

Dimitrios P. Tsakiris, Master of Science, 1988

Thesis directed by: John Aloimonos
Assistant Professor
Computer Science Department

Thesis co–supervised by: P. S. Krishnaprasad
Professor
Electrical Engineering Department

Visual tracking is one of the most important applications of Computer Vision and 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 from segmentation and point correspondence problems.

A mathematical theory for visual tracking of a three–dimensional target moving rigidly in 3–D is presented here and it is shown how a monocular observer can track an initially foveated object and keep it stationary in the center of his visual field. Our attempt is to develop correspondence–free tracking schemes and take advantage of the dynamic segmentation capabilities inherent in the optical flow formalism. Moreover, a general tracking criterion, the *Tracking Constraint* is derived, which reduces tracking to an appropriate optimization problem. The connection of our tracking strategies with the Active Vision Paradigm is shown to provide a solution to the Egomotion problem.

In the first part of this work, tracking strategies based on the assumption that we know the optical flow field are examined and tracking is formulated as a constrained optimization and a penalized least–squares problem. In the second part, tracking strategies based on the recovery of the 3–D motion of the target are devised under the assumption that we know the shape of the target. A correspondence–free scheme is derived, which depends on global information about the scene (provided from linear features of the image) in order to bypass the ill–posed problem of computing the spatial derivatives of the

image intensity function and amounts to the solution of a linear system of equations in order to estimate the 3–D motion of the target. An important feature of these tracking strategies is that they do not require continuous segmentation of the image in order to locate the target. Supposing that the target is sufficiently textured , dynamic segmentation using temporal derivatives of the linear features provides sufficient information for the tracking phase. Therefore, this approach is expected to perform best when previous ones perform worst, namely in a complex visual environment.

Experimental results for the algorithms presented here demonstrate their robustness in the presence of noise.

© Copyright by

Dimitrios P. Tsakiris

1988

To my parents

ACKNOWLEDGMENTS

I would like to take this opportunity to express my gratitude to my advisor Dr. Yiannis Aloimonos for introducing me to the exciting realm of Computer Vision, for his guidance through its meanders and for his continuous encouragement and stirring enthusiasm during our collaboration.

I would also like to thank my thesis co-supervisor Dr. P.S.Krishnaprasad for his many valuable contributions to this thesis, through his persistence and exceptional ability to approach problems under various perspectives.

Moreover, I would like to thank Dr. John S. Baras for being a member of my Advisory Committee, for his constructive criticism and for the support of the Systems Research Center to this work.

Finally, Drs. A.Bandopadhay, J.Viniotis, as well as several of my fellow students at the Intelligent Servosystems and the Vision Laboratories, deserve special appreciation for enlightening discussions and assistance during this period.

TABLE OF CONTENTS

Chapter 1: Introduction 1

a. Tracking in the Primate Oculomotor System 2

b. Tracking System Description 4

c. Main results in this thesis 8

Chapter 2: The System Model 12

a. The Image Formation Model 12

b. The System Kinematics 16

c. Computational Theory of Tracking 22

I. Continuous Motion and Optical Flow 22

II. The Active Vision Paradigm for Early Vision 22

III. The Tracking Constraint 25

Chapter 3: Tracking Using Knowledge of Optical Flow 30

a. Recovery of Optical Flow 30

b. Tracking 33

Tracking Schema 3.1 : Constrained Optimization Formulation .. 37

Tracking Schema 3.2 : Generalized Cross-Validation Formulation 38

Chapter 4: Tracking Using Knowledge of Shape 43

a. The Theory of Linear Features 44

b. Recovery of Three-Dimensional Motion 46

Example : The Planar Case 49

b. Tracking 53

Tracking Schema 4.1 : Using Shape Information 54

Example : The Planar Case 55

Tracking Schema 4.2 : Without Using Shape Information 56
d. Simulation Results 59

Chapter 5: Conclusions and Future Research 73

Appendix A 76
Appendix B 80
Appendix C 86

Bibliography 89

LIST OF FIGURES

Figure **Page**

2.1 : The World, Camera and Target Coordinate Systems (General Case) 14

2.2 : The Camera and Target Coordinate Systems 15

2.3 : Inertial and Moving Frames 17

2.4 : The World and Camera Coordinate Systems 21

3.1 : 4 × 4 array of intensities 42

4.1 : Display of simulation results on the IRIS 60

4.2 : Target tracker, target translation, 0% noise 63

4.3 : Target tracker, target rotation, 0% noise 63

4.4 : Target tracker, general motion of the target, 0% noise 64

4.5 : Target tracker, general motion of the target, 30% noise 64

4.6 : Target tracker, general motion of the target, 60% noise 65

4.7 : Target tracker, general motion of the target, 90% noise 65

4.8 : Target tracker, motion along the line of sight, 90% noise 66

4.9 : Target tracker, tracking schema 4.2 67

4.10 : Motion parameter estimator and target tracker, target translation, k=10 70

4.11 : Motion parameter estimator and target tracker, target translation, k=2 70

4.12 : Motion parameter estimator and target tracker, target rotation, k=10 71

4.13 : Motion parameter estimator and target tracker, target rotation, k=2 71

4.14 : Motion parameter estimator and target tracker, general motion of the target, k=5 72
4.15 : Motion parameter estimator and target tracker, motion of the target along the line of sight, k=2 72

CHAPTER ONE

INTRODUCTION

The problem of visual tracking is related to some of the most important applications of Computer Vision and several tracking systems have been developed, which, either examine the tracking of a planar target moving in two dimensions or attempt to reduce the three–dimensional tracking problem to the tracking of a characteristic point of the target on the image–plane using image segmentation and feature correspondence–based algorithms. In the general case, visual tracking is a problem of especial difficulty, since it involves dynamic imagery and scene analysis, which is not yet fully understood. A mathematical theory for visual tracking of a three–dimensional target moving rigidly in 3–D is presented here and it is shown how a monocular observer can track an initially foveated object and keep it stationary in the center of his visual field.

In Ch. 2 a kinematic model of the system is derived and the *Tracking Constraint* is introduced, which reduces the problem of tracking to an appropriate optimization problem. The connection of our tracking strategies with the Active Vision Paradigm is shown to provide a solution to the Egomotion problem.

In Ch. 3, tracking strategies based on the assumption that we know the optical flow field on a subset of the image plane are examined. Various formulations of the tracking problem as an optimization and a penalized least–squares problem are examined and techniques such as Generalized Cross–Validation are applied for its solution.

In Ch. 4, tracking strategies based on the recovery of the 3–D motion of the

target are devised under the assumption that we know the shape of the target. A correspondence–free scheme is derived, which does not depend on measurement of local quantities, but uses global information from the image (linear features) and amounts to the solution of a linear system of equations for the estimation of the 3–D motion. An important characteristic of those tracking strategies is that they do not require segmentation of each new frame of the image in order to locate the target. Supposing that the target is sufficiently textured , dynamic segmentation, using temporal derivatives of linear features of the image, provides sufficient information for the tracking phase.

Experimental results for the algorithms presented here, demonstrate their robustness in the presence of noise.

In the next section (Ch. 1.a) we present an overview of the primate tracking system and attempt to answer the question why we need tracking. Then, in Ch. 1.b we describe previous approaches to the tracking problem and give a general description of a tracking system. In Ch. 1.c we provide the motivation for our approach to the 3–D tracking problem, by attempting to answer the question why we need to estimate the 3–D motion or the optical flow of 3–D targets moving in 3–D and show the advantages of this approach with respect to previous ones.

Ch. 1.a) Tracking in the Primate Oculomotor System

Retinal projections of the 3–D world provide the brain with valuable information about its environment. Due to the loss of information about depth during the image formation process, this information is only a partial description of the world, but most observers are able to reconstruct correctly the 3–D world. Object motion is one of the clues used for this reconstruction, but, since a specific retinal motion may have been induced by an infinite number of 3–D motions, additional information about possible regularities in the world, such as rigidity (Ullman [71], Hoffman and Benett [38]), is necessary in order to obtain a unique reconstruction of the 3–D world structure.

The retina in primates is composed by an array of two kinds of photoreceptors, the rods, which are sensitive to variations of the image intensity, and the cones, which are sensitive both to variations of intensity and to wavelength and allow color perception. Primates have developed an area with high density in cones, the macula lutea, and in its center, the fovea centralis, they have the most acute visual perception of color and

of intensity variation. From there, the most important visual information is transmitted through the optical nerve to the lateral geniculate body and to the cortical visual center. Therefore the target of interest should be moved in the fovea so that the visual perception be optimized. In a visual environment which changes continuously, due either to the motion of objects of the scene or to the motion of the observer himself, reorientation of the eye is necessary in order first to keep a moving target in the field of view and then to foveate it, so that the collected visual information be maximized. Moreover, motion induces blurring of the image formed on the retina and this prevents useful information from being seen. Physiological studies showed that even a relatively slow motion of 3 degrees per second, produces a decrement of resolution of the perceived image analogous to the one caused by myopia (Westheimer and McKee [78]). Therefore, eye movements are again necessary in order to stabilize the target image on the retina.

Four basic types of eye movements can be distinguished :

Saccades, which are fast relocations of the line of sight and are used to foveate an object. The velocity of this eye movement is large enough (in the order of 100 deg./sec) to prevent any useful vision while the eye is moving, but its purpose is to relocate the line of sight and stabilize the target of interest at the fovea as fast as possible and thus maximize the time devoted to the visual processing of a stable and foveated object.

Smooth Pursuit, which is a slow and continuous movement used to keep a target steady on the fovea despite its motion.

Vestibulo-ocular eye movements, which are used to maintain fixation during head movements and allow simultaneous motion and vision. They are driven by a direct reflex, from the vestibular organs in the inner ear which act as stability and equilibrium sensors, to the oculomotor system.

Vergence movements, which converge appropriately the line of sight of the two eyes so that the observer may look to nearby objects.

For target tracking, humans use mostly a sequence of saccades and smooth pursuit movements (Bahill and LaRitz [8]). The target is foveated by a saccadic movement and then smooth pursuit movements attempt to keep it foveated. If it drifts by a significant amount or leaves the field of view, a new saccade is necessary in order to foveate it again. In the case of fast moving targets, e.g. when athletes track fast moving balls, anticipatory saccades may put the eye ahead of the target and let the target catch with

the eye ([8]). Psychophysical experiments showed that there exists a time delay of about 150 ms in the response of the human tracking system. Saccadic motion is again employed to overcome this delay, catch up with the target and then allow smooth pursuit to follow its trajectory (Bahill and McDonald [9]). It is then obvious that saccadic eye movements act as a prediction mechanism for the primate oculomotor system. But it seems that it is the smooth pursuit mechanism that keeps the target foveated and allows the extraction of useful visual information.

Ch. 1.b) Optical Tracking System Description

Tracking systems may employ active sensors like radar, sonar and laser in order to produce range data of a scene, with explicit depth information. The problems in processing range data are very similar to those in processing visual data (Besl and Jain [14]). But they involve scanning devices, which are potentially unreliable and they consume excessive power, which is an important factor especially in space applications (Wilcox, Gennery, Bu and Litwin [80], Kern, Kugel and Hettlage [45]). Thus passive electromagnetic sensing, and vision systems in particular, are an attractive alternative, providing very good spatial and temporal resolution and accuracy (Gilbert et al. [32], Gennery [29]). Moreover, active research in this field over the last 30 years, provided solutions to a number of practical problems.

Applications of visual tracking include aircraft and missile tracking, robot manipulation of objects, navigation, traffic monitoring, cloud tracking in meteorology, cell motion and tracking of moving parts of the body (e.g. heart) in biomedicine.

In the past, the planar motion of 2–D targets (Bouthemy and Benveniste [17], Venetsanopoulos and Cappellini [73]) and 3–D targets (Schalkoff and McVey [64], Rajala, Riddle and Snyder [58], Legters and Young [46], Nagalia [51]) received considerable attention, due to its important applications, such as the motion of parts on conveyor belts in industry and the analysis of satellite pictures of atmospheric disturbances in meteorology.

Here we will consider mainly the more general problem of tracking 3–D targets moving in the 3-dimensional space.

In Roach and Aggarwal [59] tracking of rigid convex polyhedra is considered. For each new frame the objects are specified by segmentation. The centroid of each object

is computed and from its displacement an estimate of the translational 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.

In Gilbert [31] and Gilbert et al. [32], the design of the U.S.Army Videotheodolite is presented. Segmentation of the image is performed at video rate, using a statistical clustering algorithm to separate the target (missile or aircraft) from the background and plume areas. A window around the target is specified, which simplifies subsequent segmentation. The segmented image is transformed into a binary image of the target, which is used to compute the centroid and orientation of the target. These are the input to the tracking processor, which computes the change in azimuth and elevation of the telescope necessary to continue tracking.

Wallace and Mitchell [76] characterize the boundary of the projection of a known moving target (aircraft) by the coefficients of its expansion in a complex Fourier series. After normalization, these are compared to a library of coefficients corresponding to different projections of the target and the target orientation is specified. Tracking involves identification of the target in each new frame, extraction of its boundary and estimation of its orientation using library data.

Cornog [24] describes the MIT Eye–Head robot, as well as control strategies to fixate and track high contrast targets in 3–D. The targets are white spheres on black background and the image is segmented using thresholding, where the threshold varies with the illumination conditions. In the tracking phase, the previous position of the center of the sphere is used as initial point for radial search in order to specify its new position. Its deviation from the line of sight is used as an error signal to the controller that must specify the camera rotation. Error exceeding a threshold induces a saccadic repositioning of the system.

Gennery [28], [29] and Wilcox et al. [80] describe a stereo system developed at JPL for tracking 3–D targets whose models are known. In the acquisition phase, features such as vertices of polyhedra are detected and tracked in 2–D, the stereo matching problem is solved and thus the 3–D location of the features is inferred and matched to those of the object model. During the tracking phase, features are matched directly from images to the object model, without need for stereo matching between pairs of them from the cameras. The object position and orientation is predicted by extrapolation from previous data and is adjusted from new data from feature detection.

The optical tracking systems presented to this moment, are composed of two major subsystems: The Vision and the Control Modules. The first generates a desired trajectory in real-time, that the second has to implement, also in real-time.

The *Vision Module* consists of one or more cameras, frame grabbers and the necessary computer hardware and software for the processing of the visual information. It gets the analog image, digitizes it, creates an array of intensities for all points of the field of view and from this information locates the target and computes the camera positioning that will achieve tracking.

Locating the target is the part of the tracking process referred to as the *Target Acquisition* phase and corresponds in the human oculomotor system to the fixation of a target using saccades. It may encompass problems like recognition of the target and recovery of its approximate location in 3-D and those may require image segmentation, feature extraction, creation of a model of the target and matching this model to one of a database of target models. Very often range data from active sensors ([80]) or structured light ([51]) are used for the acquisition of the target, if its initial position is not preset by a human operator ([64]) or by a database of possible target locations. The more complex the scene, the more sophisticated the algorithms for this phase must be and the more sophisticated they become, the more delay do they introduce in the system.

The *Tracking* proper phase includes the computation and execution of a smooth and continuous camera motion that will achieve tracking by keeping the target foveated and corresponds in the human visual system to the smooth-pursuit of a moving object.

During this phase, the target tends to drift slowly away from the center of the image and, after some time, it may be lost. Then a new acquisition phase is required.

As in many vision applications, we face here the problem of processing the images in real-time, which is defined as the processing of images, which, when viewed on an appropriate display, has an apparent continuity ([73]). A measure of this is the number N of operations that must be performed per second to realize that processing. For an image of dimensions $m \times n$, a television scan rate of l images per second and if r operations per pixel are required, this number is $N = m \times n \times l \times r$. For the case of a 256 x 256 image updated at video rate (30 Hz), we have $N = 2 \times 10^6 r = 2 \times r\ MOPS$ (mega-operations per second). With r in the range of 10^3 to 10^5 operations per pixel, which may be necessary to solve the vision problems mentioned earlier, N becomes impressively large (10^3 to $10^5 MOPS$). Conventional uniprocessor architectures cannot match these

processing requirements, but suitable parallel architectures may be found for the specific processing task.

The Vision Module then, provides a desired trajectory, that has to be implemented in the *Control Module* of the system. The plant in this case is the camera, with the tracking mount and the actuators that drive it in the azimuth and elevation directions and is usually modeled as a second-order system ([24]). From the System Theoretic point of view, the problem of trajectory tracking, in the sense of deriving the control law that will force the plant to follow a desired trajectory, has received considerable attention and several techniques have been examined for its solution. Depending on the technique, the selection of an appropriate control strategy to implement the optimal camera trajectory may be a complicated task. In any case, the hard real-time regime of the problem will impose control cycles ranging from 0.1 to 1 ms, while the updating of the sensory information will be done at video rate (30 to 60 Hz). Bernstein [13] argues that although the computer speed improvement over the last 25 years has been in the order of 10^6, the average control computer speed improvement was only in the order of 10. This is especially important for visual servoing applications, where scene complexity may increase dramatically the processing time requirements and considerable time delays in the controller may be intolerable. A solution seems to lie in the use of a hierarchy of control computers with assembly programmed ones handling Input/Output intensive and real-time critical tasks and with high-level programmed ones handling task synchronization and supervisory control. Such hierarchical schemes were successfully used in the past ([24]).

Inertia of such a mechanical system will prevent instantaneous repositioning of the camera in case the target drifts away from the center of the camera. Therefore, prediction of future positions of the target is necessary, in order for the target to be kept foveated. Moreover, the image processing algorithms in the Vision Module will introduce additional delays, which must be compensated. A physiological analogous of a prediction mechanism does exist in the human oculomotor system (see Ch. 1.a). In Hunt and Sanderson [43], various predictive schemes for visual tracking, based on linear regression and Kalman filtering, are presented. As we saw, one important reason for tracking is to reposition a high resolution, but narrow field of view, photoreceptor array (like the fovea), so that more information about the target become available. Then, accurate prediction of the target's future positions becomes more critical, since, if the target is

lost, the time- consuming acquisition phase must be repeated. In general, due to delays introduced from the image processing and control systems, a trade-off between the velocity of the moving target and the complexity of the visual situation, which will influence our choice of the Vision Algorithms, seems inevitable in a tracking system. The delays are difficult to quantify a priori, since they will depend a lot on the hardware which is available for computations and control.

Finally, the mechanical problems arising in the design of high-performance machines, like the choice of actuators and transmission components in order to achieve the best dynamic performance and many other engineering issues related to the manufacturing of those machines, are also present in the design of a tracking system and should be given appropriate consideration.

Ch. 1.c) Main Results in this Thesis

In the following we will concentrate on the tracking phase of the tracking of a 3-D target moving in 3-D and assume that during the acquisition phase the target was brought to the center of the image plane.

Previous approaches to the problem are composed of 3 main steps : First they perform segmentation of the image or of appropriately selected parts of it in order to locate the target and extract the new position of some characteristic points of its image (centroid, feature points, etc.). Second they match those new characteristic points with their corresponding ones in previous or current (in the binocular case) frames and thus extract information about the displacement of the target. Third they compute the necessary camera rotation that will achieve tracking.

In the sequel we describe those steps and the problems that they introduce in more detail.

In the first step, *image segmentation* is used in order to separate the target from its background. In most early tracking schemes, segmentation is done at every new frame in order to specify the target location. In later schemes, segmentation is done mainly during the acquisition phase and windows are specified around the target or around feature points of the target, which are updated during the tracking phase according to the estimated target motion. These windows facilitate segmentation by restricting it to a relatively small area of the image.

In Computer Vision, Image Segmentation has been established as one of the most difficult problems, especially in complex scenes. Segmentation algorithms based on thresholding and stochastic image models for image generation have been developed, but their usefulness is usually limited for natural images, due either to their assumptions for the visual world or the need for continuous adjustment of thresholds in the case of varying imaging conditions.

The second step is a generalization of the centroid algorithms used in tracking planar targets moving in 2–D. In the 2–D case the centroid of the image corresponds to a specific physical point on the target or in the area of the target and, most important, will continue to correspond to the *same* physical point during the entire motion of the target. Therefore, the claim that its motion describes the motion of the whole target is justified, since the motion of every other point of the rigid target is kinematically related to the motion of this characteristic point. If we attempt to use a similar analysis in 3–D though, it is going to fail, since, as the target moves, the center of brightness at each time instant will not correspond to the same physical point on the target (this can be readily seen by considering the tracking of a car moving towards the observer; initially the centroid will probably correspond to a point of the front windshield, but as the car passes in front of the observer, the new centroid will correspond to a point on the door). Therefore, knowing the motion of the centroid does not necessarily provide the motion of every other point of the target, thus it does not characterize satisfactorily the motion of the target. Then the computationally simple tracking methods of the 2–D case, based on the recovery and tracking of the image center of brightness, have to be replaced by harder ones in the 3–D case, which use feature extraction, matching and feature tracking. In this case feature correspondence between consecutive frames has to be accounted for and the difficulty of this problem is proportional to the complexity of the scene.

The third step involves the computation of the camera angular velocity ω needed to achieve tracking. Usually ω is computed as $\omega = Ke$, where e is the displacement of one of the characteristic points computed in the second step and K a constant. Clearly the issue here is whether this displacement has been computed accurately despite all the segmentation and point correspondence problems and, more important, whether the displacement of this point really characterizes the displacement of the target.

In our approach we propose a class of tracking algorithms which will perform best when segmentation and feature correspondence–based algorithms perform worst, namely

in the case of a heavily textured 3–D target moving over a heavily textured background, which is the case in most natural images.

The fundamental question every visual tracking scheme has to answer is "Where is the new position of the target?" Previous methods attempted to do so by exploiting only "static" information, such as the one provided by each separate frame. We attempted to exploit *directly* information about the temporal variation of the image induced from the target motion, such as the one provided by *optical flow*. The target separation problem has then a very elegant solution based on "dynamic" information from consecutive frames, which enables us to distinguish what moves (target) from what doesn't (background). The carriers of information are then the derivatives of the image intensity functions, not the displacements of heuristically specified characteristic points of the target.

For example, in the case of a ball moving over a uniform background, the viewing situation is simple enough to allow segmentation and feature correspondence–based algorithms to perform satisfactorily. But in the case of a book moving in front of a bookcase full of books, the edge detection algorithm will produce such a large number of edges, that the segmentation task will become impossible. On the other hand, the image spatiotemporal derivatives will capture the variation of the image generated from the target motion and optical flow information will *segment dynamically* the image into moving and static parts.

In our approach, we consider a general criterion for 3–D tracking, the *Tracking Constraint*, which, if satisfied, guarantees tracking in the sense of keeping an initially foveated target stationary in the center of the camera, without being restricted to a specific target or visual environment. We follow the Continuous Motion approach and use the concept of optical flow to formulate this constraint. The Tracking Constraint uses global information from entire areas of the image, not local information from specific points and does not require feature extraction or matching. It requires though knowledge of either the optical flow field in a neighborhood of the origin of the image plane or the shape and motion of the target. Computing the optical flow field has been a research topic in Vision that attracted much interest recently and many algorithms have been devised for its computation. Since this still remains a difficult task for images with discontinuities, the Tracking Constraint was also formulated in terms of shape and 3–D motion instead of the optical flow field. An algorithm is presented, which estimates the

3–D motion of the target from image data, provided its shape is known. This estimate of the target motion parameters, together with the known shape of the target, are used from the Tracking Constraint to produce the desired camera angular velocity that will track the target. A requirement for knowledge of the shape in a general tracking scheme is by far less restrictive than an ad–hoc tracking scheme for a specific target or class of targets as was done before, since it may be used for any tracking problem where the shape is known exactly or approximately. This is a very common situation in practice and even in the human visual system, one may argue that in most cases shape is known through learning.

Assuming that we know the *optical flow* the tracking and optical flow constraints are combined to provide a cost functional, whose minimizer is the desired camera angular velocity that will achieve tracking. This minimization problem is solved in two ways : first, by formulating the problem as an equality constrained problem and applying the classical results of Optimization Theory and second, by formulating the problem as a penalized–least squares problem and applying Generalized Cross–Validation techniques.

If, on the other hand, we assume that we know the *shape* of the target, we can use the theory of linear features and moment invariants and the optical flow constraint to derive an algorithm for the estimation of the 3–D motion of the target. The *linear features* are global quantities that encompass information about whole areas of the image, not just specific points of it. Then we can use the Tracking Constraint, together with the estimated 3–D target motion parameters, to specify the desired camera angular velocity from an unconstrained minimization problem.

Neither approach uses point correspondence of some kind or local features of the image.

Simulations were made to test our algorithms and their performance and robustness in the presence of noise. Because of lack of a good algorithm for accurate computation of the optical flow in a generic imaging situation, our main focus has been on the algorithms that assume knowledge of the shape of the target, estimate its 3–D motion and use the Tracking Constraint in order to achieve tracking. The relevant results demonstrate very desirable properties for these algorithms, namely satisfactory 3–D motion estimation, accurate tracking of the target and graceful degradation of the algorithm in the presence of noisy data.

CHAPTER

TWO

THE SYSTEM MODEL

In this Chapter we will devise the model we use in our analysis of the tracking process. In Ch. 2.a the Camera Model will be provided, which gives an approximate answer about how the image is formed on the retina of the eye or the film of the camera. In Ch. 2.b the motion of the target is described with respect to a number of appropriate coordinate systems. This analysis is particularly useful for simulating the relationship between the 3–dimensional target motion and its projected motion on the image plane. In Ch. 2.c, the tracking process will be explicitly defined and a Computational Theory (in the sense of Marr) for Tracking will be derived from the above modeling process.

2.a) The Image Formation Model

Tracking requires that a mathematical model be established, which describes the relationship between the target and its image taken by the camera and between information derived from this image and the kinematic parameters of the motion of the target. To this end, we will define the coordinate systems that we will need and provide their relationships.

Suppose there exists an inertial frame in $I\!R^3$, the **world coordinate system** $SX_0Y_0Z_0$, with respect to which all quantities are measured (fig. 2.1) .

The ideal pinhole camera model, shown in fig. 2.2, is used. An image is a two–dimensional pattern of brightness projected on the image plane. Consider the **camera**

coordinate system $OXYZ$, with Z along the optical axis, i.e. the perpendicular from the pinhole to the image plane pointing toward the scene. This induces a two-dimensional coordinate system, the **image-plane coordinate system** oxy on the image plane. The *focal length* f of the camera is defined as the distance Oo.

Consider finally a **target coordinate system** $WX'Y'Z'$ attached to the rigid body that we want to track.

If $P : I\!R^3 \rightarrow I\!R^2$ is the projection function, consider a point A in $I\!R^3$ with coordinates $\vec{\mathrm{X}} = (X, Y, Z)$ relative to the camera coordinate system, which is projected to the point P_A of the image plane with coordinates $\vec{\mathrm{x}} = (x, y)$ (fig. 2.2) . Then :

$$\vec{\mathrm{x}} = \begin{pmatrix} x \\ y \end{pmatrix} = P(\vec{\mathrm{X}}) = \begin{pmatrix} P_1(\vec{\mathrm{X}}) \\ P_2(\vec{\mathrm{X}}) \end{pmatrix}. \qquad (2.a.1)$$

Under Orthographic Projection we have:

$$\vec{\mathrm{x}} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix}. \qquad (2.a.2)$$

Under Perspective Projection we have:

$$\vec{\mathrm{x}} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \end{pmatrix}. \qquad (2.a.3)$$

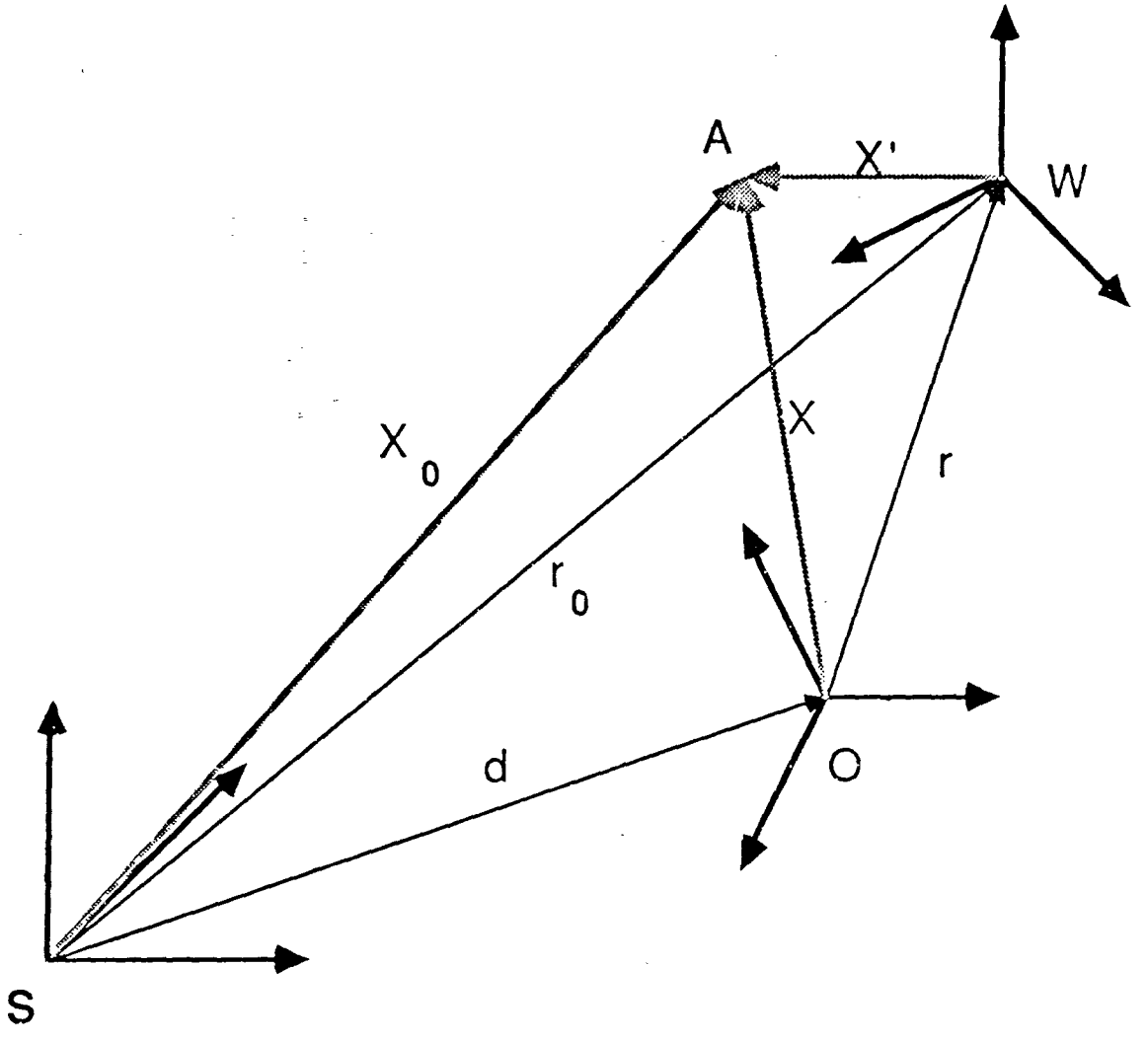


Fig. 2.1 : The World, Camera and Target Coordinate Systems
(General Case)

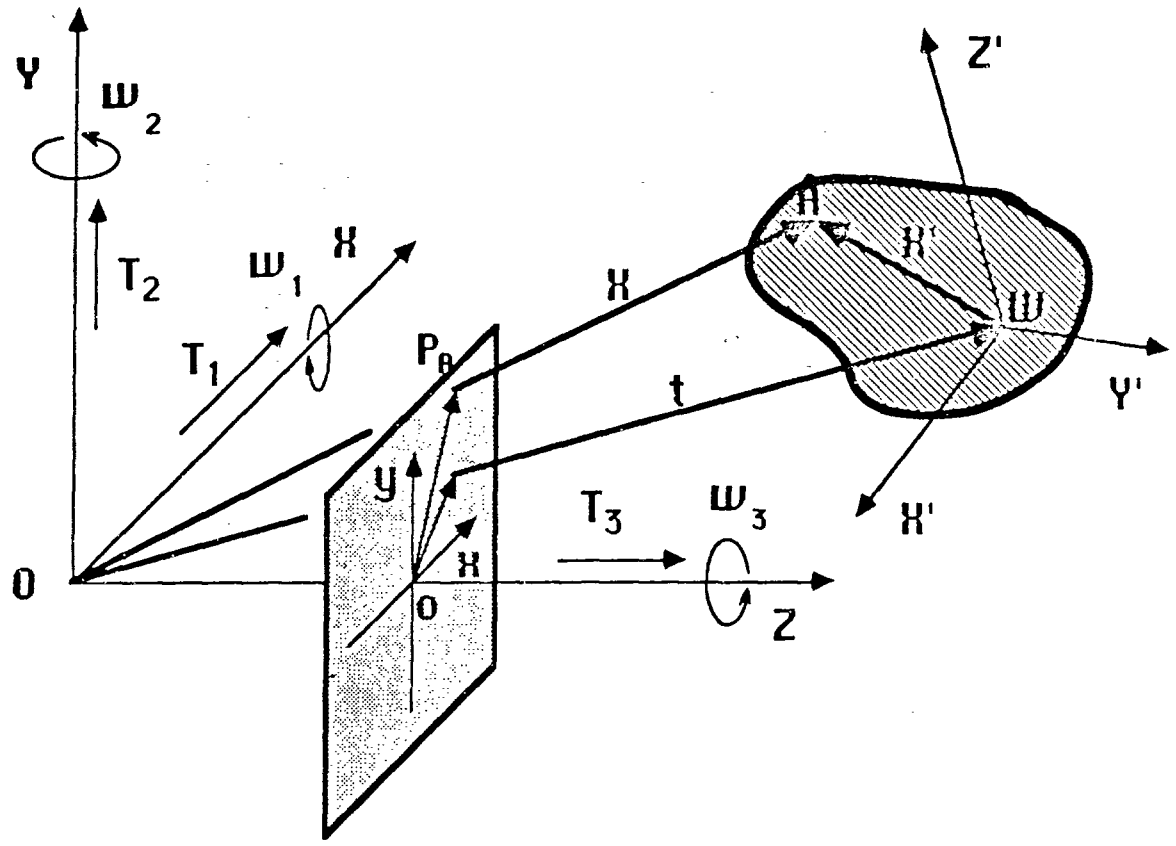


Fig. 2.2 : The Camera and Target Coordinate Systems

2.b) The System Kinematics

Consider the position vector q of a point relative to an inertial coordinate system k and its position vector Q relative to a moving coordinate system K, where k and K are oriented Euclidean spaces (fig. 2.3) .

Theorem 2.1 :

Consider two infinitesimally separated time instants t and t_0, such that $t = t_0 + dt$.

i)

$$q(t) = r(t) + B(t)Q(t) \,, \tag{2.b.1}$$

with $B(.) \in I\!R \times SO(3)$, where SO(3) is the Special Orthogonal (Rotation) Group of order 3.

ii)

$$\frac{dq(t_0)}{dt} = T + \hat{\omega}X(t_0) + B(t_0)\frac{dQ(t_0)}{dt} \,, \tag{2.b.2}$$

where T the translational velocity of K, defined as :

$$T = \frac{dr(t_0)}{dt} - \hat{\omega}r(t_0) \tag{2.b.3}$$

and ω the spatial angular velocity of K, defined as :

$$\hat{\omega} = \frac{dB(t_0)}{dt}B^{\top}(t_0) \,. \tag{2.b.4}$$

Moreover,

$$r(t) = r(t_0) + \frac{dr(t_0)}{dt}dt = Tdt + (I + \hat{\omega}dt)r(t_0) \tag{2.b.5}$$

and

$$B(t) = B(t_0) + \frac{dB(t_0)}{dt}dt = (I + \hat{\omega}dt)B(t_0) \tag{2.b.6}$$

Proof : See Appendix A.

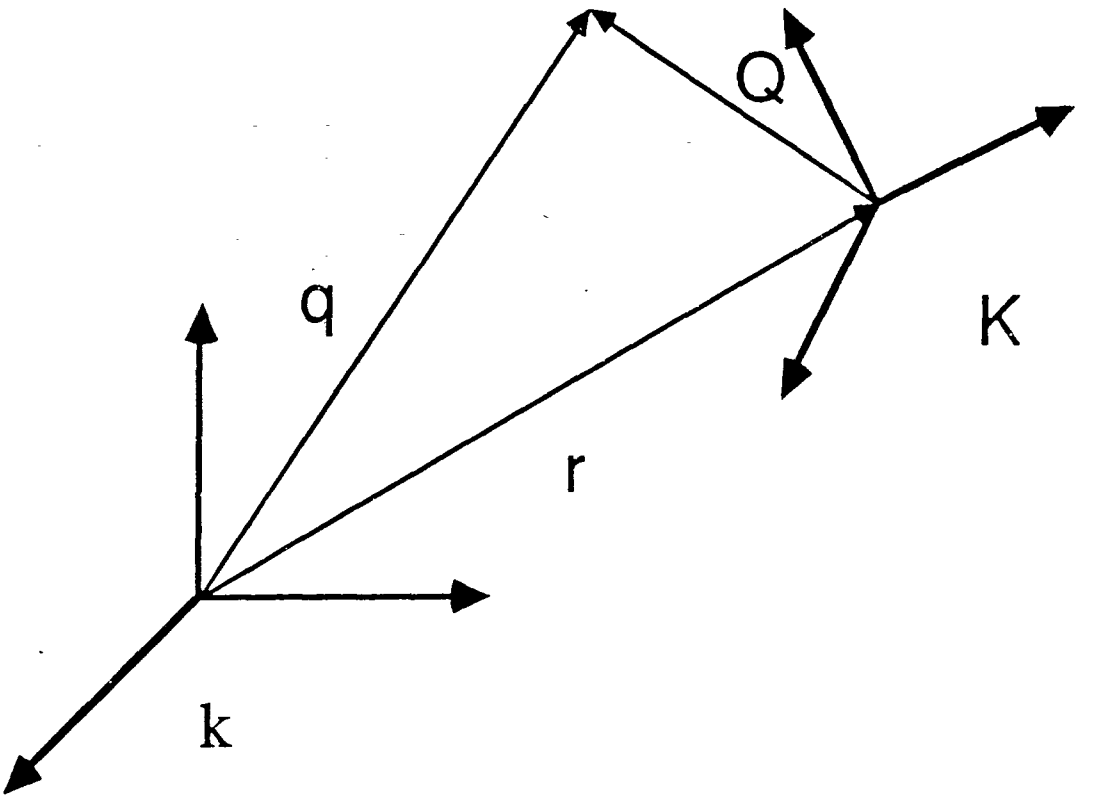


Fig. 2.3 : Inertial and Moving Coordinate Systems

Consider the three coordinate systems describing the motion of the camera and the target as shown in fig. 2.1. Consider a point A of the target with position vectors X_0, X and X' relative to the world, camera and target frames respectively.

Assume the target is stationary with respect to the target coordinate frame, i.e. $\dot{X}' = 0 \Longrightarrow X'(t) = X'(t_0) \stackrel{def}{=} X'$ and is moving rigidly with respect to the other coordinate frames.

Corollary 2.1 :

Consider the target with its point W (fig. 2.1) moving with translational velocity T and with spatial angular velocity ω relative to the *camera* frame.

i) $X(t) = r(t) + R(t)X'$, with $R(.) \in I\!R \times SO(3)$.

ii) $\frac{dX(t_0)}{dt} = T + \hat{\omega}X(t_0)$, where $\hat{\omega} = \frac{dR(t_0)}{dt}R^T(t_0)$ and $T = \frac{dr(t_0)}{dt} - \hat{\omega}r(t_0)$.

Moreover,

$$r(t) = r(t_0) + \frac{dr(t_0)}{dt}dt = Tdt + (I + \hat{\omega}dt)r(t_0)$$

and

$$R(t) = R(t_0) + \frac{dR(t_0)}{dt}dt = (I + \hat{\omega}dt)R(t_0)$$

Proof : Result is immediate from the Theorem.

Corollary 2.2 :

Consider the target with its point W (fig. 2.1) moving with translational velocity T_0 and with spatial angular velocity ω_0 relative to the *world* frame.

i) $X_0(t) = r_0(t) + R_0(t)_0X'$, with $R_0(.) \in I\!R \times SO(3)$.

ii) $\frac{dX_0(t_0)}{dt} = T_0 + \hat{\omega}_0X_0(t_0)$, where $\hat{\omega}_0 = \frac{dR_0(t_0)}{dt}R_0^T(t_0)$ and $T_0 = \frac{dr_0(t_0)}{dt} - \hat{\omega}_0r_0(t_0)$.

Moreover,

$$r_0(t) = r_0(t_0) + \frac{dr_0(t_0)}{dt}dt = T_0dt + (I + \hat{\omega}_0dt)r_0(t_0)$$

and

$$R_0(t) = R_0(t_0) + \frac{dR_0(t_0)}{dt}dt = (I + \hat{\omega}_0dt)R_0(t_0)$$

Proof : Result is immediate from the Theorem.

Corollary 2.3 :

Consider the camera with its point O (fig. 2.1) moving with translational velocity T_c and with spatial angular velocity ω_c relative to the *world* frame.

i) $X_0(t) = d(t) + \mathcal{A}(t)X(t)$, with $\mathcal{A}(.) \in I\!R \times SO(3)$.

ii) $\frac{dX_0(t_0)}{dt} = T_c + \hat{\omega}_c X_0(t_0) + \mathcal{A}(t_0)\dot{X}(t_0)$, where $\hat{\omega}_c = \frac{d\mathcal{A}(t_0)}{dt}\mathcal{A}^T(t_0)$ and $T_c = \frac{dd(t_0)}{dt} - \hat{\omega}_c d(t_0)$.

Moreover,

$$d(t) = d(t_0) + \frac{dd(t_0)}{dt}dt = T_c dt + (I + \hat{\omega}_c dt)d(t_0)$$

and

$$\mathcal{A}(t) = \mathcal{A}(t_0) + \frac{d\mathcal{A}(t_0)}{dt}dt = (I + \hat{\omega}_c dt)\mathcal{A}(t_0)$$

Proof : Result is immediate from the Theorem.

Theorem 2.2 :

Consider the system of fig. 2.1. with $T_0, T, T_c, \omega_0, \omega$ and ω_c defined as above.

i)

$$X_0(t) = r_0(t) + R_0(t)X' \,. \tag{2.b.7}$$

ii)

$$\frac{dX_0(t_0)}{dt} = T_0 + \hat{\omega}_0 X_0(t_0) \,, \tag{2.b.8}$$

where

$$\hat{\omega}_0 = \hat{\omega}_c + \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^T(t_0) \tag{2.b.9}$$

and

$$T_0 = T_c + \mathcal{A}(t_0)\hat{\omega} - \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^T(t_0)d(t_0) \,. \tag{2.b.10}$$

Moreover,

$$r_0(t) = d(t) + \mathcal{A}(t)r(t) = T_0 dt + (I + \hat{\omega}_0 dt)r_0(t_0)$$

and

$$R_0(t) = \mathcal{A}(t)R(t) = (I + \hat{\omega}_0 dt)R_0(t_0)$$

Proof : See Appendix A.

In the following we will consider that the origin S of the world and O of the camera frame coinside (fig. 2.4) .

From the above modeling process an important, for our subsequent analysis, fact emerges : The effect of a rotation of the camera on the image formation process, is exactly the same to the effect of an opposite rotation of the target on the process. Put in another way, since the formation of an image on the image plane depends on the relative motion of the camera and the target, it is irrelevant to this process whether the camera is rotating in one direction or the target is rotating in the opposite one. More formally, we have :

Corollary 2.4 :

Consider the system of fig. 2.4. Let the camera rotate with angular velocity ω_c with respect to the world coordinate system at time t, while the target remains stationary. Then, in the camera coordinate system, the target appears to rotate with angular velocity ω such that :

$$\omega = -\tilde{\omega}_c \,, \tag{2.b.11}$$

where

$$\tilde{\omega}_c = A^T(t_0)\hat{\omega}_c A(t_0) \,. \tag{2.b.12}$$

Proof : The result is immediate from Theorem 2.2 with $d = 0, T_0 = 0, \omega_0 = 0$ and $T_c = 0$.

Using this fact in the next section (Ch. 2.c) we will refer to the "camera angular velocity that achieves tracking" and mean $\tilde{\omega}_c$, not ω_c. This will allow us to make the entire analysis of Ch. 3 and 4 in the camera coordinate frame and achieve a remarkable simplification of the corresponding formulae.

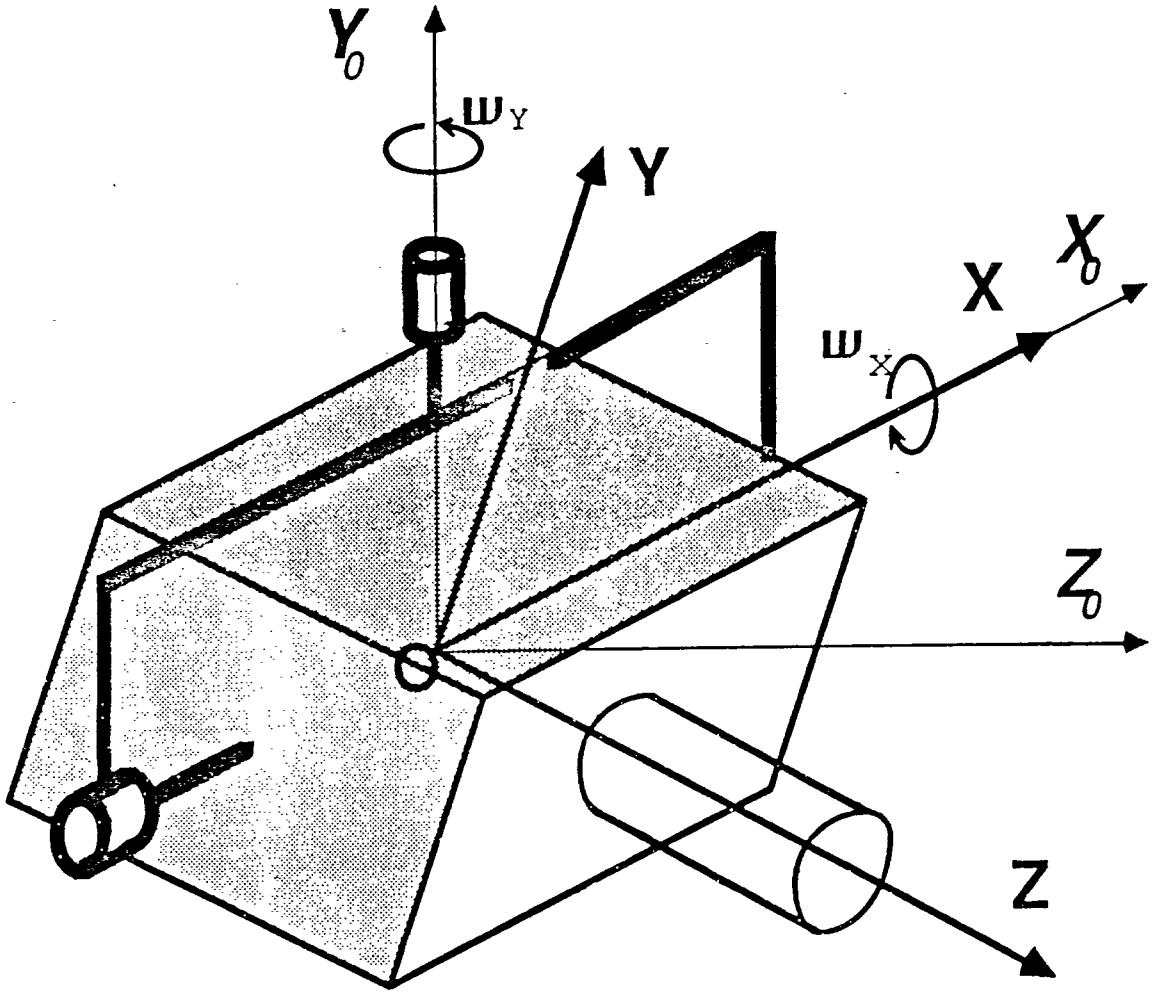


Fig. 2.4 : The World and Camera Coordinate Systems

2.c) Computational Theory of Tracking

2.c.I) Continuous Motion and Optical Flow

Optical Flow is the velocity field generated on the image plane either from the projection of objects moving in 3-D or from the motion of the observer with respect to the scene or from apparent motion, when a series of images gives the illusion of motion.

A review of the psychological work related to optical flow generated from apparent motion is given in Marr [40] and Schunk [66]. Optical flow is used quite extensively for the analysis of the Short-Range processes (Ullman [71]), e.g. in the Continuous Motion approach in the Structure from Motion problem, etc. The main problem in those schemes has always been the estimation of the optical flow field for a sequence of natural images. The task has been proved very difficult, mainly because of occluding boundaries, incompatible flow constraints from objects with different motions and noisy images.

Optical Flow Constraint Equation

The image intensity s(x,y,t) at a point (x,y) of the image plane at time t is related to the instantaneous velocity $(u^t(x,y), v^t(x,y))$ at that point with the equation:

$$s_x(x,y,t)u^t(x,y) + s_y(x,y,t)v^t(x,y) + s_t(x,y,t) = 0\,, \qquad (2.c.1)$$

The proof ,which is very easy if s(x,y,t) does not contain discontinuities and there are no optical flow discontinuities due to occlusion, can be found in Horn and Schunk [41]. Based on (2.c.1), several algorithmic schemes for the computation of optical flow were devised and will be presented later (Ch. 3.a).

Schunk [66] extended the above result so that (2.c.1) holds in the more general case where the image is smooth everywhere, except at a finite number of discontinuities, and the perceived change in image irradiance at each point on the image plane is entirely due to motion of the image pattern, not to reflectance effects. Moreover, he demonstrated that severe errors in (2.c.1) are produced from sampling the time-varying intensity function s(x,y,t) in order to produce an image sequence.

2.c.II) The Active Vision Paradigm for Early Vision Problems

The ill-posed nature of most early vision problems led to attempts to make them well–posed, but in the same time avoid the regularization process, which still presents several untractable problems, especially in the analysis of complex scenes. A very important and fruitful one was driven by the observation that human perception of a dynamically evolving scene, is an equally dynamical act, whether conscious or unconscious. As we saw in Ch. 1, humans tend to move their eyes and heads in a variety of ways, in order to obtain a better view of a scene and adjust to its specific visual requirements. In general, they attempt to focus on a certain moving object of interest and track it in time. After the object is stabilized on the retina, extraction of information about its structure and properties, as well as its motion, is no longer limited by blurring associated with motion and becomes much easier. This observation led to the *Active Vision Paradigm*, which has been recently introduced (Aloimonos, Weiss and Bandopadhay [2]). The key idea is that if we are able to move the camera at will and, if needed, focus on a target and track it, then many significant ill–posed problems in early vision become well–posed (Shape from Shading, from Contour, from Texture), while others which are unstable or where non–linear constraints have to be solved, become stable with simpler linear constraints (Structure from Motion, Egomotion).

Next, we will present an overview of the research in each of those problems and summarize the basic results of Active Vision related to each one. Finally, we will show how target tracking can influence the results of this new approach for the Egomotion problem.

The Shape from Shading problem is the recovery of a surface from image data, i.e. the recovery of its orientation at each point of the image. From the image irradiance equation we get:

$$E(\vec{x}) = R(\vec{n}, \vec{v}, \vec{s}), \qquad (2.c.2)$$

where $E(\vec{x})$ is the image irradiance at a point $\vec{x}$ of the image plane and is available by direct measurement, $R(\vec{n}, \vec{v}, \vec{s})$ is the surface reflectance map, which specifies the radiance of a surface patch as a function of its orientation $\vec{n}$, the viewing direction $\vec{v}$ and the light distribution and surface properties $\vec{s}$. The problem is to solve (2.c.2) and

recover $\vec{n}$ at each point $\vec{x}$. Several interesting attempts were made to solve this problem by variational and regularization methods (Horn[40]), which amount to formulating a surface functional depending on $\vec{n}(\vec{x})$, which should be minimal for the correct surface. Moreover, two cameras were introduced either very close to each other, or far apart, but then the correspondence problem has to be accounted for and current techniques for that are not entirely satisfactory ([40]). In the monocular case noisy data restrict severely the performance of these methods. The Active Vision provides the flexibility of controlling $\vec{v}$ in (2.c.2) and thus more viewpoints are available and the geometry of the visible surface can be recovered at each individual point.

The Shape from Contour problem is the recovery of 3–D shape and surface orientation from a 2–D contour. Using the passive approach, for a planar contour in the case of the monocular observer an infinity of solutions are possible for the orientation of the corresponding plane. Several assumptions and heuristic schemes were introduced, but the results were not generally satisfactory. In the binocular case, either we encounter again the correspondence problem, or in other formulations two sets of non–linear constraints extracted from the areas enclosed by the projections of the contours and from the perimeters of those areas, have to be solved. Even when this is possible, uniqueness and stability of the solutions cannot be guaranteed. The Active observer can rotate the cameras and, since occlusion will not be affected by the rotation, this will provide additional information about the areas of the two projections. This can lead to a system of equations linear in the shape parameters.

The Shape from Texture problem is the recovery of the orientation of a surface covered with texture, from a monocular view. In the passive observer analysis, several assumptions were made in order to overcome the non–uniqueness of the possible solutions, such as the uniform distribution of texels (Gibson [30], Bajcsy and Lieberman [7]) and, more recently, the directional isotropy of the peripheral contours in the figures of the texture, i.e. that those contours have edge segments uniformly distributed over all orientations (Witkin [79]). In the case of an Active observer and under the assumption that the observer motion is controllable, an analysis, much like the one we present in Ch. 4.b, leads in the planar case to a linear system, where from the shape parameters can be recovered.

The problem of Structure from Motion consists of recovering the shape of a moving object from a sequence of images. Closely related is the problem of Passive Navigation,

which is the recovery of 3-D motion, also from a sequence of images. For the Continuous case the two problems are equivalent in the sense that knowledge of the optical flow field and of the motion allows the unique recovery of structure. On the other hand, knowledge of the optical flow and the structure allows the unique specification of the motion parameters. For the Continuous case of the Passive Navigation problem, it was proved (Bruss and Horn [21], Prazdny [57]) that if the optical flow is known, recovery of 3-D motion is possible, using least-squares methods. Problems with this approach are the non-linearity of the relevant constraints, where from the need for iterative solution schemes emanates, and the high dimension of the space of unknowns. In the case of the monocular observer there is a degeneracy in this problem, since, even if the optical flow is known, not all 6 parameters of the motion can be recovered, but only the direction of translation and the rotation of the target (5 parameters). Longuet-Higgins and Prazdny [47] derived closed-form solutions to the problem, but used higher-order derivatives of the optical flow, which makes their scheme very error-prone in noisy data.

If the Active observer is able to track a prominent feature of the target, it was shown (Bandopadhay [10]) that in the monocular case the dimension of the space of unknowns reduces to four, without increase in the degree of nonlinearity of the respective constraints, while in the binocular case, when each of the two cameras tracks the target, it was shown that closed form solutions can be obtained for the Navigation problem and that if the rotation, angular velocity and angular acceleration of the tracking observer are available, the optical flow need not be computed. Moreover, the Active Vision approach attacks not just the passive navigation problem, where an observer is moving with respect to a static scene, but also the more general one, namely Egomotion, when the observer is moving in a moving environment. This approach is named Active Navigation. Therefore, if the tracking schemes presented here are combined with the Active Navigation method, a solution to the continuous case of the Egomotion problem can be formulated.

2.c.III) The Tracking Constraint

Let $\mathcal{R}$ be the projection of our target on the image plane S. Obviously $\mathcal{R} \subset S \subset I\!R^2$. Suppose that our target has a prominent feature that we want to track, located at a point W in $I\!R^3$, which can be easily identified during the target acquisition phase and which is projected to a point P_W of the image plane. Because we will devise tracking schemes

where only global information from large areas of the image will be used, not local one from specific points, we don't need an exact solution to the feature extraction problem, i.e. the exact location of the point P_W. All we need is a neighborhood $\mathcal{B}_W(P_W, \epsilon_W) \subset \mathcal{R}$, with ϵ_W acceptably small. This is very important, since recovering the area $\mathcal{B}_W$ in the target acquisition phase is going to be much more robust to noise, than attempting to recover a single point P_W.

Suppose that during the target acquisition phase, the area $\mathcal{B}_W$ was recovered and moved at the origin of the image plane.

We define the *Tracking Problem* as follows : Find the camera rotation (ω_x, ω_y) that will hold $\mathcal{B}_W(P_W, \epsilon_W)$ stationary at a neighborhood $\mathcal{B}(\mathbf{o}, \epsilon)$ of the origin $\mathbf{o}$ of the image plane.

Suppose now that we get an image s(x,y,t) at time t and we want to solve the tracking problem defined above. We must specify a camera rotation (ω_x, ω_y) that will act from time t to $t + dt$ and that will keep $\mathcal{B}_W$ at the origin despite the motion of the target. Let the optical flow at time t be $(u^t(x,y), v^t(x,y))$ at a point $(x,y) \in S$ and at time $t + dt$ be $(u^{t+dt}(x,y), v^{t+dt}(x,y))$. Now, (u^{t+dt}, v^{t+dt}) will be the superposition of the optical flow induced by two motions : The motion of the target by (T, ω) and the tracking motion of the camera by $(\omega_x, \omega_y, 0)$. Suppose that due to the infinitesimal nature of the motions analyzed in the Continuous Motion approach to the Egomotion problem, the target motion in the time interval [t, t+dt] that will produce the first component of (u^{t+dt}, v^{t+dt}) will be almost identical to its motion in the time interval [t-dt, t] that induced the optical flow (u^t, v^t). Therefore this first component of (u^{t+dt}, v^{t+dt}) will be (u^t, v^t) and is supposed to be known or retrievable from visual data. The second component of (u^{t+dt}, v^{t+dt}) will be induced from the motion of the camera and let it be (u_{CAM}, v_{CAM}).Therefore:

$$u^{t+dt} \approx u^t + u_{CAM} \tag{2.c.3}$$

and

$$v^{t+dt} \approx v^t + v_{CAM} \,. \tag{2.c.4}$$

Let see now exactly how the motion influences the induced optical flow.

Theorem 2.3 :

Let the point W of the target move with translational velocity $T = (T_1, T_2, T_3)^\mathsf{T}$ and the target with angular velocity $\omega = (\omega_1, \omega_2, \omega_3)^\mathsf{T}$ at time t relative to the camera

frame. This target motion induces on the image plane an optical flow field, which at the projection (x, y) on the image plane of the point (X, Y, Z) of the surface of the target, will be:

$$u = \frac{fT_1 - xT_3}{Z} - \omega_1 \frac{xy}{f} + \omega_2 \frac{(x^2 + f^2)}{f} - \omega_3 y \tag{2.c.5}$$

$$v = \frac{fT_2 - yT_3}{Z} - \omega_1 \frac{(y^2 + f^2)}{f} + \omega_2 \frac{xy}{f} + \omega_3 x \tag{2.c.6}$$

Proof : See Appendix A.

Corollary 2.5 :

For focal length $f = 1$ we get :

$$u^t(x, y) = \frac{T_1 - xT_3}{Z} - \omega_1 xy + \omega_2(x^2 + 1) - \omega_3 y \tag{2.c.7}$$

$$v^t(x, y) = \frac{T_2 - yT_3}{Z} - \omega_1(y^2 + 1) + \omega_2 xy + \omega_3 x\,. \tag{2.c.8}$$

Proof : Result is immediate from the Theorem.

The more general case of an arbitrary projection $P : I\!R^3 \rightarrow I\!R^2$ is treated in Ch. 4.b.

Therefore u^t, v^t are related to the target motion by (2.c.5) and (2.c.6). Let see how u_{CAM}, v_{CAM} are related to the camera motion.

Here we take advantage of the nature of optical flow as the velocity field induced by the *relative* motion of the camera and the target; therefore, we can argue somehow informally that the camera moving by ω_x, ω_y generates optical flow u_{CAM}, v_{CAM}, which is exactly the same to the one that would be generated by the target, if it underwent an additional motion by $-\omega_x, -\omega_y$. This was formally shown in Corollary 2.4.

Then the respective camera-induced optical flow will be given from (2.c.5) and (2.c.6), with T=0 and $\omega_1 = -\omega_x$, $\omega_2 = -\omega_y$, $\omega_3 = 0$:

$$u_{CAM} = \omega_x \frac{xy}{f} - \omega_y \frac{(x^2 + f^2)}{f} \tag{2.c.9}$$

$$v_{CAM} = \omega_x \frac{(y^2 + f^2)}{f} - \omega_y \frac{xy}{f} \tag{2.c.10}$$

Thus, assuming we know (u^t, v^t), for $f = 1$ the total optical flow at time t+dt will be:

$$u^{t+dt} = u^t + \omega_x xy - \omega_y(x^2 + 1) \tag{2.c.11}$$

$$v^{t+dt} = v^t + \omega_x(y^2 + 1) - \omega_y xy \tag{2.c.12}$$

Now, for the tracking to be successful, at time t+dt the point P_W must be stationary at the origin and therefore the optical flow at the origin must be zero.Therefore we want:

$$u^{t+dt}(0,0) = v^{t+dt}(0,0) = 0 \tag{2.c.13}$$

or

$$u^t(0,0) + u_{CAM}(0,0) = v^t(0,0) + v_{CAM}(0,0) = 0$$

and the idea is that by appropriate choice of ω_x and ω_y, the induced u_{CAM} and v_{CAM} be such that (2.c.13) holds.

In order to increase robustness, in addition to (2.c.13), we require that the optical flow on an entire neighborhood $\mathcal{B}(o, \epsilon)$ of the origin (with $\epsilon \geq \epsilon_W$), be minimal. This is then our Tracking Constraint and is formulated as follows:

The Tracking Constraint : The desired camera angular velocity (ω_x^*, ω_y^*) that will achieve tracking is the minimizer of:

$$\begin{aligned} f(\omega_x, \omega_y) &= \iint_{\mathcal{B}} \left[(u^{t+dt})^2 + (v^{t+dt})^2\right] dx\, dy \\ &= \iint_{\mathcal{B}} \left[\left(u^t + \omega_x xy - \omega_y(x^2+1)\right)^2 + \left(v^t + \omega_x(y^2+1) - \omega_y xy\right)^2\right] dx\, dy \end{aligned} \tag{2.c.14}$$

Notice that the tracking constraint is expressed in terms of the neighborhood $\mathcal{B}$ of the origin of the camera, not some neighborhood of a target feature whose specification would require segmentation of the image and feature extraction during the tracking phase.

An **algorithm** then for tracking the motion of an object moving in 3–D, suggests itself :

Step 1 :

Compute the optical flow induced by this motion, either directly from the image, or, first, compute $(\frac{T_1}{Z}, \frac{T_2}{Z}, \frac{T_3}{Z}, \omega_1, \omega_2, \omega_3)$ from the image and then derive the optical flow using eq. (2.c.5) and (2.c.6).

Step 2 :

Minimize the expression (2.c.14) and derive the desired camera angular velocities that will achieve tracking.

Typically Step 1 involves the Optical Flow Constraint and Step 2 the Tracking Constraint. In Step 1 if we assume that the Optical Flow is computed directly from the image, we get the class of algorithms presented in Ch. 3. If we attempt to compute first the target motion parameters, we get the algorithms presented in Ch. 4.

CHAPTER

THREE

TRACKING USING KNOWLEDGE OF OPTICAL FLOW

In this Chapter we will present tracking strategies that will assume knowledge of the Optical Flow on the entire image. Methods for the recovery of the optical flow field from the image intensity function s(x,y,t) will be reviewed in Ch. 3.a. In Ch. 3.b the utilization of this information in order to track a moving target will be formulated as a *Constrained Optimization* problem (Tracking Schema 3.1) and as a *Penalized Least-Squares* problem (Tracking Schema 3.2), using the Tracking and the Optical Flow Constraint equations presented in Ch. 2.

3.a) Methods for the Recovery of Optical Flow

The difficulty in computing optical flow starts from difficulty in detecting motion. Due to the aperture problem, local operators are unable to detect motion tangent to a moving contour. Moreover, the ambiguity of the optical flow field, i.e. the non-uniqueness of the possible estimates of this field from a sequence of images, makes necessary additional constraints about the imaging situation.

Hildreth [35] presents an overview of previous methods to compute optical flow and classifies them in three categories according to the major assumption they make about the nature of the motion that generated the image : Algorithms of the first category assume locally constant velocity over small areas of the image. Algorithms of the second category assume rigid motion of the projections of objects on the image plane

(not all 3-D motions generate projected motion that corresponds to 2-D rigid motion). Algorithms of the third category assume that the velocity field is smooth, claiming that the real world consists mainly of solid objects with smooth surfaces, which, in general, will generate a smoothly varying velocity field. One of the first algorithms that made this last assumption was presented in Horn and Schunck [41], where a functional of the type

$$\iint [(s_x u + s_y v + s_t)^2 + \alpha^2(u_x^2 + u_y^2 + v_x^2 + v_y^2)]\, dx\, dy$$

was used for the computation of (u, v) and expressed the requirement that the optical flow satisfy the Constraint Equation and be as smooth as possible. In this paper they did not present a specific method for the computation of α, but later Poggio, Torre and Koch [56] mentioned the use of Generalized Cross-Validation techniques.

Hildreth [35] examined the extraction of the optical flow field from moving contours and ways to bypass the aperture problem. If s denotes arclength on the contour, the optical flow V(s) can be decomposed into components tangent and perpendicular to the contour as follows: $\vec{V}(s) = V^{\top}(s)\vec{u}^{\top}(s) + V^{\perp}(s)\vec{u}^{\perp}(s)$, where $\vec{u}^{\top}(s)$ and $\vec{u}^{\perp}(s)$ are the corresponding unit vectors Supposing that $V^{\perp}(s)$ can be measured from zero-crossings in the output of the convolution of the image with an $\nabla^2 G$ local operator, but there is error in the measurements, then $V(s) = \begin{pmatrix} V_x \\ V_y \end{pmatrix}$ can be recovered from the solution of a one-dimensional minimization problem along a closed zero-crossing contour, with minimization functional of the form:

$$\int (\vec{V} \cdot \vec{u}^{\perp} - V^{\perp})^2 ds + \beta \int [(\frac{\partial V_x}{\partial s})^2 + (\frac{\partial V_y}{\partial s})^2] ds\,.$$

It was proven that a unique and physically plausible solution can be determined from the above optimization procedure.

These approaches were generalized when the ill-posed nature of the problem was realized and Tikhonov Regularization techniques became a standard analysis tool for this type of problems (Poggio, Torre and Koch [56] , Poggio and Koch [55], Verri and Poggio [74]). Tikhonov [70] proposed the pth-order Sobolev norm

$$||u||_p^2 = \sum_{m=0}^{p} \int_R w_m(x)(\frac{d^m u(x)}{dx^m})^2 dx\,,$$

as a stabilizer for univariate regularization, while for the multivariate case, generalized spline functionals of the following form were proposed (Brady and Horn [18], Terzopoulos [68]):

$$||u||_m^2 = \sum_{i_1,\ldots,i_m=1}^{d} \int_{R^d} \left(\frac{\partial^m u(\vec{x})}{\partial x_{i_1} \ldots \partial x_{i^m}}\right)^2 d\vec{x},$$

Those are quadratic functionals and when used as stabilizers in regularization they do not apply to problems involving discontinuities.

A unification of the above schemes was attempted in Brockett [19], in the context of the least squares theory, which was used in order to determine necessary and sufficient conditions for the identification of the unique optical flow field fitting best with the data and the objects.

In the regularization schemes that have been presented, the global smoothness implied by the quadratic stabilizing functionals considered, enforces smooth variation of the optical flow even across the image of occluding edges or of texture and illumination discontinuities. This is not desirable, and several methods exploiting discontinuities of scene characteristics started to emerge.

Cornelius and Kanade [23] deactivated the smoothness requirement in the neighborhood of zero–crossing contours. Nagel [52] and Nagel and Enkelmann [53] examined the 'oriented smoothness' condition, which allows variations of the displacement vector field only in directions with small or no variations of the image intensity. Under the assumption that the displacement is locally constant around points where the gradient of the intensity function changes fast enough (gray value corners), both its components are computable from the Constraint Equation, which led to a minimization functional of the type:

$$\iint (\nabla s^{\top} u + s_t)^2 + \alpha^2 trace(\nabla u^{\top} W \nabla u) dx\, dy\,,$$

where W a weight matrix that encompasses the idea of oriented smoothness (as opposed to the 'unoriented' one introduced in Horn and Schunck [41], which corresponds to W=I, the identity matrix).

Terzopoulos [69] considered the problem of specifying alternatives to Tikhonov stabilizing functionals, that will accommodate visual discontinuities of arbitrary orders and suggests a class of multidimensional 'controlled- continuity' stabilizers, where, by adjusting a set of weighting functions their continuity properties can be controlled spatially

and the stabilizers can be used to reconstruct localized discontinuities. He proposes as stabilizing functionals for piecewise continuous regularization the following combination of splines:

$$||u||^2_{p,w} = \sum_{m=0}^{p} \int_{R^d} w_m(\vec{x}) \sum_{j_1+\ldots+j_d=m} \frac{m!}{j_1!\ldots j_d!} \left(\frac{\partial^m u(\vec{x})}{\partial x_1^{j_1}\ldots\partial x_d^{j_m}}\right)^2 d\vec{x},$$

The weighting functions $w_m(\vec{x})$ are allowed to make jump transitions to zero over $I\!R^d$, which provides the capability of selectively introducing specific discontinuities into the solution (depth discontinuities, orientation discontinuities, occluding contours).

Aloimonos and Shulman [3] investigated learning techniques for the regularization of ill-posed problems with non-linear constraints. They required that the surface in view be smooth and they attempt to minimize a functional of the type :

$$\iint \left[L^2(f,g,x,y) + \lambda(f_x^2 + f_y^2 + g_x^2 + g_y^2)\right] dx\, dy,$$

where L(f,g,x,y) the nonlinear constraint derived from the 'shape from X' problems and f,g the stereographic space coordinates. They used Stochastic Approximation techniques such as the Robbins-Monro procedure to analyze learning by examples of quadratic smoothness constraints by a connectionist network. Once the network has learned the regularization parameter by examples, an iterative scheme can provide unique solutions, under certain conditions for the nonlinear constraint, to the 'shape from X' problem.

3.b) Tracking

Assume that we know the Optical Flow field (u^t, v^t) at time t for every point (x,y) of our image S. We want to specify the camera angular velocities (ω_x, ω_y) that will achieve tracking.

We saw in Corollary 2.5 that the optical flow at time $t + dt$ induced by the relative motion of the target and the camera will be :

$$u^{t+dt} = u^t + \omega_x xy - \omega_y(x^2 + 1) \qquad (3.b.1)$$

$$v^{t+dt} = v^t + \omega_x(y^2 + 1) - \omega_y xy \qquad (3.b.2)$$

As a result of successful tracking, the optical flow at $t + dt$ in a neighborhood $\mathcal{B}$ of the origin of the image plane should be minimal. Therefore, from the Tracking Constraint that we obtained in Ch. 2.c.III, the ω_x, ω_y should minimize:

$$f(\omega_x, \omega_y) = \iint_{\mathcal{B}} \left[(u^{t+dt})^2 + (v^{t+dt})^2 \right] dx\, dy$$
$$= \iint_{\mathcal{B}} \left[\left(u^t + \omega_x xy - \omega_y (x^2+1) \right)^2 + \left(v^t + \omega_x (y^2+1) - \omega_y xy \right)^2 \right] dx\, dy \tag{3.b.3}$$

Moreover, the Optical Flow Constraint Equation should hold at time $t + dt$ at every point of the image, i.e. ω_x, ω_y should be such that:

$$s_x(x,y,t+dt)u^{t+dt}(x,y) + s_y(x,y,t+dt)v^{t+dt}(x,y) + s_t(x,y,t+dt) = 0 \quad , \quad \forall (x,y) \in S \, ,$$

or, equivalently:

$$\iint_S \left[s_x(x,y,t+dt)u^{t+dt}(x,y) + s_y(x,y,t+dt)v^{t+dt}(x,y) + s_t(x,y,t+dt) \right]^2 dx\, dy \; = \; 0 \, . \tag{3.b.4}$$

If we expand s_x, s_y and s_t in Taylor series around the point (x,y,t), we get:

$$\begin{aligned} \frac{\partial s}{\partial x}(x,y,t+dt) &= \frac{\partial s}{\partial x}(x,y,t) + \frac{\partial^2 s}{\partial t \partial x}(x,y,t)dt + \dots \\ \frac{\partial s}{\partial y}(x,y,t+dt) &= \frac{\partial s}{\partial y}(x,y,t) + \frac{\partial^2 s}{\partial t \partial y}(x,y,t)dt + \dots \\ \frac{\partial s}{\partial t}(x,y,t+dt) &= \frac{\partial s}{\partial t}(x,y,t) + \frac{\partial^2 s}{\partial t^2}(x,y,t)dt + \dots \end{aligned} \tag{3.b.5}$$

In the proof of the Optical Flow Constraint (Horn and Schunck [41]), because of the Continuous Motion assumption, the second-order derivatives of s(x,y,t) at (x,y,t) were neglected as very small. For the same reason in (3.b.5) we get

$$s_x(x,y,t+dt) = s_x(x,y,t) \triangleq s_x \, ,$$

$$s_y(x,y,t+dt) = s_y(x,y,t) \triangleq s_y \, ,$$

$$s_t(x,y,t+dt) = s_t(x,y,t) \triangleq s_t$$

and these are computable from the image frames at time $t - dt$ and t. Therefore (3.b.4) becomes:

$$\begin{aligned} g(\omega_x, \omega_y) &= \iint_S (s_x u^{t+dt} + s_y v^{t+dt} + s_t)^2 dx\, dy \\ &= \iint_S \Big(s_x(u^t + \omega_x xy - \omega_y(x^2+1)) \\ &\qquad + s_y(v^t + \omega_x(y^2+1) - \omega_y xy) + s_t \Big)^2 dx\, dy\,, \end{aligned} \tag{3.b.6}$$

where s_x, s_y, s_t, u^t and v^t are given from previous image data.

Then the appropriate ω_x, ω_y will be the ones that minimize a cost functional which encompasses the above requirements (3.b.3) and (3.b.6), like the following:

$$\begin{aligned} J(\omega_x, \omega_y, \lambda) &= g(\omega_x, \omega_y) + \lambda f(\omega_x, \omega_y) \\ &= \iint_S (s_x u^{t+dt} + s_y v^{t+dt} + s_t)^2 dx\, dy + \lambda \iint_B \left[(u^{t+dt})^2 + (v^{t+dt})^2 \right] dx\, dy \qquad (3.b.7) \\ &= \iint_S \Big(s_x(u^t + \omega_x xy - \omega_y(x^2+1)) + s_y(v^t + \omega_x(y^2+1) - \omega_y xy) + s_t \Big)^2 dx\, dy \\ &\qquad + \lambda \iint_B \Big[(u^t + \omega_x xy - \omega_y(x^2+1))^2 \\ &\qquad\qquad + (v^t + \omega_x(y^2+1) - \omega_y xy)^2 \Big] dx\, dy \qquad (3.b.8) \\ &= a_2\omega_x^2 + b_2\omega_y^2 + c_2\omega_x + d_2\omega_y + e_2\omega_x\omega_y + f_2 \\ &\qquad + \lambda(a_1\omega_x^2 + b_1\omega_y^2 + c_1\omega_x + d_1\omega_y + e_1\omega_x\omega_y + f_1) \qquad (3.b.9)\,, \end{aligned}$$

where

$$
\begin{aligned}
a_1 &= \iint_{\mathcal{B}} \left[x^2y^2 + (y^2+1)^2\right] dx\, dy\,, \\
b_1 &= \iint_{\mathcal{B}} \left[(x^2+1)^2 + x^2y^2\right] dx\, dy\,, \\
c_1 &= 2\iint_{\mathcal{B}} \left[u^t xy + v^t(y^2+1)\right] dx\, dy\,, \\
d_1 &= -2\iint_{\mathcal{B}} \left[u^t(x^2+1) + v^t xy\right] dx\, dy\,, \\
e_1 &= -2\iint_{\mathcal{B}} \left[xy(x^2+y^2+2)\right] dx\, dy\,, \\
f_1 &= \iint_{\mathcal{B}} \left[(u^t)^2 + (v^t)^2\right] dx\, dy\,,
\end{aligned}
\tag{3.b.10}
$$

$$
\begin{aligned}
a_2 &= \iint_{S} \left[s_x xy + s_y(y^2+1)\right]^2 dx\, dy\,, \\
b_2 &= \iint_{S} \left[s_x(x^2+1) + s_y xy\right]^2 dx\, dy\,, \\
c_2 &= 2\iint_{S} \left[s_x xy + s_y(y^2+1)\right]\left[s_x u^t + s_y v^t + s_t\right] dx\, dy\,, \\
d_2 &= -2\iint_{S} \left[s_x(x^2+1) + s_y xy\right]\left[s_x u^t + s_y v^t + s_t\right] dx\, dy\,, \\
e_2 &= -2\iint_{S} \left[s_x xy + s_y(y^2+1)\right]\left[s_x(x^2+1) + s_y xy\right] dx\, dy\,, \\
f_2 &= \iint_{S} \left[s_x u^t + s_y v^t + s_t\right]^2 dx\, dy\,.
\end{aligned}
$$

Provided we know the parameter λ, it is a simple optimization problem to express the optimal ω_x^*, ω_y^* as functions of λ and the parameters $a_1, \ldots, f_2$ as follows:

$$
\begin{aligned}
\omega_x^* &= \frac{(d_1e_1 - 2b_1c_1)\lambda^2 + (e_1d_2 + d_1e_2 - 2b_1c_2 - 2c_1b_2)\lambda + (d_2e_2 - 2b_2c_2)}{(4a_1b_1 + e_1^2)\lambda^2 + 2(2b_1a_2 + 2a_1b_2 + e_1e_2)\lambda + (4a_2b_2 + e_2^2)} \\
\omega_y^* &= \frac{(c_1e_1 - 2a_1d_1)\lambda^2 + (c_1e_2 + e_1c_2 - 2d_1a_2 - 2a_1d_2)\lambda + (e_2c_2 - 2a_2d_2)}{(4a_1b_1 + e_1^2)\lambda^2 + 2(2b_1a_2 + 2a_1b_2 + e_1e_2)\lambda + (4a_2b_2 + e_2^2)}\,.
\end{aligned}
\tag{3.b.11}
$$

The selection of λ can be thought of as our 'weighting' of the relative importance of the two constraints for ω_x, ω_y and its selection may be viewed as an engineering design decision. Nevertheless, there are ways to select an appropriate λ based on the data of the

problem. One is to consider the retrieval of ω_x, ω_y that correspond to the Tracking and Optical Flow Constraints (eq. (3.b.6) and (3.b.8)) as an equality constrained minimization problem, where we want to minimize $f(\omega_x, \omega_y)$ under the constraint $g(\omega_x, \omega_y) = 0$. Then λ can be seen as the Lagrange multiplier of the problem (Tracking Schema 3.1). Another way is to consider λ as a regularization parameter for the problem (3.b.7), reduce it in a penalized least-squares problem and apply Generalized Cross-Validation techniques in order to retrieve λ from the data. Then, the corresponding ω_x and ω_y can be found from (3.b.11) or from an equivalent formulation (Tracking Schema 3.2).

Tracking Schema 3.1 : Constrained Optimization Formulation

Problem :

$$\min_{\omega_x, \omega_y} \{ f(\omega_x, \omega_y) | g(\omega_x, \omega_y) = 0 \} . \qquad \text{(P)}$$

We formulate the Lagrangian of the problem:

$$\begin{aligned} F(\omega_x, \omega_y, \lambda) &= f(\omega_x, \omega_y) + \lambda g(\omega_x, \omega_y) \\ &= a_1\omega_x^2 + b_1\omega_y^2 + c_1\omega_x + d_1\omega_y + e_1\omega_x\omega_y + f_1 \\ &\quad + \lambda(a_2\omega_x^2 + b_2\omega_y^2 + c_2\omega_x + d_2\omega_y + e_2\omega_x\omega_y + f_2) \end{aligned} \quad , \qquad (3.b.12)$$

where $a_1, \ldots, f_2$ are defined in (3.b.10).

Theorem 3.1 :

i) (First-Order Necessary Conditions)

The *local extrema* of (P) are the solutions $\omega = (\omega_x, \omega_y)$ of the following system:

$$A\omega_x^2 + (B_1\omega_y + B_2)\omega_x + (\Gamma_1\omega_y^2 + \Gamma_2\omega_y + \Gamma_3) = 0 \qquad (3.b.13)$$

and

$$\alpha\omega_y^4 + \beta\omega_y^3 + \gamma\omega_y^2 + \delta\omega_y + \epsilon = 0 , \qquad (3.b.14)$$

where $A, B_1, B_2, \Gamma_1, \Gamma_2, \Gamma_3, \alpha, \beta, \gamma, \delta, \epsilon$ are functions of $a_1, \ldots, f_2$ and are defined in Appendix B. The equation (3.b.14) has at most 4 real solutions ω_y, which are specified in Appendix B. To each one of them correspond at most 2 real solutions ω_x of (3.b.13). Then

there exist at most 8 real solutions $\omega = (\omega_x, \omega_y)$ of the system and the corresponding Lagrange multiplier is:

$$\lambda(\omega) = -\frac{2a_1\omega_x + c_1 + e_1\omega_y}{2a_2\omega_x + c_2 + e_2\omega_y}.$$

ii) (Second-Order Sufficiency Conditions)

The *strict local minima* $\omega = (\omega_x, \omega_y)$ of (P) satisfy:

$$\left[(a_1 + \lambda(\omega)a_2\right]K^2(\omega) + \left[(e_1 + \lambda(\omega)e_2\right]K(\omega) + \left[b_1 + \lambda(\omega)b_2\right] > 0\,,$$

where

$$K(\omega) = -\frac{2b_2\omega_y + d_2 + e_2\omega_x}{2a_1\omega_x + c_2 + e_2\omega_y}.$$

iii) (Global minima)

Supposing that there exist l strict local minima ω^i of (P), with $i = 1,\ldots,l$ and $0 \leq l \leq 8$, the *global minima* ω^* are the ones that minimize $f(\omega^i)$.

Proof See Appendix B.

If more than one global minima exist, then additional criteria (e.g. minimization of $||\omega||^2$) can be used in order to compute the most suitable ω.

Tracking Schema 3.2 : Generalized Cross – Validation Formulation

Consider the ridge regression model

$$y = X\theta + \epsilon\,,$$

where y is an n-dimensional vector of measurements, θ a p-dimensional parameter vector, X an $n \times p$ design matrix and ϵ an n-dimensional error vector, with $E\epsilon = 0$ and $E\epsilon\epsilon^{\top} = \sigma^2 I$

Consider the problem of minimizing $\frac{1}{n}||y - X\theta||^2 + \lambda||\theta||^2$, over θ . Here $||.||$ indicates the Euclidean norm. There has been substantial interest in obtaining a good estimate of λ from the data and several methods have been proposed ([34]). One of the most successful ones has been the *Generalized Cross-Validation* technique (GCV), developed by Golub, Heath and Wahba [34] as a generalization of the Ordinary Cross-Validation technique (Allen [1]), which obtains an optimal λ as the minimizer of V(λ) given by:

$$V(\lambda) = \frac{(1/n)||(I - A(\lambda))y||^2}{[\frac{1}{n}Trace(I - A(\lambda))]^2},$$

where $A(\lambda) = X(X^\top X + n\lambda I)^{-1}X^\top$ and then the ridge estimate of θ is :

$$\hat{\theta}(\lambda) = (X^\top X + n\lambda I)^{-1}X^\top y\,.$$

Generalized Cross-Validation has been applied to a wide range of problems such as ridge regression (Golub, Heath and Wahba [34]), thin plate smoothing splines (Wahba and Wendelberger [75]) and ill-posed problems (Hilgers [36], O' Sullivan and Wahba [54]). The last class of applications is particularly important for Computer Vision, since many early vision problems are ill-posed (see Ch. 2.c.II). Based on this technique, GCVPACK, a package of routines for calculations in data analysis, was developed (Bates, Lindstrom, Wahba and Yandell [11]). The most general GCV calculation considered there is the solution of the *penalized least-squares* problem with an objective function of the form:

$$S_\lambda(\theta) \;=\; \frac{1}{n}||y - X\theta||^2 + \lambda\theta^\top\Sigma\theta\,, \qquad (3.b.15)$$

where θ is a p-dimensional parameter vector, y is an n-dimensional response vector, X is an $n \times p$ design matrix and Σ is a $p \times p$ positive semi-definite symmetric matrix, defining the smoothness penalty. The corresponding GCVPAK driver is the *subroutine dsnsm.*

Our problem (3.b.7) can be formulated as a penalized least-squares problem, if we consider the corresponding discretized problem. Let (u_{ij}, v_{ij}) denote the optical flow at a pixel (i,j) at time $t + dt$. Consider S as a $k \times k$ array of pixels and let $\mathcal{B} \subset \mathcal{S}$ be a neighborhood of the origin of the image-plane frame (fig. 3.1) . Then the discretized form of the cost functional (3.b.7)

$$J_\lambda \;=\; \iint_S (s_x u^{t+dt} + s_y v^{t+dt} + s_t)^2 dx\,dy + \lambda \iint_{\mathcal{B}} \left[(u^{t+dt})^2 + (v^{t+dt})^2\right] dx\,dy$$

is :

$$J_\lambda^D \;=\; \sum_S\sum \left((s_x(i,j)u_{ij} + s_y(i,j)v_{ij} + s_t(i,j)\right)^2 + \lambda \sum_{\mathcal{B}}\sum \left(u_{ij}^2 + v_{ij}^2\right).$$

Theorem 3.2 :

Consider $J_\lambda^D = \sum_S\sum ((s_x(i,j)u_{ij} + s_y(i,j)v_{ij} + s_t(i,j))^2 + \lambda \sum_B\sum (u_{ij}^2 + v_{ij}^2)$.

Define

$$\theta = \begin{pmatrix} u_{11} \\ v_{11} \\ u_{12} \\ v_{12} \\ \vdots \\ u_{kk} \\ v_{kk} \end{pmatrix}, \quad y = k \begin{pmatrix} -s_{t_{11}} \\ -s_{t_{12}} \\ \vdots \\ -s_{t_{kk}} \end{pmatrix}$$

and

$$X = k \begin{pmatrix} s_{x_{11}} & s_{y_{11}} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & s_{x_{12}} & s_{y_{12}} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & s_{x_{kk}} & s_{y_{kk}} \end{pmatrix},$$

where θ is a $2k^2$–vector, y is a k^2–vector and X is a $k^2 \times 2k^2$ matrix. Then J_λ^D can be formulated as

$$J_\lambda^D = \frac{1}{k^2}||y - X\theta||^2 + \lambda\theta^\top\Sigma\theta\,,$$

where Σ is the following $2k^2 \times 2k^2$ symmetric matrix :

$$\Sigma = \begin{pmatrix} \Sigma_{11} & 0 & \dots & 0 \\ 0 & \Sigma_{12} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma_{kk} \end{pmatrix},$$

with

$$\Sigma_{ij} = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{if } (i,j) \in B; \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, & \text{otherwise.} \end{cases}$$

If B is nonempty, then Σ is positive–definite.

Proof : See Appendix B.

Therefore, the GCV is readily applicable to the above problem with $p = 2k^2$ and $n = k^2$.

If we can find an estimate of the optical flow field at time $t + dt$, i.e. $(u_{ij}^{t+dt}, v_{ij}^{t+dt})$, from the above scheme, then, we can compute ω_x and ω_y from (3.b.1) and (3.b.2) applied at the origin or from an equivalent formulation like:

$$\omega_x \sum_{\mathcal{B}}\sum ij - \omega_y \sum_{\mathcal{B}}\sum (i^2+1) = \sum_{\mathcal{B}}\sum (u_{ij}^{t+dt} - u_{ij}^{t}) \qquad (3.b.24)$$

$$\omega_x \sum_{\mathcal{B}}\sum (j^2+1) - \omega_y \sum_{\mathcal{B}}\sum ij = \sum_{\mathcal{B}}\sum (v_{ij}^{t+dt} - v_{ij}^{t}) \qquad (3.b.25)$$

In both cases, a very simple non-homogeneous second-order linear system has to be solved.

Alternatively, since from the above scheme the regularization parameter λ can be estimated, we can compute ω_x and ω_y from (3.b.11).

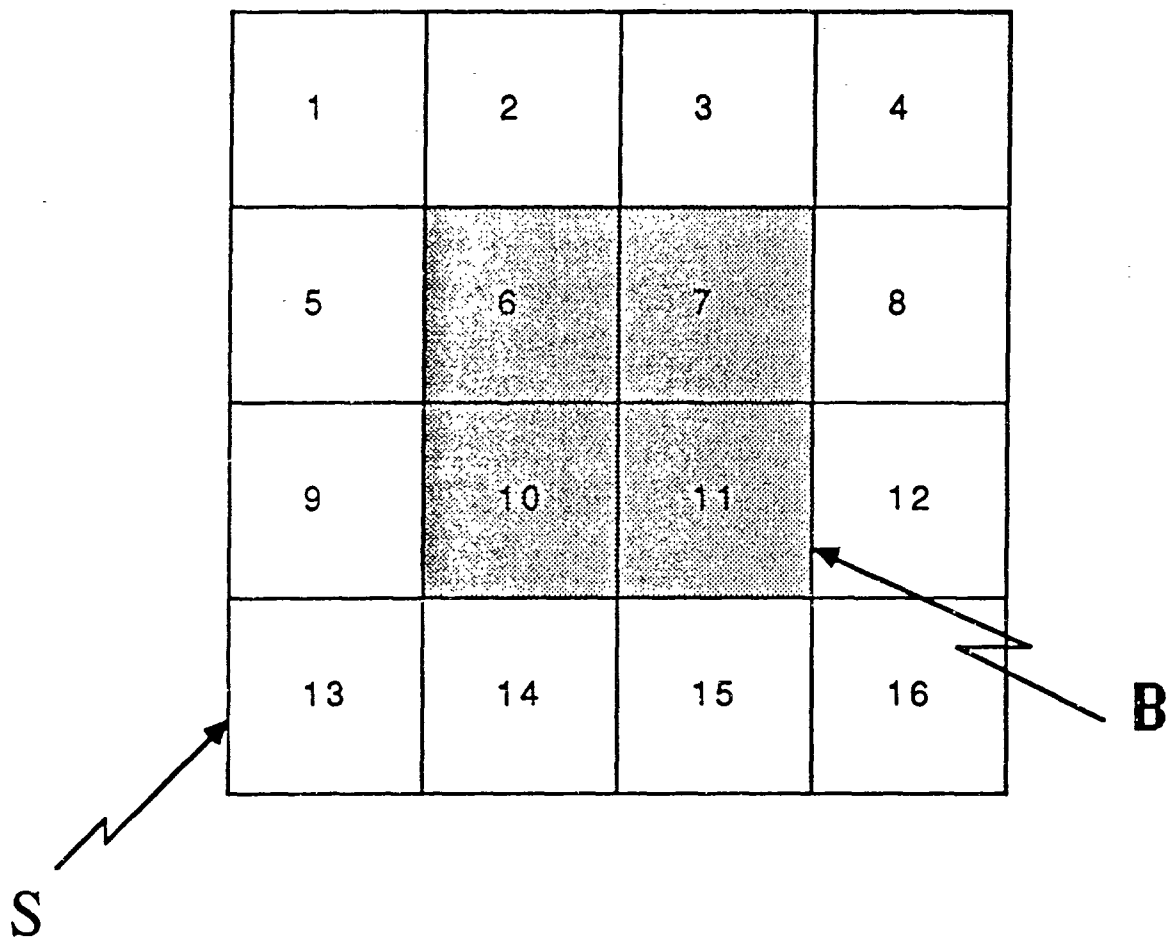


Fig. 3.1 : Simple 4 x 4 array of intensities

CHAPTER

FOUR

TRACKING USING KNOWLEDGE OF THE SHAPE OF THE TARGET

Certain important shortcomings handicap the general applicability of the methods of Ch. 3, as well as every method based on the assumption that we know the optical flow on the entire image. Assumptions like this base the whole tracking scheme on the solution of an ill–posed problem. These problems are solvable using the Tikhonov Regularization Theory, with quadratic stabilizing functionals, which implicitly impose global smoothness constraints on the possible solutions. Therefore discontinuities in the scene (e.g. occluding contours), present serious difficulties to standard regularization methods. The utilization of smoothness constrains as above in vision has often received *ad hoc* justification, that is implicitly based on computational convenience, rather than theoretical consideration, but even though there exists some excellent research on *discontinuous regularization* (see Ch. 3.a) the problem seems to be far from solved. Additional errors will be introduced in the previous tracking scheme, even in the case of a simple scene, from the numerical differentiation of the image intensity function $s(x, y, t)$, which is an ill–posed problem by itself, from our model of the optical flow at time t+dt, as well as from the estimation of the regularization parameter using Generalized Cross–Validation.

We attempted to bypass these problems by making a sensible assumption : That we know the shape of the moving target. Our theory of tracking an object whose shape is known, proceeds as follows:

First, we compute the three–dimensional motion parameters of the object (direction of translation and angular velocity). An estimate of the optical flow can then be found, which depends only on information available from the image, where we used the shape information in order to eliminate the depth terms of eq. (2.c.5) and (2.c.6). Based on this and on the theory of linear features (Ch. 4.a), we can estimate the target motion parameters, by solving a system of linear equations (Ch. 4.b). The system does not use point correspondence, only global information from the picture.

Second, from the Tracking Constraint, we formulate a cost functional, whose minimizers are the camera angular velocities ω_x and ω_y, which are necessary in order for the camera to track the moving target. Our estimate of the target motion parameters computed in the first step is used and we can solve this minimization problem for ω_x and ω_y (Ch. 4.c) .

Simulation results for the above algorithms are presented in Ch. 4.d.

4.a) The Theory of Linear Features

Linear Features of an image are sets of functionals of the type

$$F = \{f_i(t) | f_i(t) = \iint_D s(x,y,t)\mu_i(x,y)dx\,dy\ ,\ i = 1,\ldots,n\}\ ,$$

where s is the image intensity at a point (x,y) of a subset D of the image at time t and $\mu_i : I\!R^2 \longrightarrow I\!R$ is the *measuring function.*

Linear features have been used extensively in Pattern Recognition for the study of pattern classification and the invariants of perception. The objective of pattern recognition is to derive from an image a description of what it depicts. If the desired description is just a classification of the image as one of a set of prespecified classes (e.g. classify the picture of a single printed character as 'A','B',...), then this can be done by measuring some global properties of the image. If, on the other hand, the description should involve relations between objects on the image, then accurate location of the objects (image segmentation) should precede the measurement of their properties. In the first case (pattern classification), the problem is to find a function that maps a set of patterns into the desired classes. This is done in three steps : First some preprocessing of the image is done in order to reduce the noise, second, feature extraction is performed and the image is mapped into an n–dimensional real features vector and third, classification

of the image from the position of this vector in $I\!R^n$. In Feature Extraction the main problem is how to select an adequate set of features for the given classification task. A general approach is to use knowledge about the structure of patterns and the definition of classes.

In the human visual system, objects are recognized, no matter where they are located on the field of view, what their orientation is, their apparent size, their motion or their projective and binocular distortion. Physiological and neuroanatomical evidence suggests that after an initial visual processing, an abstraction of the visual scene is constructed, which, even though it carries all information the shapes and contours of the image can provide, is invariant to object transformations like the above. This abstract image on the striate cortex is called the *visual manifold* (Hoffman [39]). It is possible to formulate mathematically the perception invariances as Lie groups of transformations over the visual manifold and specify those invariant transformations (Hoffman [39], Foster [26], Blaivas [15]).

Features of patterns that remain invariant under specific transformations of the pattern were studied and linear features in particular have received much attention. *Moment invariants*, i.e. linear features with $\mu_i(x,y) = x^p y^q$, $p,q = 0,1,2,\ldots$ were introduced by Hu [42] for the recognition of planar geometric figures. He established their invariance under 2-D translation, scaling and rotation using algebraic invariance theory. Moment invariants were used by Teague [67] for the reconstruction of characters and were applied to aircraft identification problems by Dudani et al. [25]. In cases where the patterns can be normalized, e.g. in character recognition systems, the Fourier coefficients of an image function, i.e. linear features with $\mu_i(x,y) = e^{\alpha_i x + \beta_i y}$, $\alpha_i, \beta_i \in \mathbb{C}$ were also considered (Rosenfeld [60], Rosenfeld and Kak [61], Wallace and Mitchell [76]).

A connection between these two lines of research was attempted by Amari [4], [5] where the problem of determining the invariant transformations from changes in the features of the pattern under transformation is considered for the case of linear features of planar visual patterns. He uses group theoretical methods to prove the invariance of moment features under the group of planar affine transformations.

Our purpose for introducing linear features in the study of the 3-D motion estimation problem is not to classify or reconstruct a pattern of some kind, but is related to the errors introduced to the Optical Flow Constraint equation when it is used to process discretized information. In principle, the 3-D motion parameters could be recovered (up

to depth) by a formulation like the following: From $s_x u + s_y v + s_t = 0$ and (2.c.7) and (2.c.8) we get for every point (x,y) :

$$s_x\left[\frac{T_1}{Z} - x\frac{T_3}{Z} - \omega_1(y^2+1) + \omega_2 xy + \omega_3 x\right] + s_y\left[\frac{T_2}{Z} - y\frac{T_3}{Z} - \omega_1 xy + \omega_2(x^2+1) - \omega_3 y\right] + s_t = 0\,.$$

Then :

$$s_x\frac{T_1}{Z} + s_y\frac{T_2}{Z} - (s_x x + s_y y)\frac{T_3}{Z} - [s_x(y^2+1) s_y xy]\omega_1$$
$$+ [s_x xy + s_y(x^2+1)]\omega_2 + [s_x x - s_y y]\omega_3 = -s_t\,.$$

Given enough points (x,y) we will take a system of linear equations $Ax = y$, where $x = (\frac{T_1}{Z}, \frac{T_2}{Z}, \frac{T_3}{Z}, \omega_1, \omega_2, \omega_3)$ and solve it to derive the motion parameters. The problem is that A depends on the spatiotemporal derivatives of s at each point (x, y) and, since numerical differentiation is an ill-posed problem, severe errors will be introduced if we attempt to evaluate numerically these derivatives.

Therefore an algorithm is presented in the next section, where, using linear features to collect global information from an area of the image, the computation of the spatial derivatives of the image is no longer necessary (Ito and Aloimonos [44]). A problem common in this approach and in the use of linear features in Pattern Recognition is how to select the most appropriate measuring functions and what the length of the linear feature vector should be, so that the characterization of the image by this vector be "rich" enough in information. This point will be discussed further in Ch. 4.d.

4.b) Recovery of the Three – Dimensional Motion

Suppose that an object exists in $I\!R^3$, with its surface given as a function Z=g(X,Y) in the camera coordinate system $OXYZ$. The following analysis is done with respect to the camera coordinate frame. It can be readily extended to the world frame using the relationships between the two frames of Ch. 2.b. A point A on the surface of the object is projected to the point P_A of the image plane. Suppose further that A has coordinates $\vec{X} = (X, Y, Z)$ with respect to the camera coordinate system at time t and that P_A has coordinates $\vec{x} = (x, y)$ with respect to the image–plane frame at the same time instant.

Theorem 4.1 :

The optical flow $\dot{\vec{x}} = (u, v)$ generated from the motion of a rigid target relative to our camera is a linear function of the target motion parameters of the form :

$$\dot{\vec{x}} = \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^{6} m_k \vec{u}_k(\vec{x}), \tag{4.b.1}$$

where m_k are the motion parameters (direction of translation and rotation) and $\vec{u}_k$ are known functions of the shape, the projection function and the retinal coordinates.

Proof :

As we saw in Ch. 2.a if $P : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the projection function, then :

$$\vec{x} = P(\vec{X}) = \begin{pmatrix} P_1(\vec{X}) \\ P_2(\vec{X}) \end{pmatrix} \tag{4.b.2}$$

Suppose that the object in view performs a rigid motion. In Ch. 2.b we saw that each point A will move with a velocity V, which is a function of its position $\vec{X}$ with respect to the camera coordinate system. More precisely from Corollary 2.1 :

$$\frac{d\vec{X}}{dt} \triangleq \vec{V}(\vec{X}) = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = T + \hat{\omega}\vec{X} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} + \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{4.b.3}$$

where T is the translational velocity of the center W of the target frame and ω the angular velocity of the target. The velocity $\vec{V}$ can be written as:

$$\begin{aligned} \vec{V}(\vec{X}) &= T_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + T_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + T_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \omega_1 \begin{pmatrix} 0 \\ -Z \\ Y \end{pmatrix} + \omega_2 \begin{pmatrix} Z \\ 0 \\ -X \end{pmatrix} + \omega_3 \begin{pmatrix} -Y \\ X \\ 0 \end{pmatrix} \\ &= \sum_{k=1}^{6} m_k V_k(\vec{X}), \end{aligned} \tag{4.b.4}$$

where m_i are the target motion parameters defined as follows:

$$m_1 = T_1, m_2 = T_2, m_3 = T_3, m_4 = \omega_1, m_5 = \omega_2, m_6 = \omega_3$$

and

$$V_1(\vec{X}) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, V_2(\vec{X}) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, V_3(\vec{X}) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$V_4(\vec{X}) = \begin{pmatrix} 0 \\ -Z \\ Y \end{pmatrix}, V_5(\vec{X}) = \begin{pmatrix} Z \\ 0 \\ -X \end{pmatrix}, V_6(\vec{X}) = \begin{pmatrix} -Y \\ X \\ 0 \end{pmatrix}.$$

As the target moves in $I\!R^3$, its image also moves on the image plane. The projection P_A of the point A, with image plane coordinates $\vec{x} = P(\vec{X})$, moves with velocity $\dot{\vec{x}}$, which is the optical flow at this point. From (4.b.2) :

$$\dot{\vec{x}}(\vec{X}) = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \triangleq \begin{pmatrix} u \\ v \end{pmatrix} = \frac{d}{dt}P(\vec{X}) = \frac{\partial P}{\partial \vec{X}}(\vec{X})\vec{V}(\vec{X}) = \begin{pmatrix} \frac{\partial P_1}{\partial X} & \frac{\partial P_1}{\partial Y} & \frac{\partial P_1}{\partial Z} \\ \frac{\partial P_2}{\partial X} & \frac{\partial P_2}{\partial Y} & \frac{\partial P_2}{\partial Z} \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix}. \tag{4.b.5}$$

Moreover, from the projection function $\vec{x} = P(\vec{X})$ we can derive the inverse projection mapping P_g^{-1} from the image to the surface, i.e. $\vec{X} = P_g^{-1}(\vec{x})$, which is known up to a factor (depth), since the shape Z=g(X,Y) of the target is known. We can then consider $\frac{\partial P}{\partial \vec{X}}$ and $\vec{V}$ and thus the optical flow $\dot{\vec{x}}$ itself as functions of the retinal coordinates $\vec{x}$. Then:

$$\frac{\partial P}{\partial \vec{X}} = \frac{\partial P}{\partial \vec{X}}(P_g^{-1}(\vec{x})) \, , \; \vec{V} = \vec{V}(P_g^{-1}(\vec{x})) \, ,$$

and

$$\dot{\vec{x}} = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{\partial P}{\partial \vec{X}}(P_g^{-1}(\vec{x}))\, \vec{V}(P_g^{-1}(\vec{x})) \, .$$

where for $A \subset I\!R^2$ and $B \subset I\!R^3$ we have $P_g^{-1} : A \rightarrow B$ the inverse transformation of P. The function P is surjective, thus the inverse function exists, it may not be injective though and then the inverse transformation is not unique, but physical intuition may solve in most cases this non-uniqueness.

From (4.b.4) and (4.b.5) we have:

$$\dot{\vec{x}} = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{\partial P}{\partial \vec{X}} \sum_{k=1}^{6} m_k V_k(\vec{X}) = \sum_{k=1}^{6} m_k \frac{\partial P}{\partial \vec{X}}(P_g^{-1}(\vec{x}))V_k(P_g^{-1}(\vec{x})) = \sum_{k=1}^{6} m_k \begin{pmatrix} u_{k_1}(\vec{x}) \\ u_{k_2}(\vec{x}) \end{pmatrix}$$

where we defined :

$$\vec{u}_k(\vec{x}) \triangleq \begin{pmatrix} u_{k_1}(\vec{x}) \\ u_{k_2}(\vec{x}) \end{pmatrix} = \frac{\partial P}{\partial \vec{X}}(P_g^{-1}(\vec{x}))V_k(P_g^{-1}(\vec{x})) \, . \tag{4.b.6}$$

Obviously $\vec{u}_k(\vec{x})$ depends only on the shape of the target and is known. Then:

$$\dot{\vec{x}} = \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^{6} m_k \vec{u}_k(\vec{x}), \tag{4.b.7}$$

and we have expressed the optical flow at a point of the image as a function of the shape of the target, its motion parameters m_i and the retinal coordinates. Moreover it is given as a linear combination of the motion parameters.

Example : The Planar Case

The above analysis is particularly simple in the planar case. We will show how (4.b.7) transforms in this case under perspective projection. Suppose we want to track a point W on the plane and suppose that this point translates with velocity T and the plane rotates with angular velocity ω with respect to the camera frame.

The projection function is:

$$\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} = P(\vec{X}) = \begin{pmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \end{pmatrix} \tag{4.b.8}$$

Consider the plane $Z = g(X,Y) = pX + qY + c$. The shape parameters p and q are supposed to be known, but not the depth factor c. It is simple to find the inverse projection transformation $P_g^{-1}(\vec{x})$:

$$\vec{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = P_g^{-1}(\vec{x}) = \frac{c}{f - px - qy} \begin{pmatrix} x \\ y \\ f \end{pmatrix} \tag{4.b.9}$$

Then from (4.b.8) and (4.b.9):

$$\frac{\partial P}{\partial \vec{X}}(\vec{X}) = \begin{pmatrix} \frac{f}{Z} & 0 & -\frac{fX}{Z^2} \\ 0 & \frac{f}{Z} & -\frac{fY}{Z^2} \end{pmatrix}$$

$$\Rightarrow \frac{\partial P}{\partial \vec{X}}(P_g^{-1}(\vec{x})) = \frac{f - px - qy}{c} \begin{pmatrix} 1 & 0 & -\frac{x}{f} \\ 0 & 1 & -\frac{y}{f} \end{pmatrix}$$

It is easy to prove from (4.b.3) and (4.b.9) that:

$$V(P_g^{-1}(\vec{x})) = \frac{c}{f - px - qy} \begin{pmatrix} \frac{T_1}{c}(f - px - qy) + \omega_2 f - \omega_3 y \\ \frac{T_2}{c}(f - px - qy) + \omega_3 x - \omega_1 f \\ \frac{T_3}{c}(f - px - qy) + \omega_1 y - \omega_2 x \end{pmatrix}$$

Defining $m_1 = \frac{T_1}{c}$, $m_2 = \frac{T_2}{c}$, $m_3 = \frac{T_3}{c}$, $m_4 = \omega_1$, $m_5 = \omega_2$, $m_6 = \omega_3$ and

$$\vec{V}_1(\vec{x}) = \frac{c}{f - px - qy}\begin{pmatrix} f - px - qy \\ 0 \\ 0 \end{pmatrix} , \vec{V}_2(\vec{x}) = \frac{c}{f - px - qy}\begin{pmatrix} 0 \\ f - px - qy \\ 0 \end{pmatrix} ,$$

$$\vec{V}_3(\vec{x}) = \frac{c}{f - px - qy}\begin{pmatrix} 0 \\ 0 \\ f - px - qy \end{pmatrix} , \vec{V}_4(\vec{x}) = \frac{c}{f - px - qy}\begin{pmatrix} 0 \\ -f \\ y \end{pmatrix} ,$$

$$\vec{V}_5(\vec{x}) = \frac{c}{f - px - qy}\begin{pmatrix} f \\ 0 \\ -x \end{pmatrix} , \vec{V}_6(\vec{x}) = \frac{c}{f - px - qy}\begin{pmatrix} -y \\ x \\ 0 \end{pmatrix} ,$$

we get from (4.b.6):

$$\vec{u}_1(\vec{x}) = \begin{pmatrix} f - px - qy \\ 0 \end{pmatrix} , \vec{u}_2(\vec{x}) = \begin{pmatrix} 0 \\ f - px - qy \end{pmatrix} , \vec{u}_3(\vec{x}) = \begin{pmatrix} -\frac{x(f-px-qy)}{f} \\ -\frac{y(f-px-qy)}{f} \end{pmatrix} ,$$

$$\vec{u}_4(\vec{x}) = \begin{pmatrix} -\frac{xy}{f} \\ -\frac{(f^2+y^2)}{f} \end{pmatrix} , \vec{u}_5(\vec{x}) = \begin{pmatrix} \frac{(f^2+x^2)}{f} \\ \frac{xy}{f} \end{pmatrix} , \vec{u}_6(\vec{x}) = \begin{pmatrix} -y \\ x \end{pmatrix} .$$

Then from (4.b.7):

$$\dot{\vec{x}} = \sum_{k=1}^{6} m_k \vec{u}_k(\vec{x}) ,$$

with the parameters $\vec{u}_k$ depending on the shape parameters of the target and on retinal coordinates. Moreover the optical flow (u, v) is expressed as linear combination of the motion parameters m_i.

The above scheme requires knowledge of the shape parameters (p,q) of the plane at each time instant. This may seem a restrictive assumption, but anatomical, physiological and psychophysical evidence suggests that early image analysis in human vision is indeed performed by a variety of distinct systems in the brain. In particular, psychophysical data exist (Savoy [62]) about the separation of the Orientation and Motion Subsystems of Vision, which may act surprisingly independently, except that the output of one may act as an input to the other. An important question is then how the brain manages to synthesize, out of relatively isolated and independent components extracted from an image under the analysis process, an internal representation that seems to capture the

salient characteristics of the visual stimulus. An answer could be provided using the above scheme .

From the Computer Vision point of view, several methods have been proposed for the computation of orientation and in particular the orientation of a plane. Those methods lie in two broad categories, the *shape from X* techniques, which attempt to recover surface orientation based on physical constraints and natural assumptions and the *model-based matching techniques*, which attempt to recover 3-D information by comparing 2-D image features with a known set of 3-D object models. In techniques of the first kind, various clues can be used for the computation of shape, like occluding contours, textural variations in the image and shading. An important alternative is the Photometric Stereo Algorithm (Horn[40]), when orientation of surface patches can be recovered from a number of images taken under different lighting conditions of the scene, provided the reflectance map is known. Many of the above approaches lead to ill-posed problems and various schemes have been provided for their solution, such as Regulatization and Active Vision (see Ch. 2.c.II).

Estimation of the Motion Parameters $\mathbf{m_i}$

The problem now is to compute the motion parameters $m_i, i = 1, \ldots, 6$ of (4.b.1), without actually computing the optical flow. For this purpose, we use the Theory of Linear Features presented in Ch. 4.a.

Consider a set F of linear features over a neighborhood $\mathcal{B}(0, \epsilon)$ of the origin of the image plane $F(t) = \{f_i(t) = \iint_{\mathcal{B}} s(x, y, t))\mu_i(x, y)dx\,dy,\ i = 1, \ldots, n\}$. From the Optical Flow Constraint Equation we have : $s_x u + s_y v + s_t = 0$ or $\frac{\partial s}{\partial t} = -\dot{\vec{x}}^T \nabla s$. Consider the temporal derivative of the i-th linear feature:

$$\begin{aligned} \dot{f}_i &= \iint_{\mathcal{B}} \frac{\partial s}{\partial t}\mu_i dx\,dy \\ &= -\iint_{\mathcal{B}} \mu_i(\dot{\vec{x}}^T \nabla s)dx\,dy \\ &\overset{(4.b.7)}{=} -\sum_{k=1}^{6} m_k \iint_{\mathcal{B}} \mu_i(u_{k_1} s_x + u_{k_2} s_y)dx\,dy\,. \end{aligned} \tag{4.b.10}$$

Defining $h_{ik} = -\iint_{\mathcal{B}} \mu_i(u_{k_1} s_x + u_{k_2} s_y)dx\,dy$, we get $\dot{f}_i = \sum_{k=1}^{6} m_k h_{ik}$ and then:

$$\mathbf{H}\vec{m} = \dot{\vec{f}}, \tag{4.b.11}$$

where $\mathbf{H}$ is the $n \times 6$ matrix of the coefficients h_{ik} and $\vec{m} = (m_1, \ldots, m_6)^T$. The vector of derivatives of linear features $\dot{\vec{f}}$ and the matrix $\mathbf{H}$ depend on measurable quantities. We want to solve the linear system (4.b.11) and obtain the motion parameter vector $\vec{m}$. Since the number n of linear features will be greater than 6, the system will be inconsistent in the general case. We can select the linear features such that $rank(\mathbf{H}) = 6$. Therefore, a least-squares approximate solution becomes necessary.

In our case, the system (4.b.11) has the unique minimal norm solution :

$$\vec{m} = \mathbf{H}^{\dagger}\dot{\vec{f}}, \tag{4.b.14}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose inverse of $\mathbf{H}$, which can be computed either directly from the formula $\mathbf{H}^{\dagger} = (\mathbf{H}^{\top}\mathbf{H})^{-1}\mathbf{H}^{\top}$ if $\mathbf{H}$ has $rank(\mathbf{H}) = m$, or using Greville's Algorithm, an iterative procedure that computes $\mathbf{H}^{\dagger}$ in m iterations. At the k-th iteration it computes $\mathbf{H}_{\mathbf{k}}^{\dagger}$, where $\mathbf{H_k}$ is the submatrix of $\mathbf{H}$ consisting of its first k columns. The advantage of the second method is that we don't have to check the rank of $\mathbf{H}$ and we avoid the enormous analytic expressions created e.g. from *macsyma* for $(\mathbf{H}^{\top}\mathbf{H})^{-1}\mathbf{H}^{\top}$. Greville's Algorithm is very well suited for our case, since it will compute $\mathbf{H}^{\dagger}$ in $m = 6$ steps, no matter how long the linear features vector is. For the definition of the Moore–Penrose inverse , Greville's Algorithm and related results, see Appendix C.

In our target motion parameter estimation scheme no explicit calculation of the optical flow field is needed, as we have seen up to this point. In Ch. 4.c we will see that neither is it needed for the tracking schemes developed there. Moreover, the only calculation involving the derivatives of the image intensity function s is done in order to obtain the parameters h_{ik}. Spatial differentiation of s is needed there, but, since numerical differentiation is an ill-posed problem, we attempted to bypass it, by using integration by parts in (4.b.10) .

Supposing that the area $\mathcal{B}$ is a rectangle $\mathcal{B} = [a, b] \times [c, d]$ in image plane coordinates,

we have:

$$h_{ik} = -\int_a^b\int_c^d \mu_i(u_{k_1}s_x + u_{k_2}s_y)dx\,dy$$

$$= -\int_c^d [\mu_i(b,y)u_{k_1}(b,y)s(b,y) - \mu_i(a,y)u_{k_1}(a,y)s(a,y)]\,dy$$

$$-\int_a^b [\mu_i(x,d)u_{k_2}(x,d)s(x,d) - \mu_i(x,c)u_{k_2}(x,c)s(x,c)]\,dx$$

$$+\int_a^b\int_c^d s(x,y)[\frac{\partial}{\partial x}(\mu_i(x,y)u_{k_1}(x,y)) + \frac{\partial}{\partial y}(\mu_i(x,y)u_{k_2}(x,y))]dx\,dy$$

. Therefore, spatial differentiation of s was replaced by differentiation of μ_i, u_{k_1} and u_{k_2}, which can be done analytically.

Due to the problem degeneracy (see Ch. 2.b.III), from the above process, as well as from every method for the recovery of motion from monocular data, we can only recover the rotation and the direction of translation, due to the unknown depth scale. This is obvious from our example, where we defined $m_i = \frac{T_i}{c}, i = 1,2,3$, where c is essentially a measure of depth.

4.c) Tracking

From the algorithm presented in Ch. 4.b we get an estimate of the motion parameters of our target. It remains to see how we use it in order to fulfill the Tracking Condition presented in Ch. 2.c.III. We derived two schemata for tracking, one where we use our knowledge of the shape of the target (Tracking Schema 4.1) and one where we don't (Tracking Schema 4.2). In both cases we use the Tracking Constraint and reduce the problem of specifying the camera angular velocities to an unconstrained optimization problem, namely that of minimizing a cost functional of the form:

$$J(\omega_x,\omega_y) = \alpha\omega_x^2 + \beta\omega_y^2 + \gamma\omega_x + \delta\omega_y + \epsilon\omega_x\omega_y + \zeta\,. \qquad (4.c.1)$$

Lemma 4.1 :

Consider a cost functional of the form :

$$J(x,y) = \alpha x^2 + \beta y^2 + \gamma x + \delta y + \epsilon xy + \zeta\,,$$

with $\alpha, \beta \geq 0$, not simultaneously zero and $4\alpha\beta - \epsilon^2 \neq 0$. Such a cost functional has a unique global minimizer $\mathbf{x}^* = (x^*, y^*)$ of the form:

$$x^* = \frac{\delta\epsilon - 2\beta\gamma}{4\alpha\beta - \epsilon^2}, \tag{4.c.2}$$

$$y^* = \frac{\gamma\epsilon - 2\alpha\delta}{4\alpha\beta - \epsilon^2}. \tag{4.c.3}$$

Proof See Appendix C.

Supposing that the target motion induces optical flow $(u^t(x,y), v^t(x,y))$ at the point (x,y) of the image plane at time t, while the camera motion induces optical flow $(u_{CAM}(x,y), v_{CAM}(x,y))$, the Tracking Constraint of Ch. 2.c.III says that the ω_x, ω_y we seek are the minimizers of:

$$J(\omega_x, \omega_y) = \iint_B \left[\left(u^t(x,y) + u_{CAM}(x,y)\right)^2 + \left(v^t(x,y) + v_{CAM}(x,y)\right)^2 \right] dx\, dy \tag{4.c.6}$$

In the sequel, we attempt to bring (4.c.6) in the form (4.c.1) using two different approaches.

Tracking Schema 4.1 : Using Shape Information

Under the assumption that we know the shape of our target, we derived an estimate of the target motion parameters $m_i, i = 1, \ldots, 6$ (Ch. 4.b). We derived also an expression of the *target-induced* optical flow of the form (eq. (4.b.7)) :

$$\begin{pmatrix} u^t(x,y) \\ v^t(x,y) \end{pmatrix} = \sum_{i=1}^{6} m_i \begin{pmatrix} u_{i1}(x,y) \\ u_{i2}(x,y) \end{pmatrix}, \tag{4.c.7}$$

where u_{ij} are *known* functions of the shape of the target (see example of planar case in Ch. 4.b) and m_i the known estimates of the target motion parameters (where (m_1, m_2, m_3) is the direction of target translation and (m_4, m_5, m_6) is the target rotation). Similarly, the *camera-induced* optical flow will be given by an expression like (4.c.7) with $m_1 = 0, m_2 = 0, m_3 = 0, m_4 = -\omega_x, m_5 = -\omega_y$ and $m_6 = 0$:

$$\begin{pmatrix} u_{CAM}(x,y) \\ v_{CAM}(x,y) \end{pmatrix} = -\omega_x \begin{pmatrix} u_{41}(x,y) \\ u_{42}(x,y) \end{pmatrix} - \omega_y \begin{pmatrix} u_{51}(x,y) \\ u_{52}(x,y) \end{pmatrix}, \tag{4.c.8}$$

where $u_{41}, u_{42}, u_{51}, u_{52}$ are known functions and ω_x, ω_y are the unknown camera rotation parameters. By substituting $u^t, v^t, u_{CAM}, v_{CAM}$ from (4.c.7) and (4.c.8) to (4.c.6), the only unknowns will be ω_x and ω_y and J can be brought to the form :

$$J(\omega_x, \omega_y) = \alpha\omega_x^2 + \beta\omega_y^2 + \gamma\omega_x + \delta\omega_y + \epsilon\omega_x\omega_y + \zeta\,, \tag{4.c.9}$$

where

$$\alpha = \iint_{\mathcal{B}} [u_{41}^2(x,y) + u_{42}^2(x,y)]dx\,dy\,,$$

$$\beta = \iint_{\mathcal{B}} [u_{51}^2(x,y) + u_{52}^2(x,y)]dx\,dy\,,$$

$$\gamma = -2\iint_{\mathcal{B}} [u^t(x,y)u_{41}(x,y) + v^t(x,y)u_{42}(x,y)]dx\,dy\,,$$

$$\delta = -2\iint_{\mathcal{B}} [u^t(x,y)u_{51}(x,y) + v^t(x,y)u_{52}(x,y)]dx\,dy\,,$$

$$\epsilon = 2\iint_{\mathcal{B}} [u_{41}(x,y)u_{51}(x,y) + u_{42}(x,y)u_{52}(x,y)]dx\,dy\,,$$

$$\zeta = \iint_{\mathcal{B}} [(u^t(x,y))^2 + (v^t(x,y))^2]dx\,dy\,,$$

where $u^t(x,y) = \sum_{i=1}^{6} m_i u_{i1}(x,y)$ and $v^t(x,y) = \sum_{i=1}^{6} m_i u_{i2}(x,y)$. Obviously $\alpha, \beta \geq 0$. Simulation results show that they are not simultaneously zero and $4\alpha\beta - \epsilon^2 \neq 0$ (see also the example below). Then from Lemma 4.1, ω_x^*, ω_y^* are given from (4.c.2) and (4.c.3):

$$\omega_x^* = \frac{\delta\epsilon - 2\beta\gamma}{4\alpha\beta - \epsilon^2} \qquad \text{and} \qquad \omega_y^* = \frac{\gamma\epsilon - 2\alpha\delta}{4\alpha\beta - \epsilon^2}\ .$$

Example : The Planar Case

Consider the same planar case as before and let apply Tracking Schema 4.1. We expect this tracking scheme to produce a motion of the camera in the direction of that of the target, if our tracking goal is to be accomplished. Suppose that the point W is near the center of the image–plane coordinate frame and is moving with a translational velocity T and the target rotates with angular velocity ω and assume that these are exactly known from the estimation part of our algorithm. Then, in the small area $\mathcal{B}$ that we consider around the origin of the image plane, we will have $x \approx 0$ and $y \approx 0$. Therefore, the expressions we found for $u_1, \ldots, u_6$, will be as follows:

$$u_1 \approx \begin{pmatrix}1\\0\end{pmatrix}, u_2 \approx \begin{pmatrix}0\\1\end{pmatrix}, u_3 \approx \begin{pmatrix}0\\0\end{pmatrix}, u_4 \approx \begin{pmatrix}0\\-1\end{pmatrix}, u_5 \approx \begin{pmatrix}1\\0\end{pmatrix}, u_6 \approx \begin{pmatrix}0\\0\end{pmatrix}.$$

Therefore the optical flow in this area will be:

$$u^t(x,y) \approx \frac{T_1}{c} + \omega_2 \text{ and } v^t(x,y) \approx \frac{T_2}{c} - \omega_1$$

Defining $A \triangleq \iint_B dx\, dy$, the parameters of $J(\omega_x, \omega_y)$ will become:

$$\alpha \approx A\ ,\ \beta \approx A\ ,\ \gamma \approx 2(\frac{T_2}{c} - \omega_1)A\ ,\ \delta \approx -2(\frac{T_1}{c} + \omega_2)A\ ,\ \epsilon \approx 0\ .$$

Thus, from (4.c.2) and (4.c.3) we get:

$$\omega_x^* \approx -\frac{\gamma}{2\alpha} \approx -\frac{T_2}{c} + \omega_1$$

and

$$\omega_y^* \approx -\frac{\delta}{2\beta} \approx \frac{T_1}{c} + \omega_2$$

From fig. 2.2 we can see that this is exactly the motion our camera should perform in order to track the target. In Ch. 4.d we present simulation results that show how this scheme is actually working in this case.

Tracking Schema 4.2 : Without Using Shape Information

A different approach can be considered and lead to a more general tracking scheme, where the requirement for knowledge of the shape of the target is replaced by an assumption for well-behavedness of the shape in the neighborhood $\mathcal{B}_W$ of the feature P_W that we attempt to track. The target motion parameters (direction of translation $\frac{T_1}{T_3}, \frac{T_2}{T_3}$ and angular velocity $\omega_1, \omega_2, \omega_3$) are supposed to have been estimated in the first part of the algorithm. In the above expression (4.c.6), the target-motion induced optical flow is given from equations (2.c.5) and (2.c.6) and the camera-induced optical flow is given from equations (2.c.9) and (2.c.10), i.e. for f=1 :

$$u^t(x,y) = \frac{T_1}{T_3}\frac{T_3}{Z} - x\frac{T_3}{Z} - \omega_1 xy + \omega_2(x^2+1) - \omega_3 y, \qquad (4.c.10)$$

$$v^t(x,y) = \frac{T_2}{T_3}\frac{T_3}{Z} - y\frac{T_3}{Z} - \omega_1(y^2+1) + \omega_2 xy + \omega_3 x, \qquad (4.c.11)$$

$$u_{CAM}(x,y) = \omega_x xy - \omega_y(x^2+1), \qquad (4.c.12)$$

$$v_{CAM}(x,y) = \omega_x(y^2+1) - \omega_y xy. \qquad (4.c.13)$$

In the above equations, the unknowns are $\frac{T_3}{Z_W}, \omega_x, \omega_y$. Suppose now that the target surface is smooth enough around W, so that the depth Z corresponding to all points projected in $\mathcal{B}_W$ is almost the same, i.e. Z_W. Then for all points $(x,y) \in \mathcal{B}_W$, we have:

$$u^t(x,y) = \frac{T_1}{T_3}\frac{T_3}{Z_W} - x\frac{T_3}{Z_W} - \omega_1 xy + \omega_2(x^2+1) - \omega_3 y, \qquad (4.c.14)$$

$$v^t(x,y) = \frac{T_2}{T_3}\frac{T_3}{Z_W} - y\frac{T_3}{Z_W} - \omega_1(y^2+1) + \omega_2 xy + \omega_3 x, \qquad (4.c.15)$$

From the Optical Flow Constraint (3.b.6.a) we get :

$$s_x(x,y)[u^t(x,y) + u_{CAM}(x,y)] + s_y(x,y)[v^t(x,y) + v_{CAM}(x,y)] + s_t(x,y) = 0\,. \qquad (4.c.16)$$

Then:

$$\iint_{\mathcal{B}} \Big[s_x(x,y)[u^t(x,y)+u_{CAM}(x,y)] + s_y(x,y)[v^t(x,y)+v_{CAM}(x,y)] + s_t(x,y) \Big]\, dx\, dy = 0\,, \qquad (4.c.17)$$

with (u^t, v^t) as above. From this we can obtain $\frac{T_3}{Z_W}$ as an affine function of ω_x, ω_y of the form:

$$\frac{T_3}{Z_W} = M_1\omega_x + M_2\omega_y + M_3, \qquad (4.c.18)$$

where

$$M_1 = -\frac{\iint_{\mathcal{B}} \Big[s_x xy + s_y(y^2+1) \Big]\, dx\, dy}{\iint_{\mathcal{B}} \Big[s_x(\frac{T_1}{T_3} - x) + s_y(\frac{T_2}{T_3} - y) \Big]\, dx\, dy}\,,$$

$$M_2 = \frac{\iint_{\mathcal{B}} \Big[s_x(x^2+1) + s_y xy \Big]\, dx\, dy}{\iint_{\mathcal{B}} \Big[s_x(\frac{T_1}{T_3} - x) + s_y(\frac{T_2}{T_3} - y) \Big]\, dx\, dy}\,,$$

$$M_3 = \frac{\iint_{\mathcal{B}} \Big[s_x\lambda(x,y) + s_y\mu(x,y) - s_t \Big]\, dx\, dy}{\iint_{\mathcal{B}} \Big[s_x(\frac{T_1}{T_3} - x) + s_y(\frac{T_2}{T_3} - y) \Big]\, dx\, dy}\,,$$

with

$$\lambda(x,y) = \omega_1 xy - \omega_2(x^2+1) + \omega_3 y\ ,$$

$$\mu(x,y) = \omega_1(y^2+1) - \omega_2 xy - \omega_3 x\ .$$

We can integrate by parts the above expressions and avoid the computation of the spatial derivatives of the image intensity function s(x,y). Using (4.c.18) in (4.c.14) and (4.c.15), we can express these estimates of the optical flow as functions only of the camera rotation parameters ω_x and ω_y. Then, if we substitute the expressions for $u^t, v^t, u_{CAM}, v_{CAM}$ in (4.c.6), the only remaining unknowns will be ω_x and ω_y and we can easily bring $J(\omega_x,\omega_y)$ in the form

$$J(\omega_x,\omega_y) = \alpha\omega_x^2 + \beta\omega_y^2 + \gamma\omega_x + \delta\omega_y + \epsilon\omega_x\omega_y + \zeta\ , \qquad (4.c.19)$$

where

$$\alpha = \iint_{\mathcal{B}} \left[((\frac{T_1}{T_3} - x)M_1 + xy)^2 + ((\frac{T_2}{T_3} - y)M_1 + (y^2+1))^2 \right] dx\,dy\ ,$$

$$\beta = \iint_{\mathcal{B}} \left[((\frac{T_1}{T_3} - x)M_2 - (x^2+1))^2 + ((\frac{T_2}{T_3} - y)M_2 - xy)^2 \right] dx\,dy\ ,$$

$$\gamma = 2\iint_{\mathcal{B}} \Big[((\frac{T_1}{T_3} - x)M_1 + xy)((\frac{T_1}{T_3} - x)M_3 - \lambda(x,y)) + ((\frac{T_2}{T_3} - y)M_1 + (y^2+1))((\frac{T_2}{T_3} - y)M_3 - \mu(x,y)) \Big] dx\,dy\ ,$$

$$\delta = 2\iint_{\mathcal{B}} \Big[((\frac{T_1}{T_3} - x)M_2 - (x^2+1))((\frac{T_1}{T_3} - x)M_3 - \lambda(x,y)) + ((\frac{T_2}{T_3} - y)M_2 - xy)((\frac{T_2}{T_3} - y)M_3 - \mu(x,y)) \Big] dx\,dy\ ,$$

$$\epsilon = 2\iint_{\mathcal{B}} \Big[((\frac{T_1}{T_3} - x)M_1 + xy)((\frac{T_1}{T_3} - x)M_2 - (x^2+1)) + ((\frac{T_2}{T_3} - y)M_1 + (y^2+1))((\frac{T_2}{T_3} - y)M_2 - xy) \Big] dx\,dy\ ,$$

$$\zeta = \iint_{\mathcal{B}} \left[((\frac{T_1}{T_3} - x)M_1 - \lambda(x,y))^2 + ((\frac{T_2}{T_3} - y)M_3 - \mu(x,y))^2 \right] dx\,dy\ .$$

Then, from Lemma 4.1 we can compute the optimal ω_x^* and ω_y^*, which will be:

$$\omega_x^* = \frac{\delta\epsilon - 2\beta\gamma}{4\alpha\beta - \epsilon^2} \qquad \text{and} \qquad \omega_y^* = \frac{\gamma\epsilon - 2\alpha\delta}{4\alpha\beta - \epsilon^2}\ .$$

4.d) Tracking System Simulation

The simulation of the tracking system was written in C and used the graphics environment of a Silicon Graphics Inc. IRIS workstation. The simulation is divided in the following parts:

The Image Formation Process

Our target is a plane in R^3 with surface normal vector $(-p, -q, 1)$. We consider the target covered by a grid and for each of its points we specify the respective projection on the image plane from the perspective projection function, the target kinematics and its position of the point in the target coordinate frame. The image on the image plane is then a textured shape like the one shown in fig. 4.1. ¿From the displacement of the target, we can specify the new array of intensities and thus simulate the image formation process. The kinematic models of Ch. 2.b are used for this purpose.

The I/O Process

Contains the graphics routines that represent the image-plane view of the moving target. Input is the array of intensities s(x,y,t) generated at the Image Formation process at time t. Output on the display of the IRIS is the image frame at time t with the image of the moving target, the actual and estimated target motion parameters and a plot of the distance (measured in pixels) of the point P_W, the projection on the image plane of the point W of the target (which is marked with a blue cross) from the origin of the image plane coordinate system (marked with a red cross on the image) at each time instant. On this diagram yellow segments represent the target motion, while red segments represent the tracking action of the camera (fig. 4.1).

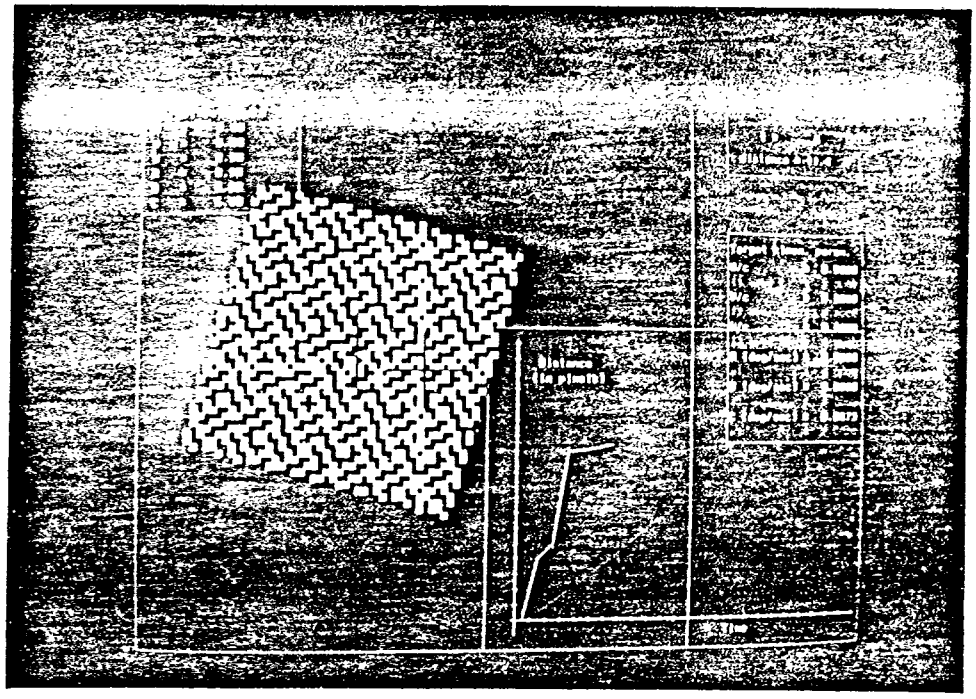


Fig. 4.1 : Display of simulation results on the IRIS

The Target Tracker

The tracking schemes of Ch. 4.c were implemented. They assume that a feature extraction process has found the feature P_W on the image-plane during the acquisition phase, but, due to noise and computational errors, its position may be specified only up to a neighborhood $\mathcal{B}_W$ and they attempt to stabilize $\mathcal{B}_W$ in the center of the image plane.

1) Tracking Schema 4.1:

The exact formulation of this Tracking Schema for the planar case used in the simulations was given as an example in Ch. 4.c. Very good results were obtained for this tracking scheme fed with "perfect" estimates of the target motion (accurate direction of translation and rotation) and for very general motions. Results degenerate "gracefully" as the quality of estimates decreases due to very fast motion or noisy images. The random number generator of C was used in order to generate a sequence of "noisy" target motion estimates in order to test the performance of the algorithm under inaccurate estimates. The algorithm performs well with 30% to 60% noise in target motion estimation and even in the case when some of the parameters are estimated with 90% deviation from real values, it performs satisfactorily enough. In the experimental results below, the target translational velocities are given in (length units) per (time unit) and the angular velocities in degrees per (time unit).

a . Only translational motion of the target ($T_1 = -20, T_2 = 30, T_3 = 5$). 0% noise in motion estimation. The tracking algorithm keeps the target foveated (fig. 4.2).

b . Only rotation of the target ($\omega_1 = -2, \omega_2 = 3, \omega_3 = 4$). 0% noise in motion estimation. The tracking algorithm keeps the target foveated (fig. 4.3).

c . General motion of target ($T_1 = 20, T_2 = 30, T_3 = 5, \omega_1 = 2, \omega_2 = 3, \omega_3 = 5$). 0% noise in motion estimation. The tracking algorithm keeps the target foveated (fig. 4.4).

d . General motion of target ($T_1 = 20, T_2 = 30, T_3 = 5, \omega_1 = 2, \omega_2 = 3, \omega_3 = 5$). 30% noise in motion estimation. Very small deviation of the target (2-3 pixels) from the center of the image plane (fig. 4.5).

e . General motion of target ($T_1 = 20, T_2 = 30, T_3 = 5, \omega_1 = 2, \omega_2 = 3, \omega_3 = 5$). 60% noise in motion estimation. Small deviation of the target (6-7 pixels) from the center of the image plane (fig. 4.6).

f . General motion of target ($T_1 = 20, T_2 = 30, T_3 = 5, \omega_1 = 2, \omega_2 = 3, \omega_3 = 5$). 90% noise in motion estimation. Deviation of the target about 8 pixels from the center of the image plane (fig. 4.7). Small enough compared to the resolution of the image we consider (91 x 91) pixels.

g . General motion of target along the line of sight ($T_1 = 20, T_2 = 0, T_3 = 5, \omega_1 = 0, \omega_2 = -2, \omega_3 = 10$). 60% noise in motion estimation. The tracking algorithm keeps the target foveated (fig. 4.8).

2) Tracking schema 4.2:

Poor results compared to the previous scheme. When the denominator of the M_i 's

$$\iint_{\mathcal{B}} \left[s_x(\frac{T_1}{T_3} - x) + s_y(\frac{T_2}{T_3} - y) \right] dx\, dy$$

becomes very small, the whole scheme becomes unstable and since $\mathcal{B}$ is a neighborhood of the origin of the image plane, if the spatial variation of the target intensity is not large enough, this expression will be very small (fig. 4.9) .

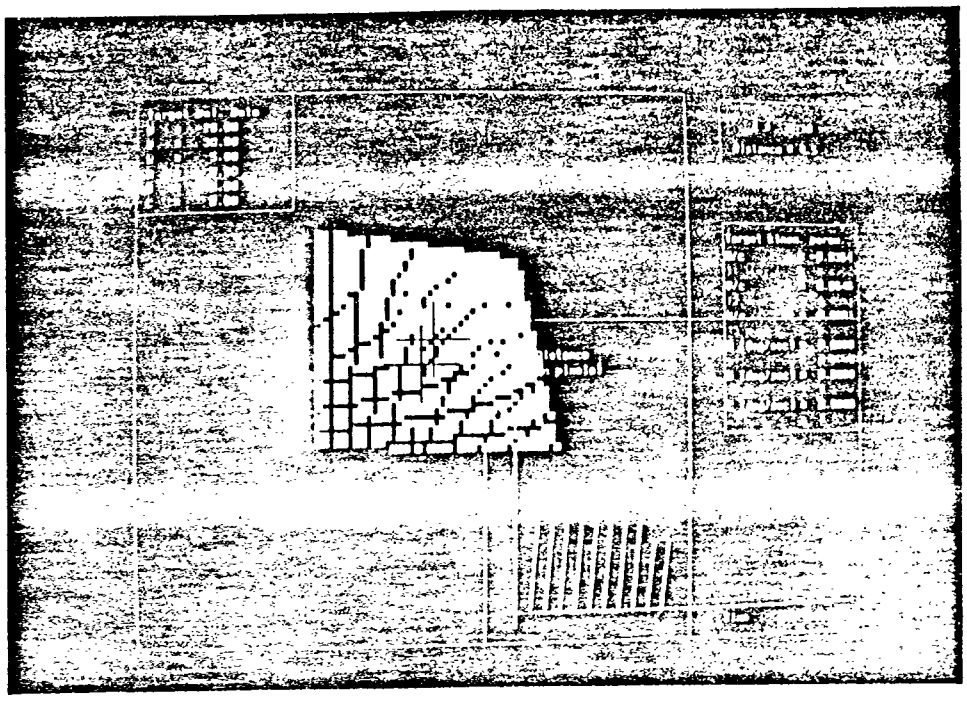


Fig. 4.2 : Target tracker, target translation, 0% noise.

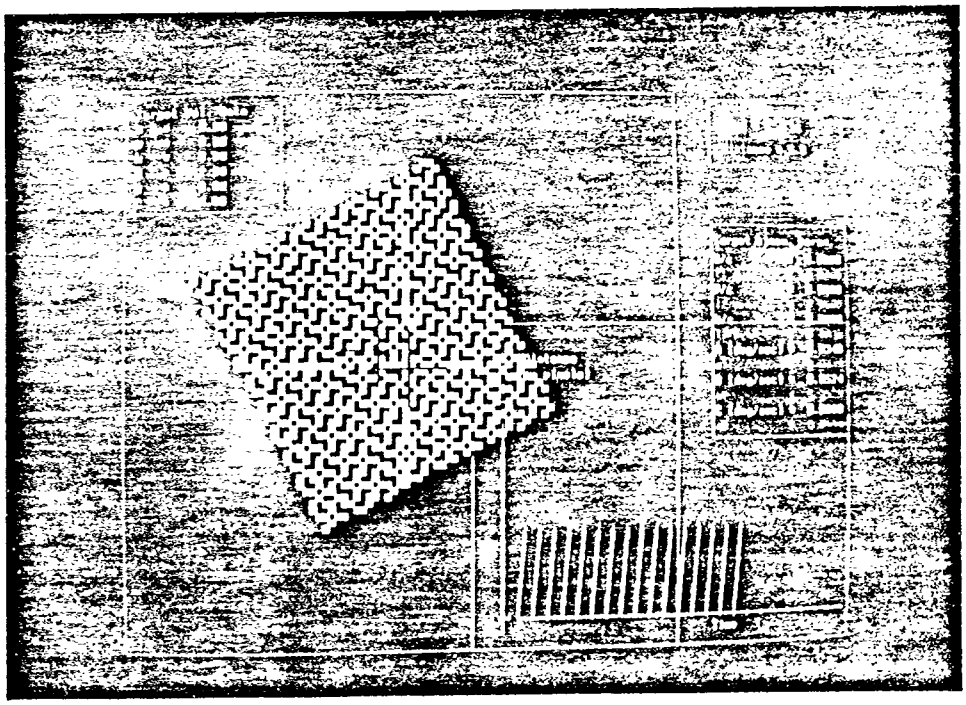


Fig. 4.3 : Target tracker, target rotation, 0% noise.

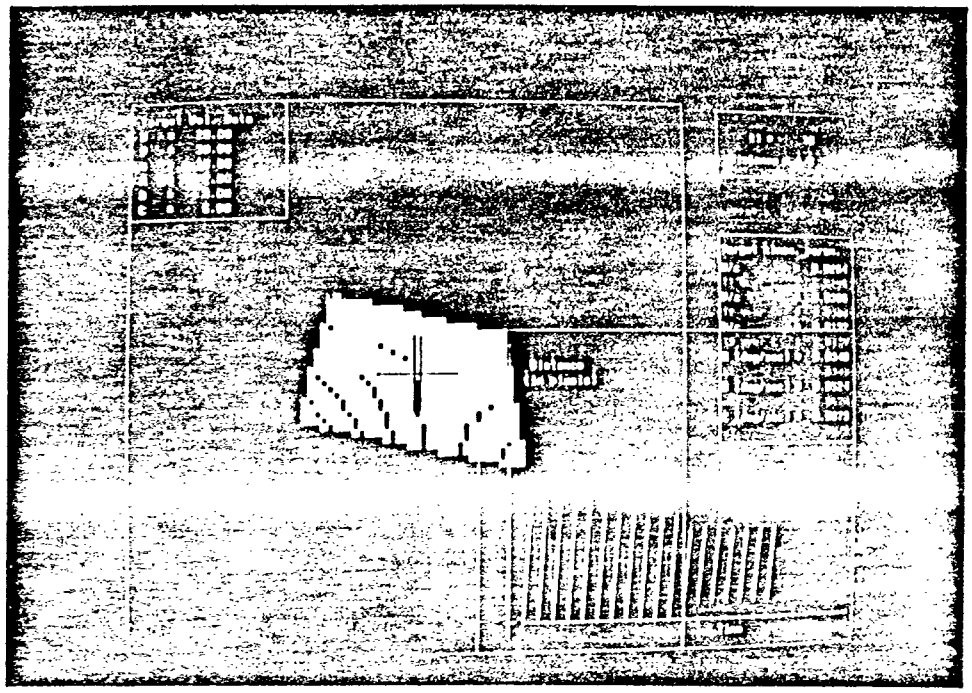


Fig. 4.4 : Target tracker, general motion of target, 0% noise.

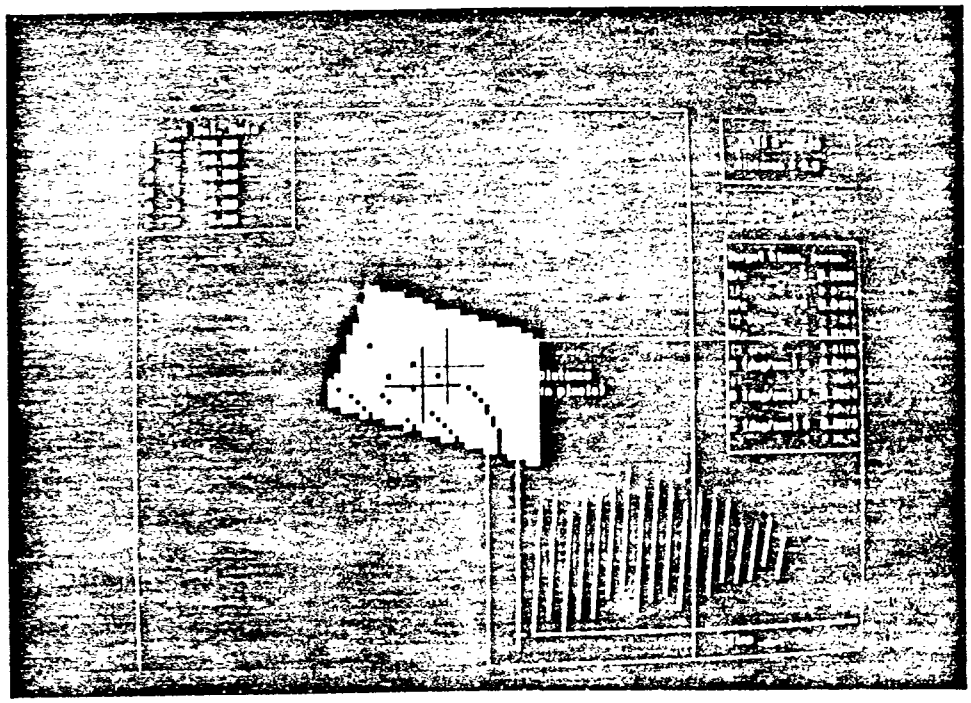


Fig. 4.5 : Target tracker, general motion of target, 30% noise.

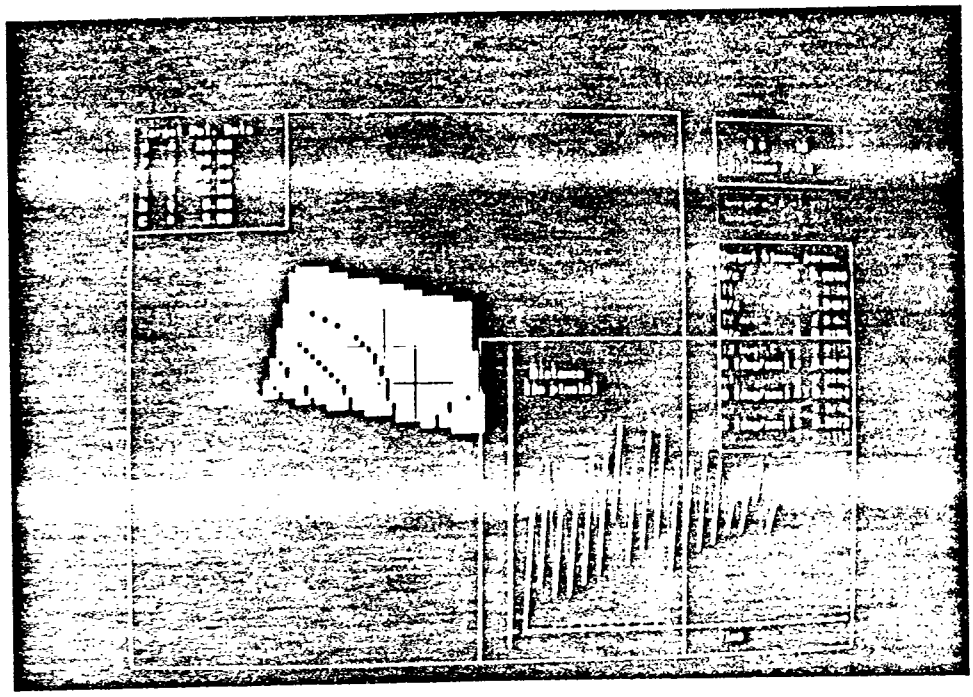


Fig. 4.6 : Target tracker, general motion of target, 60% noise.

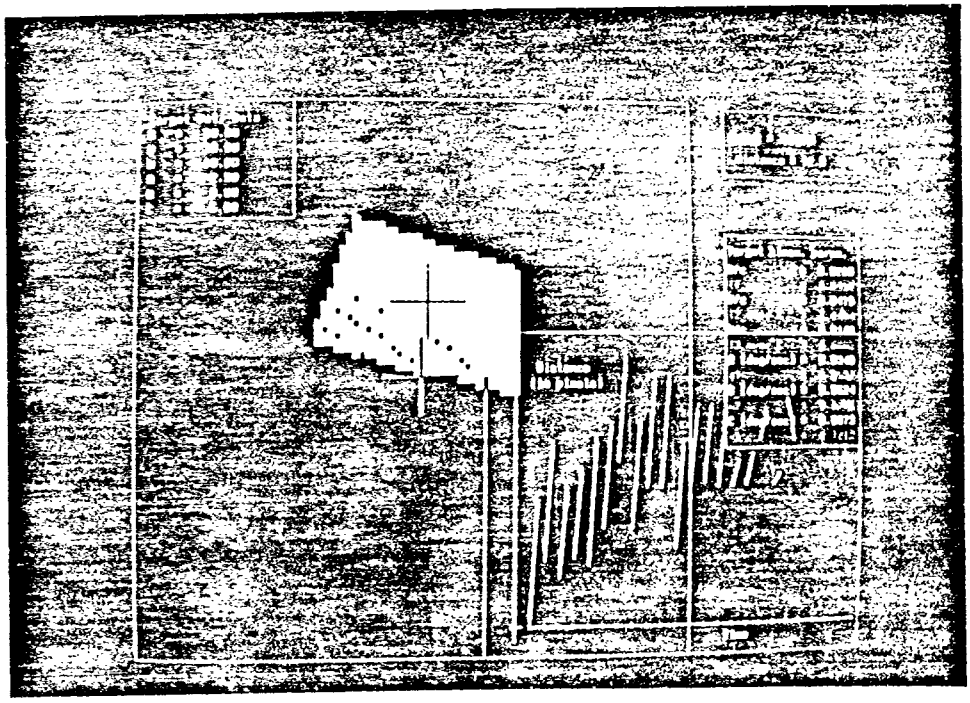


Fig. 4.7 : Target tracker, general motion of target, 90% noise.

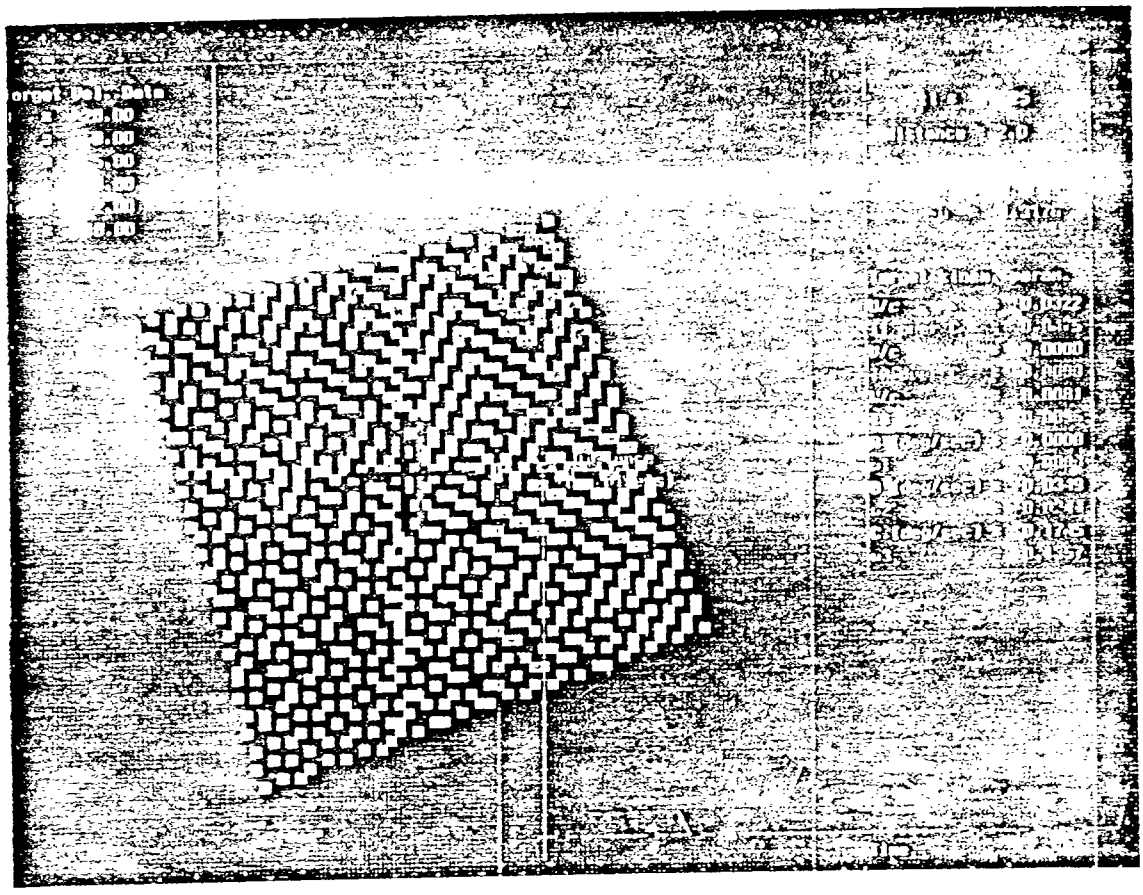


Fig. 4.8 : Target tracker, target motion along the line of sight, 60% noise.

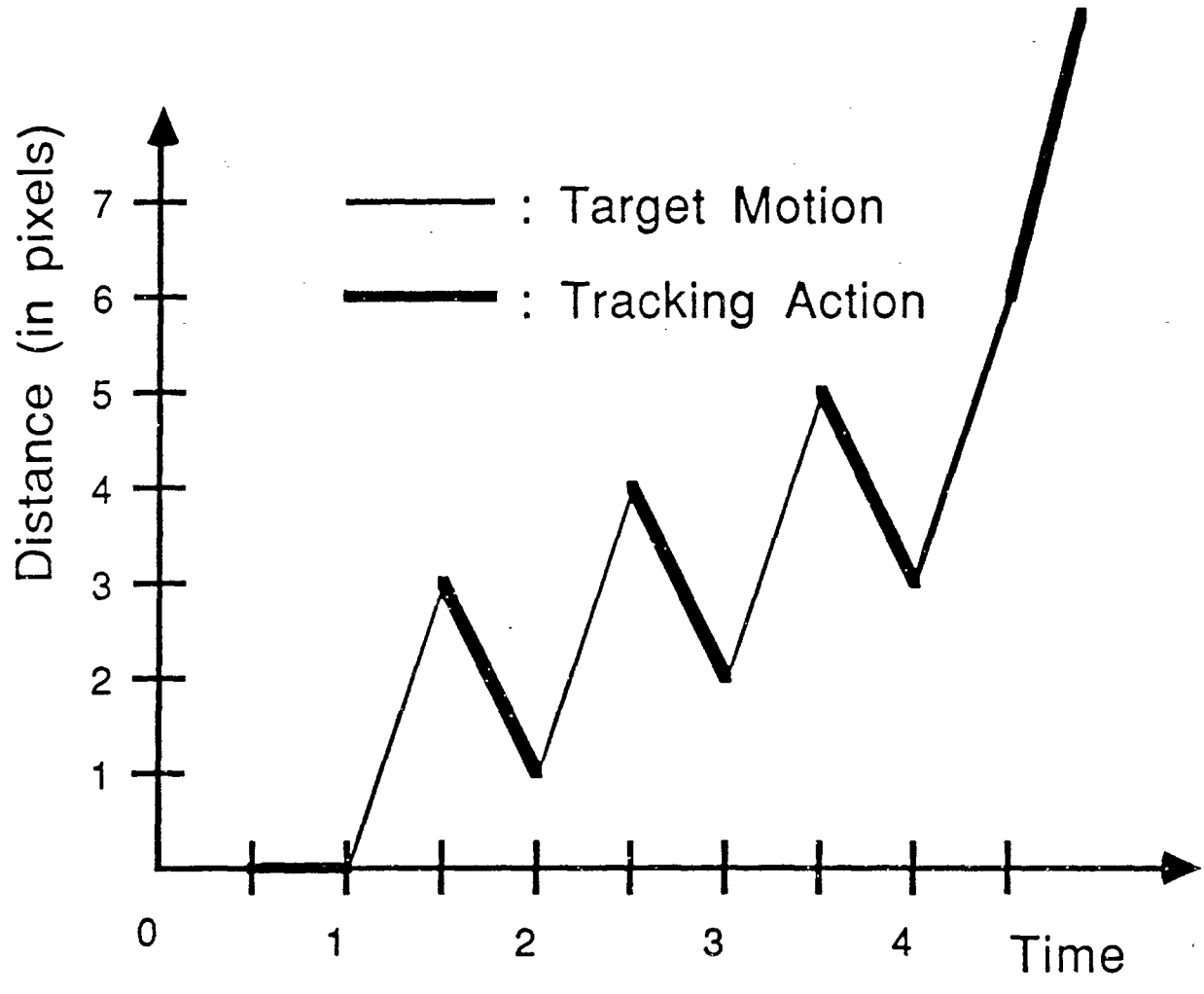


Fig. 4.9 : Target tracker, tracking schema 4.2.

The Target Motion Parameter Estimator

The motion estimation involves the formulation and solution of a linear system of equations of the form $\mathbf{H}m = \dot{f}$, where m is the target motion parameter vector, $\dot{f}$ is derived from image information and H is derived from image and target shape information. The formulation of this system for the planar case considered in the simulations was given in the example of Ch. 4.b. The system is solved using Greville's algorithm for partitioned Generalized Inverses, an iterative scheme, which determines the Moore–Penrose pseudoinverse $\mathbf{H}^{\dagger}$ of the general $n \times m$ matrix H in m steps. This algorithm is particularly well suited for our case, where $m = 6$ and the computation of $\mathbf{H}^{\dagger}$ is completed in 6 steps no matter what the length n of the vector of linear features is. This flexibility is not allowed if we attempt to estimate $\mathbf{H}^{\dagger}$ from the formula $(\mathbf{H}^{\top}\mathbf{H})^{-1}\mathbf{H}^{\top}$, where in addition a rank test is required.

Several sets of linear features were considered before establishing experimentally that the best results are obtained with moments of the image, i.e. with linear features of the form

$$\iint_{\mathcal{B}} x^i y^j s(x,y,t)dx\,dy \ , \ 0 \le i+j \le k.$$

The quality of the motion parameter estimation depends on the order k of the moments considered. Moments up to order k=10 were considered and in certain cases significant improvement in performance was obtained from experiments with moments of order up to k=4 or less.

The results that follow show the target motion parameter estimator combined with the tracking schema 4.1 for an image with resolution 21 x 21.

a . Only translational motion of the target ($T_1 = 30$). Moments up to order k=10 are used (fig. 4.10) .

b . Only translational motion of the target ($T_1 = 30$). Moments up to order k=2 are used (fig. 4.11) .

c . Only rotation of the target ($\omega_2 = 5$). Moments up to order k=10 are used (fig. 4.12) .

d . Only rotation of the target ($\omega_2 = 5$). Moments up to order k=2 are used (fig. 4.13) .

e . General motion of target ($T_1 = 40, \omega_2 = -2, \omega_3 = 5$). Moments up to order k=5 are used (fig. 4.14) .

f . Motion of target along the line of sight ($T_1 = 20, T_2 = 0, T_3 = 5, \omega_1 = 0, \omega_2 = -2, \omega_3 = 10$). Moments up to order k=2 are used (fig. 4.15) .

Implementation Issues

Errors are introduced in this scheme from the following sources:

1) Discretization effects, which will affect mainly the Optical Flow Constraint equation used in the derivation of H and the discrete temporal derivatives of the linear features.

2) Numerical instabilities in the solution of the linear system

$$\mathrm{H}m = \dot{f}.$$

3)The coarse resolution of the image used in simulations. Due to this a translation may have the same visual effect as a specific rotation and vice versa. This may affect the motion estimation algorithm, but ultimately it will not affect tracking. Using the fine resolution of a real system, this effect is expected to disappear.

The use of moments as linear features has the advantage that they are easily implementable in hardware (Cagney and Mallon [22]), therefore the above scheme can be used for real–time applications. Moreover, multiprocessor architectures such as the fine–grain Connection Machine (Hillis [37]) are an interesting alternative for the real–time computation of the matrix H and the vector of linear features, which are the computational bottlenecks of the above algorithm.

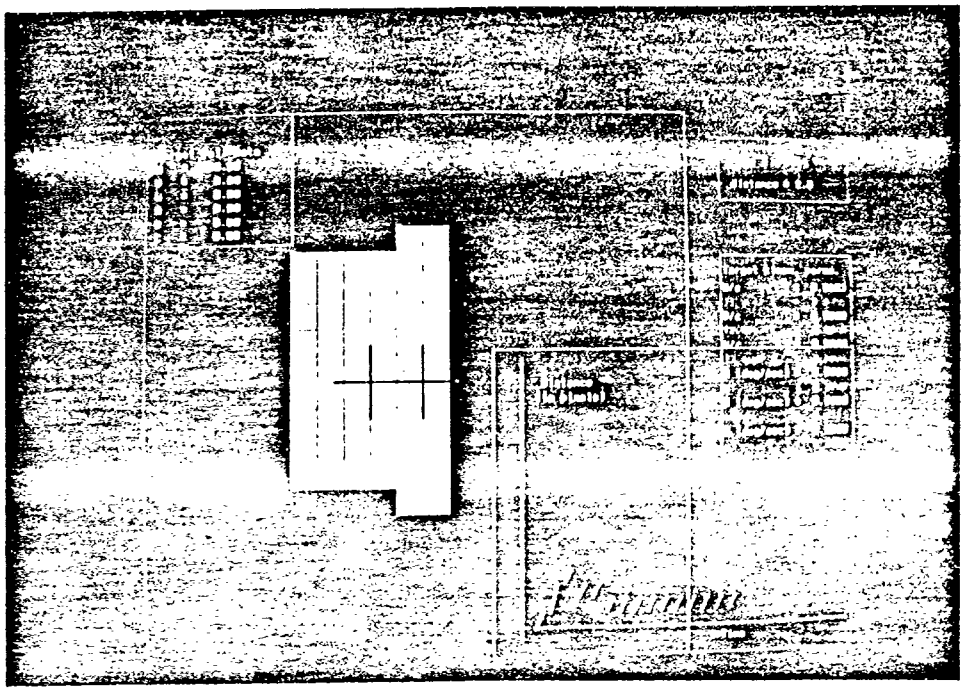


Fig. 4.10 : Motion parameter estimator and target tracker, target translation, k=10

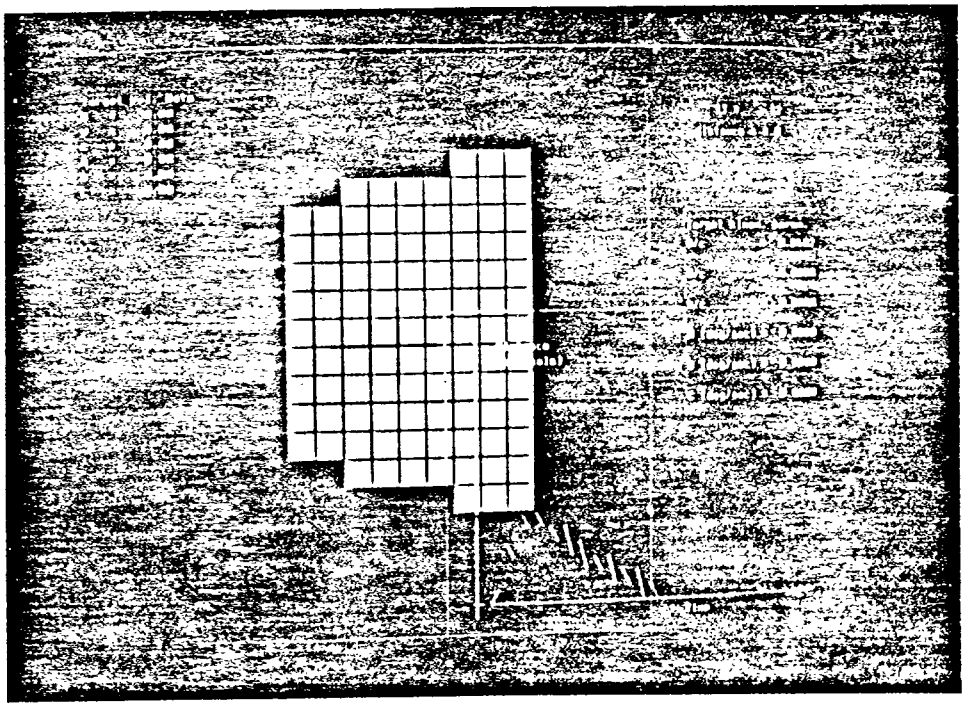


Fig. 4.11 : Motion parameter estimator and target tracker, target translation, k=2

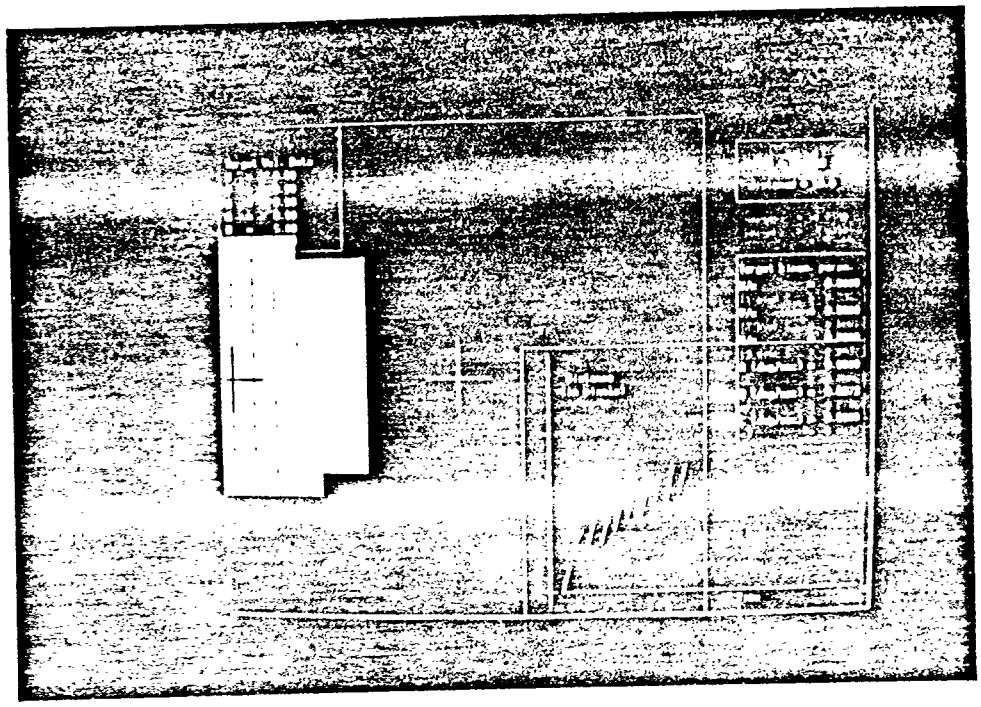


Fig. 4.12 : Motion parameter estimator and target tracker, target rotation, k=10

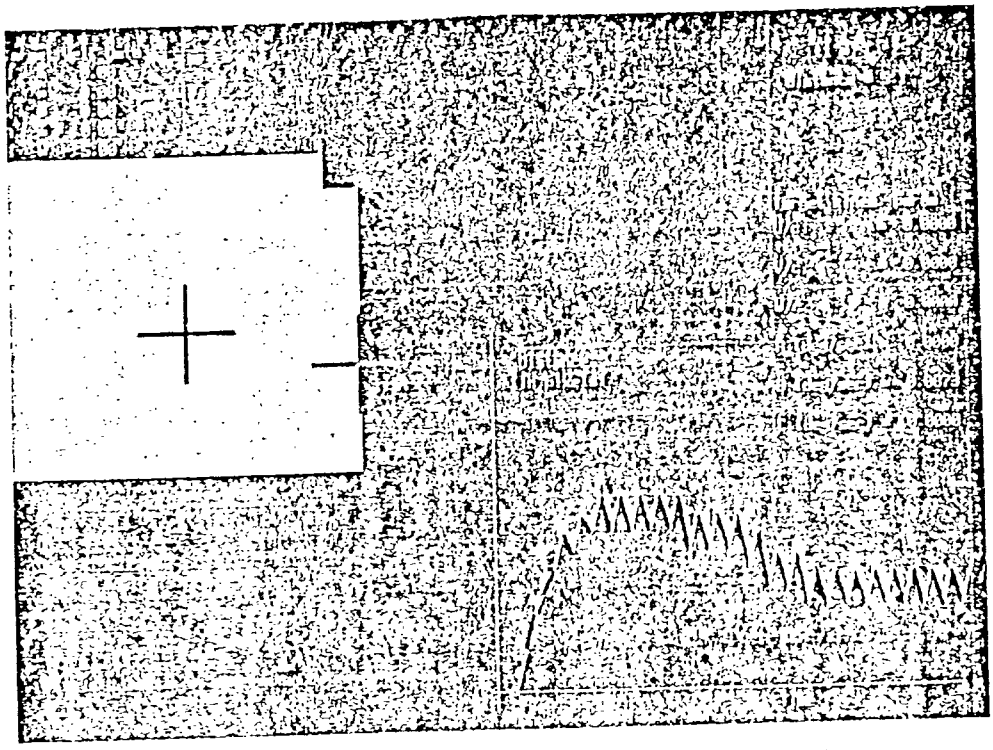


Fig. 4.13 : Motion parameter estimator and target tracker, target rotation, k=2

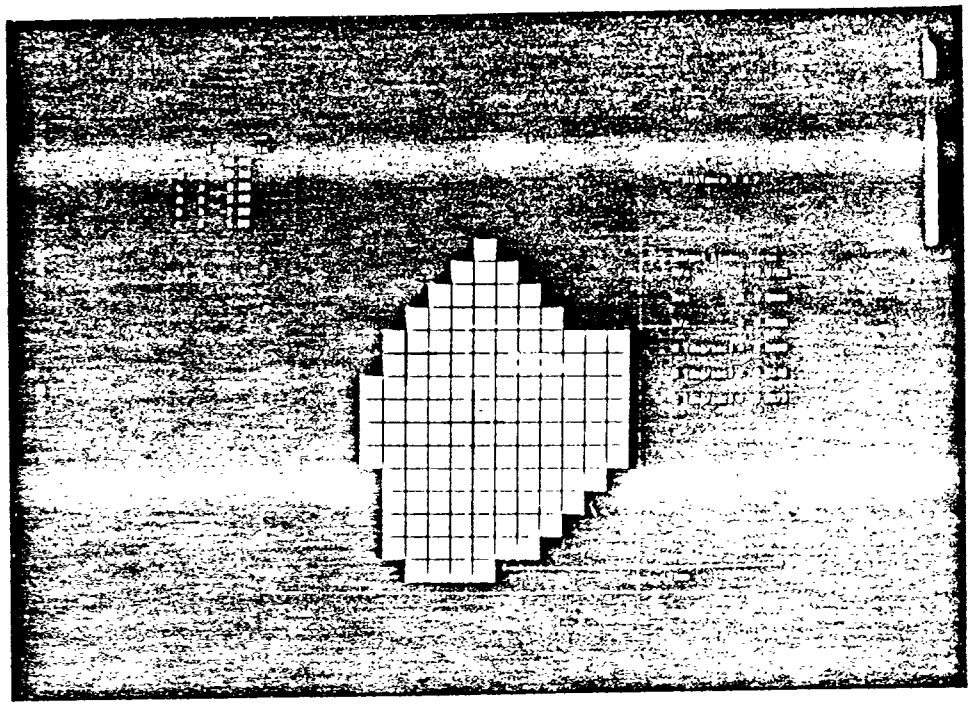


Fig. 4.14 : Motion parameter estimator and target tracker,
general motion of the target, k=5

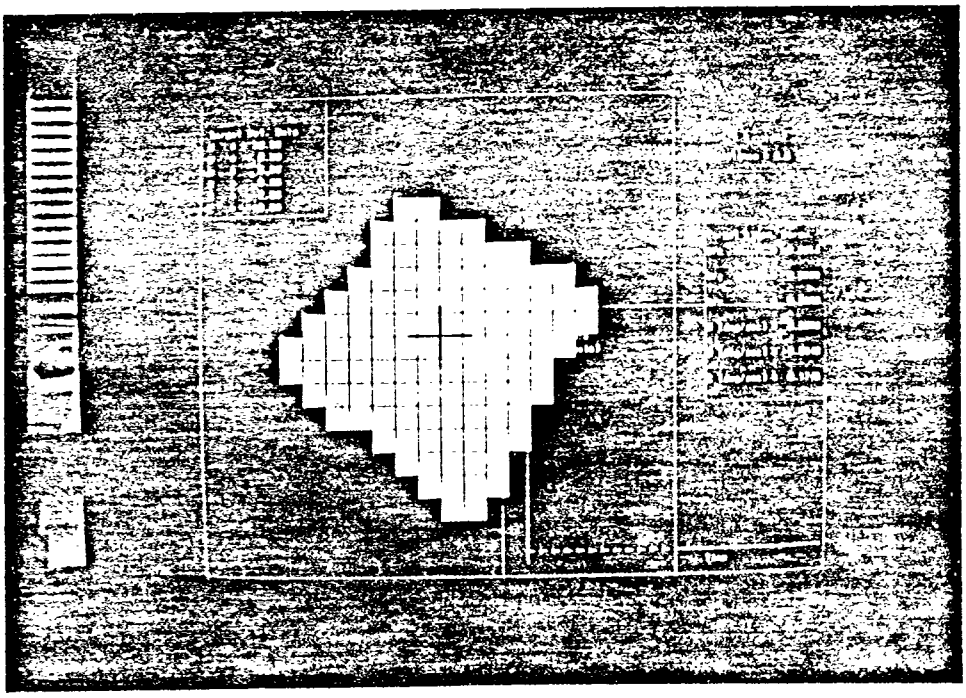


Fig. 4.15 : Motion parameter estimator and target tracker,
motion of the target along the line of sight, k=2

CHAPTER

FIVE

CONCLUSIONS AND FUTURE RESEARCH

The problem examined in this thesis was the tracking phase of the visual tracking of 3–dimensional targets moving in 3–dimensions, which is the most general tracking situation in practice. We devised a general class of algorithms based on the continuous motion and optical flow formalism, which, contrary to previous approaches, use dynamic segmentation of the image sequence and are correspondence–free. Therefore, two major shortcomings of most previous tracking systems, namely segmentation and feature matching, are bypassed and a novel formulation of the visual tracking problem as an optimization problem is presented, which leads to the *Tracking Constraint*. This expresses the camera angular velocity that will achieve tracking as the minimizer of an appropriate cost functional, which requires that for the tracking to be successful, the optical flow in a neighborhood of the origin of the image plane has to be minimized.

The Tracking Constraint, together with the Optical Flow Constraint are used to create two classes of algorithmic schemes for tracking, one where we assume knowledge of the optical flow field on a neighborhood of the origin of the image plane and one where we assume knowledge of the shape of the target. In the second case, the optical flow can be expressed as a linear combination of the motion parameters (direction of translation and rotation), which can be estimated using linear features of the image, which in turn provide us with global, correspondence–free information about the temporal variation of the image. The linear features formulation allows us not to compute numerically the

spatial derivatives of the image intensity function, which is an ill-posed problem, but use instead the spatial derivatives of the corresponding measuring functions, which can be computed analytically.

Simulation results demonstrate the success of this class of algorithms in tracking 3-D targets, even under very noisy information like very coarse image resolution and erroneous motion parameter estimation.

Hardware implementation and testing of our algorithms in a real visual environment is one of our primary future research goals. The moment invariants which we used as linear features have the advantage of being easily implementable on special hardware or computable on multiprocessor computer systems such as the Connection Machine. Therefore the computational bottleneck, which in our simulations was the computation of the lineal features vector and the matrix H (see Ch. 4.b), can be eliminated and real-time application of the algorithms becomes possible. These algorithms will generate a desired trajectory for the camera actuators to follow and appropriate modeling and control schemes are necessary, in order to implement this desired motion in real-time.

In Aloimonos, Bandopadhay and Weiss [2] a 3-D shape estimation scheme was presented, which assumes knowledge of the 3-D motion and estimates shape, following a very similar approach to the one used in our 3-D motion estimation algorithm. A cooperative scheme based on these two algorithms may be able to achieve tracking in the case when both approximate shape and approximate motion information are available *a priori*. Interaction between these two algorithms, together with additional visual information, may be able to improve the shape and motion estimates as time evolves, until some kind of equilibrium is reached.

Provided we have an estimate of the target 3-D motion from the algorithm presented in Ch. 4.b, this information can be used not just for tracking, but as part of an augmented system able to acquire a target (track and approach it) and/or perform obstacle avoidance. Apart from obvious obstacle avoidance applications in robot navigation, this may have other interesting applications such as grabbing a rotating satellite based on visual information or combining visual and tactile information in an intelligent control scheme (Fu [27], Saridis [63]) that will allow a robot to approach, grasp and manipulate a moving object in a flexible manufacturing environment.

Even though linear features have been proven a very important tool both in classical Pattern Recognition and in novel correspondence-free motion and shape estimation al-

gorithms, like the ones presented here, no theoretical investigation has been undertaken on the relative merits of specific measuring functions μ_i and the optimal length k of the linear features vector. The experimental attempt has been to make this vector as rich in image-related information as possible, but the exact meaning of "richness" has never been stated formally. A formal measure of the richness of information depending on the selection of μ_i and k will possibly enable the use of our powerful optimization tools in order to derive optimality results applicable to each specific visual situation.

Finally, the whole previous approach was related to rigid targets. Humans are perfectly able to track the motion of non-rigid ones, such as clouds, flags, sea waves and other humans, even though their shape or motion parameters may not be exactly known. This is related to the understanding of non-rigid motion from visual information and several researchers are already studying this problem (see e.g. Shulman and Aloimonos [65]). An answer to the problem may be provided through learning and neural nets, but several aspects of such an approach are still not very well understood and require further investigation.

APPENDIX A

Proof of Theorem 2.1 :

i) See Arnold [6] and Goldstein [33].

ii) Consider two time instants t_0 and t with $t_0 > t$.

$$q(t) = r(t) + B(t)Q(t), \tag{A.1}$$

$$q(t_0) = r(t_0) + B(t_0)Q(t_0), \tag{A.2}$$

with $B(t), B(t_0) \in SO(3)$.

The matrix B is a function of time t and we can expand it to Taylor series around t_0 (Bottema and Roth [16]). Defining $\Delta t = t - t_0$, we get:

$$B(t) = B(t_0) + \frac{dB(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2B(t_0)}{dt^2}\Delta t^2 + \dots ,$$

Since $B(.) \in I\!R \times SO(3)$, we have:

$$B^{-1}(t) = B^{\mathsf{T}}(t) = B^{\mathsf{T}}(t_0) + \frac{dB(t_0)}{dt}^{\mathsf{T}}\Delta t + \frac{1}{2!}\frac{d^2B(t_0)}{dt^{2\,\mathsf{T}}}\Delta t^2 + \dots ,$$

and from $B(t)B^{\mathsf{T}}(t) = I$, we get:

$$B(t_0)\frac{dB(t_0)}{dt}^{\mathsf{T}} + \frac{dB(t_0)}{dt}B^{\mathsf{T}}(t_0) = 0$$

Therefore $\frac{dB(t_0)}{dt}B^{\mathsf{T}}(t_0)$ is a skew-symmetric matrix. The corresponding vector ω is the spatial angular velocity of the point (i.e. its angular velocity with respect to the inertial frame, not the moving one) and then:

$$\hat{\omega} = \frac{dB(t_0)}{dt}B^{\mathsf{T}}(t_0) = \frac{dB(t_0)}{dt}B^{-1}(t_0). \tag{A.3}$$

Similarly we can expand r(t), Q(t) and q(t):

$$r(t) = r(t_0) + \frac{dr(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2r(t_0)}{dt^2}\Delta t^2 + \dots , \tag{A.4}$$

$$Q(t) = Q(t_0) + \frac{dQ(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2Q(t_0)}{dt^2}\Delta t^2 + \ldots, \tag{A.5}$$

$$q(t) = q(t_0) + \frac{dq(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2q(t_0)}{dt^2}\Delta t^2 + \ldots, \tag{A.6}$$

From (A.1), (A.4), (A.5) and (A.6), we get:

$$\begin{aligned}
q(t) &= r(t) + B(t)Q(t) \\
&= [r(t_0) + \frac{dr(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2r(t_0)}{dt^2}\Delta t^2 + \ldots] \\
&\quad + [B(t_0) + \frac{dB(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2B(t_0)}{dt^2}\Delta t^2 + \ldots] \\
&\quad \times [Q(t_0) + \frac{dQ(t_0)}{dt}\Delta t + \frac{1}{2!}\frac{d^2Q(t_0)}{dt^2}\Delta t^2 + \ldots] \qquad (A.7) \\
&= [r(t_0) + B(t_0)Q(t)] + [\frac{dr(t_0)}{dt} + \frac{dB(t_0)}{dt}Q(t) + B(t_0)\frac{dQ(t_0)}{dt}]\Delta t \\
&\quad + [\frac{1}{2!}\frac{d^2r(t_0)}{dt^2} + \frac{1}{2!}\frac{d^2B(t_0)}{dt^2}Q(t) \\
&\quad + \frac{1}{2!}B(t_0)\frac{d^2Q(t_0)}{dt^2} + \frac{dB(t_0)}{dt}\frac{dQ(t_0)}{dt}]\Delta t^2 + \ldots
\end{aligned}$$

In the Continuous approach to the 3-D Motion problem in Vision, images are taken in very small time intervals, so that the motion of the target between any two frames can be considered infinitessimal. The infinitessimal velocity distribution can be obtained as an approximation of the general expression (A.7) for $q(t)$, if we neglect terms of order higher than one in Δt and consider $\Delta t \longrightarrow dt$.

Then the velocity distribution is described by the first derivative of q as follows:

$$\begin{aligned}
\frac{dq(t_0)}{dt} &= \frac{dr(t_0)}{dt} + \frac{dB(t_0)}{dt}Q(t) + B(t_0)\frac{dQ(t_0)}{dt} \\
&= \frac{dr(t_0)}{dt} + \frac{dB(t_0)}{dt}B^{-1}(t_0)[X(t_0) - r(t_0)] + B(t_0)\frac{dQ(t_0)}{dt} \\
&\overset{(A.3)}{=} \frac{dr(t_0)}{dt} + \hat{\omega}[X(t_0) - r(t_0)] + B(t_0)\frac{dQ(t_0)}{dt} \qquad (A.8) \\
&= [\frac{dr(t_0)}{dt} - \hat{\omega}r(t_0)] + \hat{\omega}X(t_0) + B(t_0)\frac{dQ(t_0)}{dt} \\
&= T + \hat{\omega}X(t_0) + B(t_0)\frac{dQ(t_0)}{dt}\,.
\end{aligned}$$

Then:

$$q(t) = Tdt + (I + \hat{\omega}dt)q(t_0) + B(t_0)\dot{Q}(t_0)dt\,. \qquad (A.9)$$

Therefore, supposing that we know the position of the target at time t_0 (i.e. $r(t_0)$ and $B(t_0)$) and its velocity distribution T and ω, we can specify its position at time t. From $T = \frac{dr(t_0)}{dt} - \hat{\omega}r(t_0)$ we have:

$$r(t) = r(t_0) + \frac{dr(t_0)}{dt}dt = Tdt + (I + \hat{\omega}dt)r(t_0)$$

¿From $\hat{\omega} = \frac{dB(t_0)}{dt}B^{\top}(t_0)$ we have :

$$B(t) = B(t_0) + \frac{dB(t_0)}{dt}dt = (I + \hat{\omega}dt)B(t_0)$$

Proof of Theorem 2.2 :

i) From Corollaries 2.3 and 2.1 :

$$X_0(t) = d(t) + \mathcal{A}(t)X(t) = d(t) + \mathcal{A}(t)r(t) + \mathcal{A}R(t)X'\,.$$

Defining $r_0(t) = d(t) + \mathcal{A}(t)r(t)$ and $R_0(t) = \mathcal{A}(t)R(t)$ we get (2.b.7).

ii) From Corollaries 2.1 and 2.3 :

$$\begin{aligned} R_0(t) = \mathcal{A}(t)R(t) &= (I + \hat{\omega}_c dt)\mathcal{A}(t_0)(I + \hat{\omega}dt)R(t_0) \\ &= \mathcal{A}(t_0)R(t_0) + \mathcal{A}(t_0)\hat{\omega}R(t_0)dt + \hat{\omega}_c\mathcal{A}(t_0)R(t_0)dt + \hat{\omega}_c\mathcal{A}(t_0)\hat{\omega}R(t_0)dt^2 \end{aligned}$$

Ignoring the terms of order higher than one in dt and since from (i) : $R_0(t_0) = \mathcal{A}(t_0)R(t_0)$, we get :

$$\begin{aligned} R_0(t) &= R_0(t_0) + \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^{\top}(t_0)R_0(t_0)dt + \hat{\omega}_c R_0(t_0)dt \\ &= \left[I + (\hat{\omega}_c + \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^{\top}(t_0)dt\right]R_0(t_0) \end{aligned}$$

Therefore, $\hat{\omega}_0 = \hat{\omega}_c + \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^{\top}(t_0)$.

From Corollaries 2.3 and 2.1 :

$$\begin{aligned} \frac{dX_0(t_0)}{dt} &= T_c + \hat{\omega}_c X_0(t_0) + \mathcal{A}(t_0)\dot{X}(t_0) \\ &= T_c + \hat{\omega}_c X_0(t_0) + \mathcal{A}(t_0)[T + \hat{\omega}X(t_0)] \\ &= T_c + \hat{\omega}_c X_0(t_0) + \mathcal{A}(t_0)T + \mathcal{A}(t_0)\hat{\omega}X(t_0) \\ &= T_c + \mathcal{A}(t_0)T - \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^{\top}(t_0)d(t_0) \\ &\qquad + [\hat{\omega}_c + \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^{\top}(t_0)][d(t_0) + \mathcal{A}(t_0)X(t_0)]\,. \end{aligned}$$

Defining $T_0 = T_c + \mathcal{A}(t_0)\hat{\omega} - \mathcal{A}(t_0)\hat{\omega}\mathcal{A}^T(t_0)d(t_0)$ and since from Corollary 2.3 we have : $X_0(t_0) = d(t_0) + \mathcal{A}(t_0)X(t_0)$, we take (2.b.8).

Proof of Theorem 2.3 :

Assume perspective projection and consider a point A of the target with coordinates $\vec{X} = (X, Y, Z)$ with respect to the camera frame, which is projected on the image plane at a point with coordinates (x,y). Differentiating the projection equations (2.a.3), we get:

$$u \triangleq \dot{x} = f\frac{\dot{X}}{Z} - f\frac{X}{Z^2}\dot{Z}$$

and

$$v \triangleq \dot{y} = f\frac{\dot{Y}}{Z} - f\frac{Y}{Z^2}\dot{Z}$$

¿From Corollary 2.1 :

$$\dot{\vec{X}} = T + \hat{\omega}\vec{X} \quad \Rightarrow \quad \begin{cases} \dot{X} = T_1 + (\omega_2 Z - \omega_3 Y) \\ \dot{Y} = T_2 + (\omega_3 X - \omega_1 Z) \\ \dot{Z} = T_3 + (\omega_1 Y - \omega_2 X) \end{cases}$$

Therefore:

$$u = \frac{fT_1 - xT_3}{Z} - \omega_1\frac{xy}{f} + \omega_2\frac{(x^2 + f^2)}{f} - \omega_3 y$$

$$v = \frac{fT_2 - yT_3}{Z} - \omega_1\frac{(y^2 + f^2)}{f} + \omega_2\frac{xy}{f} + \omega_3 x \ .$$

APPENDIX B :

Proof of Theorem 3.1 :

Consider the Lagrangian of the problem:

$$\begin{aligned} F(\omega_x,\omega_y,\lambda) &= f(\omega_x,\omega_y) + \lambda g(\omega_x,\omega_y) \\ &= a_1\omega_x^2 + b_1\omega_y^2 + c_1\omega_x + d_1\omega_y + e_1\omega_x\omega_y + f_1 \\ &\quad + \lambda(a_2\omega_x^2 + b_2\omega_y^2 + c_2\omega_x + d_2\omega_y + e_2\omega_x\omega_y + f_2) \end{aligned} \quad , \qquad (B.1)$$

where $a_1,\ldots,f_2$ are defined in (3.b.10).

i) *First-Order Necessary Conditions*

Supposing $\omega = (\omega_x,\omega_y)$ is a local minimum for the above problem and that $\frac{\partial g(\omega_x,\omega_y)}{\partial \omega}$ is full row rank, there exists $\lambda \in I\!R$ such that:

$$\nabla F(\omega_x,\omega_y,\lambda) = 0 \Rightarrow \begin{pmatrix} \frac{\partial F}{\partial \omega_x}(\omega_x,\omega_y,\lambda) \\ \frac{\partial F}{\partial \omega_y}(\omega_x,\omega_y,\lambda) \\ \frac{\partial F}{\partial \lambda}(\omega_x,\omega_y,\lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow$$

$$2a_1\omega_x + c_1 + e_1\omega_y + \lambda(2a_2\omega_x + c_2 + e_2\omega_y) = 0 \Rightarrow \lambda = -\frac{2a_2\omega_x + c_2 + e_2\omega_y}{2a_1\omega_x + c_1 + e_1\omega_y} \Rightarrow$$

$$2b_1\omega_y + d_1 + e_1\omega_x - \frac{2a_2\omega_x + c_2 + e_2\omega_y}{2a_1\omega_x + c_1 + e_1\omega_y}(2b_2\omega_y + d_2 + e_2\omega_x) = 0$$

$$\Rightarrow \quad A\omega_x^2 + B(\omega_y)\omega_x + \Gamma(\omega_y) = 0 \qquad (B.2)$$

where $A = 2(a_1e_2 - e_1a_2)$, $B(\omega_y) = B_1\omega_y + B_2 = 4(a_1b_2 - b_1a_2)\omega_y + 2(a_1d_2 - d_1a_2) + (c_1e_2 - e_1c_2)$ and $\Gamma(\omega_y) = \Gamma_1\omega_y^2 + \Gamma_2\omega_y + \Gamma_3 = 2(e_1b_2 - b_1e_2)\omega_y^2 + (2(c_1b_2 - b_1c_2) + (e_1d_2 - d_1e_2))\omega_y + (c_1d_2 - d_1c_2)$. ¿From $g(\omega_x,\omega_y) = 0$, we get:

$$a_2\omega_x^2 + b_2\omega_y^2 + c_2\omega_x + d_2\omega_y + e_2\omega_x\omega_y + f_2 = 0\,, \qquad (B.3)$$

Solving (B.2) for ω_x and replacing its expression to (B.3), we can get the following fourth-order equation, from whose real solutions we can specify ω_y :

$$\alpha\omega_y^4 + \beta\omega_y^3 + \gamma\omega_y^2 + \delta\omega_y + \epsilon = 0\,, \qquad (B.4)$$

where

$$\alpha = 4\Big[a_2A^2\Gamma_1^2 + b_2^2A^4 + a_2b_2A^2B_1^2 + (e_2^2 - 2a_2b_2)A^3\Gamma_1 - a_2e_2A^2B_1\Gamma_1 - b_2e_2A^3B_1\Big],$$

$$\begin{aligned}\beta = 4\Big[&2a_2^2A^2\Gamma_1\Gamma_2 + 2a_2b_2A^2B_1B_2 + a_2d_2A^2B_1^2 + (e_2^2 - 2a_2b_2)A^3\Gamma_2 - a_2c_2A^2B_1\Gamma_1\\ &+2(c_2e_2 - a_2d_2)A^3\Gamma_1 - a_2e_2A^2(B_1\Gamma_2 + B_2\Gamma_1)\\ &-(b_2c_2 + d_2\epsilon_2)A^3B_1 + 2b_2d_2A^4 - b_2e_2A^3B_2\Big],\end{aligned}$$

$$\begin{aligned}\gamma = 4\Big[&a_2^2A^2(\Gamma_2^2 + 2\Gamma_1\Gamma_3) + d_2^2A^4 + a_2b_2A^2B_2^2 + 2a_2d_2A^2B_1B_2 + a_2f_2A^2B_1^2\\ &+(e_2^2 - 2a_2b_2)A^3\Gamma_3 - a_2c_2A^2(B_1\Gamma_2 + B_2\Gamma_1) + 2(c_2e_2 - a_2d_2)A^3\Gamma_2\\ &-a_2e_2A^2(B_1\Gamma_3 + B_2\Gamma_2) + (c_2^2 - 2a_2f_2)A^3\Gamma_1 - (b_2c_2 + d_2e_2)A^3B_2\\ &-(c_2d_2 - e_2f_2)A^3B_1 + 2b_2f_2A^4\Big],\end{aligned}$$

$$\begin{aligned}\delta = 4\Big[&2a_2^2A^2\Gamma_2\Gamma_3 + a_2d_2A^2B_2^2 + 2a_2f_2A^2B_1B_2 - a_2c_2A^2(B_1\Gamma_3 + B_2\Gamma_2)\\ &+2(c_2e_2 - a_2d_2)A^3\Gamma_3 - a_2e_2A^2B_2\Gamma_3 + (c_2^2 - 2a_2f_2)A^3\Gamma_2\\ &-(c_2d_2 + e_2f_2)A^3B_2 - c_2f_2A^3B_1 + 2d_2f_2A^4\Big],\end{aligned}$$

$$\epsilon = 4\Big[a_2^2A^2\Gamma_3^2 + f_2^2A^4 + a_2f_2A^2B_2^2 - a_2c_2A^2B_2\Gamma_3 + (c_2^2 - 2a_2f_2)A^3\Gamma_3 - c_2f_2A^3B_2\Big].$$

By substituting $y = \omega_y + \frac{\beta}{4\alpha}$ and dividing by α, we bring (B.4) in the following form:

$$y^4 + py^2 + qy + r = 0\,, \qquad (B.5)$$

where $p = \frac{\gamma}{\alpha} - \frac{3\beta^2}{8\alpha^2}$, $q = \frac{\delta}{\alpha} + \frac{\beta^3}{8\alpha^3} - \frac{\beta\gamma}{2\alpha^2}$, and $r = \frac{\epsilon}{\alpha} - \frac{\beta\delta}{4\alpha^2} - \frac{3\beta^4}{256\alpha^4} + \frac{\beta^2\gamma}{16\alpha^3}$. Consider now the *cubic resolvent* of (B.5), which is the following cubic equation in normal form :

$$z^3 + 2pz^2 + (p^2 - 4r)z - q^2 = 0\,. \qquad (B.6)$$

Substitute $w = z + \frac{2p}{3}$ and get its reduced form:

$$w^3 + sw + t = 0\,, \qquad (B.7)$$

with $s = -\frac{1}{3}(p^2 + 12r)$ and $t = -(\frac{2}{27}p^3 - \frac{8}{3}pr + q^2)$.

Consider the discriminant of (B.7): $D_1 = \frac{s}{3}^3 + \frac{t}{2}^2$. ¿From Cardano's formula (Bronshtein and Semendyayev [20]), the solutions of (B.7) are : $w_1 = u + v, w_2 = -\frac{1}{2}(u+v) + i\sqrt{\frac{3}{2}}(u-v), w_3 = -\frac{1}{2}(u+v) - i\sqrt{\frac{3}{2}}(u-v)$, with $u = \sqrt[3]{-\frac{t}{2} + \sqrt{D_1}}$ and $v = \sqrt[3]{-\frac{t}{2} - \sqrt{D_1}}$.

Then, the solutions of the cubic resolvent (B.6) are: $z_i = w_i - \frac{2p}{3}, i = 1,2,3$ and the solutions of the fourth-order equation (B.4) are:

$$\begin{aligned}
\omega_y^{(1)} &= \frac{1}{2}(\sqrt{z_1} + \sqrt{z_2} + \sqrt{z_3}) - \frac{\beta}{4\alpha}, \\
\omega_y^{(2)} &= \frac{1}{2}(\sqrt{z_1} - \sqrt{z_2} - \sqrt{z_3}) - \frac{\beta}{4\alpha}, \\
\omega_y^{(3)} &= \frac{1}{2}(-\sqrt{z_1} + \sqrt{z_2} - \sqrt{z_3}) - \frac{\beta}{4\alpha}, \\
\omega_y^{(4)} &= \frac{1}{2}(-\sqrt{z_1} - \sqrt{z_2} + \sqrt{z_3}) - \frac{\beta}{4\alpha}.
\end{aligned} \tag{B.8}$$

For $D_1 \leq 0$, the equation (B.7) has 3 real solutions w_i, otherwise it has a real one and a pair of complex conjugate ones. ¿From the theorem of Vieta: $z_1 z_2 z_3 = q^2 \geq 0$, therefore, either all 3 solutions of (B.6) will be non-negative, or one will be non-negative and two negative. Then, for $D_1 \leq 0$ and $z_i \geq 0, i = 1,2,3$, the fourth-order equation (B.4) has at most four distinct real solutions $\omega_y^{(i)}, i = 1,2,3,4$. For $D_1 > 0$ the equation has at most two distinct real solutions $\omega_y^{(i)}, i = 1,2$. In all other cases it does not have real solutions.

If we get real solutions $\omega_y^{(i)}$ for (B.4), we should compute the corresponding ω_x from (B.2). For each $\omega_y^{(i)}$ we get a different quadratic equation in ω_x of the form:

$$A\,\omega_x^{(i)^2} + B(\omega_y^{(i)})\omega_x^{(i)} + \Gamma(\omega_y^{(i)}) = 0. \tag{B.9}$$

Consider its discriminant $D_2^{(i)} = B(\omega_y^{(i)})^2 - 4A\Gamma(\omega_y^{(i)})$. We are interested only in real solutions of (B.9), i.e. in the case $D_2^{(i)} \geq 0$. Then we get : $\omega_{x_1}^{(i)} = -\frac{1}{2A}(B + \sqrt{D_2(i)})$ and $\omega_{x_2}^{(i)} = -\frac{1}{2A}(B - \sqrt{D_2(i)})$.

Thus, from the necessary conditions for the minimization problem (P), we get the local extrema $(\omega_{x_j}^{(i)}, \omega_y^{(i)})$, for i=1,..,m and j=1,..,n, with $1 \leq m \leq 4$ and $1 \leq n \leq 2$. To

every such extremum corresponds a Lagrange multiplier of the form:

$$\lambda^{(i)}(\omega_{x_j}^{(i)}, \omega_y^{(i)}) = -\frac{2a_1\omega_{x_j}^{(i)} + c_1 + e_1\omega_y^{(i)}}{2a_2\omega_{x_j}^{(i)} + c_2 + e_2\omega_y^{(i)}} \,. \qquad (B.10)$$

ii) *Second-Order Sufficiency Conditions*

We must check the second-order conditions for those local extrema, in order to specify the strict local minima. For $\omega \in I\!R^2$, consider the set $T(\omega) = \{h \in I\!R^2 | \frac{\partial g(\omega)}{\partial \omega} h = 0\}$. It is easy to see that for our problem $T(\omega) = \{(h_1, h_2) \in I\!R^2 | h_1 = K(\omega)h_2\}$, where $K(\omega) = -\frac{2b_2\omega_y + d_2 + e_2\omega_x}{2a_1\omega_x + c_2 + e_2\omega_y}$, which represents a line through the origin in $I\!R^2$.

Now, suppose that $\omega \in I\!R^2$ is sush that $g(\omega) = 0$ and there exists $\lambda \in I\!R$ such that $\nabla F(\omega, \lambda) = 0$. If $\forall h \in T(\omega), h \neq 0$ we have $< h, \frac{\partial^2 F(\omega,\lambda)}{\partial \omega^2} h > > 0$, then ω is a *strict local minimizer* of the above problem.

In our case for $h = (h_1, h_2) \in T(\omega)$ with $h \neq 0$, i.e. $h_1 = K(\omega)h_2, h_2 \neq 0$, we have:

$$[(a_1 + \lambda(\omega)a_2)K^2(\omega) + (e_1 + \lambda(\omega)e_2)K(\omega) + (b_1 + \lambda(\omega)b_2)]h_2^2 > 0$$

Obviously this is true for every ω sush that:

$$D_3(\omega) \triangleq (a_1 + \lambda(\omega)a_2)K^2(\omega) + (e_1 + \lambda(\omega)e_2)K(\omega) + (b_1 + \lambda(\omega)b_2) > 0 \,. \qquad (B.11)$$

Every pair $\omega^{ij} = (\omega_{x_j}^{(i)}, \omega_y^{(i)})$ that satisfies $D_3^{ij}(\omega^{ij}) > 0$ is a strict local minimizer of the problem.

iii) *Global minima*

Supposing that there exist ℓ such pairs ω^i, with $i = 1, \ldots, \ell$ and $0 \leq \ell \leq 8$, we can compute $f(\omega^i)$ and specify the **global minima** ω^* of the problem.

Proof of Theorem 3.2 :

For simplicity, consider the case when S is a 4×4 array, where $\mathcal{B}$ is as shown in fig. 3.1. Then $k = 4$ and :

$$
\begin{aligned}
J_\lambda^D &= \sum_{i=1}^{16}(s_{x_i}u_i + s_{y_i}v_i + s_{t_i})^2 + \lambda \sum_{i=6,7,10,11}(u_i^2 + v_i^2) \\
&= s_{x_1}^2u_1^2 + s_{y_1}^2v_1^2 + 2s_{x_1}s_{y_1}u_1v_1 + 2s_{x_1}s_{t_1}u_1 + 2s_{y_1}s_{t_1}v_1 + s_{t_1}^2 \\
&\quad + s_{x_2}^2u_2^2 + s_{y_2}^2v_2^2 + 2s_{x_2}s_{y_2}u_2v_2 + 2s_{x_2}s_{t_2}u_2 + 2s_{y_2}s_{t_2}v_2 + s_{t_2}^2 \\
&\quad + \ldots + \\
&\quad + s_{x_{16}}^2u_{16}^2 + s_{y_{16}}^2v_{16}^2 + 2s_{x_{16}}s_{y_{16}}u_{16}v_{16} + 2s_{x_{16}}s_{t_{16}}u_{16} + 2s_{y_{16}}s_{t_{16}}v_{16} + s_{t_{16}}^2 \\
&\quad + \lambda[u_6^2 + v_6^2 + u_7^2 + v_7^2 + u_{10}^2 + v_{10}^2 + u_{11}^2 + v_{11}^2] \\
&= \theta^\top\tilde{\Sigma}\theta - 2C^\top\theta + d + \lambda\theta^\top\Sigma\theta\,,
\end{aligned}
\tag{B.12}
$$

where:

$$
\theta = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ u_{16} \\ v_{16} \end{pmatrix}\,,\quad \tilde{\Sigma} = \begin{pmatrix} \tilde{\Sigma}_1 & 0 & \ldots & 0 \\ 0 & \tilde{\Sigma}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \tilde{\Sigma}_{16} \end{pmatrix}\,,
$$

with

$$
\tilde{\Sigma}_i = \begin{pmatrix} s_{x_i}^2 & s_{x_i}s_{y_i} \\ s_{x_i}s_{y_i} & s_{y_i}^2 \end{pmatrix}\,,
$$

$$
C = \begin{pmatrix} -s_{x_1}s_{t_1} \\ -s_{y_1}s_{t_1} \\ -s_{x_2}s_{t_2} \\ \vdots \\ -s_{x_{16}}s_{t_{16}} \\ -s_{y_{16}}s_{t_{16}} \end{pmatrix}\,,\ d = s_{t_1}^2 + s_{t_2}^2 + \ldots + s_{t_{16}}^2\,,
$$

and

$$
\Sigma = diag\{\underset{\substack{\uparrow \\ 1}}{0},\ \ldots,\ 0,\ \underset{\substack{\uparrow \\ 11}}{1},\ 1,\ 1,\ 1,\ 0,\ \ldots,\ 0,\ \underset{\substack{\uparrow \\ 19}}{1},\ 1,\ 1,\ 1,\ 0,\ \ldots,\ \underset{\substack{\uparrow \\ 32}}{0}\ \}
$$

We must now transform (B.12) in the form (B.15). Define :

$$y \triangleq k \begin{pmatrix} -s_{t_1} \\ -s_{t_2} \\ \vdots \\ -s_{t_{16}} \end{pmatrix} , X \triangleq k \begin{pmatrix} s_{x_1} & s_{y_1} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & s_{x_2} & s_{y_2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & s_{x_{16}} & s_{y_{16}} \end{pmatrix} .$$

It is easy to see that :

$$\tilde{\Sigma} = \frac{1}{k^2} X^\top X \, , \, C^\top = \frac{1}{k^2} y^\top X \, , \, d = \frac{1}{k^2} y^\top y \, .$$

Then :

$$\begin{aligned} \theta^\top \tilde{\Sigma} \theta - 2C^\top \theta + d &= \frac{1}{k^2}(\theta^\top X^\top X \theta - 2y^\top X \theta + y^\top y) \\ &= \frac{1}{k^2}(< y, y > - < y, X\theta > - < X\theta, y > + < X\theta, X\theta >) \\ &= \frac{1}{k^2} < y - X\theta, y - X\theta > \\ &= \frac{1}{k^2} ||y - X\theta||^2 \end{aligned}$$

Therefore :

$$J_\lambda^D = \frac{1}{k^2} ||y - X\theta||^2 + \lambda \theta^\top \Sigma \theta \, ,$$

where Σ is symmetric and positive definite.

APPENDIX C

Moore – Penrose Inverse :

The *Moore-Penrose Inverse* of an $n \times m$ real matrix A is defined as the unique matrix $A^\dagger \in I\!R^{m \times n}$ for which:

$$\begin{aligned} AA^\dagger A &= A\,, \\ A^\dagger AA^\dagger &= A^\dagger\,, \\ (AA^\dagger)^\top &= AA^\dagger\,, \\ (A^\dagger A)^\top &= A^\dagger A\,. \end{aligned} \tag{C.1}$$

Lemma C.1 : Let $A \in I\!R^{n \times m}$, $b \in I\!R^n$. Then, among the least–squares solutions of the equation $Ax = b$, $x_0 = A^\dagger b$ is the one of minimum norm, i.e. for any other least–squares solution x_1 of $Ax = b$, we have $||x_0|| < ||x_1||$. Conversely, if $X \in I\!R^{m \times n}$ is such that for all b, Xb is the minimum–norm least–squares solution of $Ax = b$, then $X = A^\dagger$.

Proof : See Corollary 3.2.3 of Ben–Israel and Greville [12].

Then for x_0 we have:

$$||Ax_0 - b|| \leq ||Ax - b||\,, \quad \forall x$$

and

$$||x_0|| \leq ||x_1||\,, \quad \forall x_1 \text{ such that } ||Ax_0 - b|| = ||Ax_1 - b||.$$

Therefore, the general least–squares solution of $Ax = b$ will be:

$$x_1 = x_0 + \tilde{x} = A^\dagger b + \tilde{x}\,,$$

where $\tilde{x} \in \mathcal{N}(A)$, the null space of A.

Lemma C.2 : Let $A \in I\!R^{n \times m}$ and $rank(A) = m$, then:

$$A^\dagger = (A^\top A)^{-1} A^\top$$

Proof : It is easy to prove for any matrix A that : $A^{\dagger} = (A^{\top}A)^{\dagger}A^{\top}$ by checking the properties (C.1). For A such that $rank(A) = m$, we have $rank(A^{\top}A) = m$ and since $A^{\top}A \in I\!R^{m\times m}$, then $A^{\top}A$ is nonsingular, in which case $(A^{\top}A)^{\dagger} = (A^{\top}A)^{-1}$ and the result is immediate.

Lemma C.3 : (Greville' Algorithm)
Consider the notation below:

$$\mathbf{H_k} = [\mathbf{H_{k-1}}|\mathbf{h_k}]\,,\ k = 2,\ldots,m,$$

with $\mathbf{h_k}$ the k–th column of $\mathbf{H_k}$. Also :

$$d_k = \mathbf{H}_{\mathbf{k-1}}^{\dagger}\mathbf{h_k}$$

$$c_k = \mathbf{h_k} - \mathbf{H_{k-1}}d_k$$

Then, for $k = 2,\ldots,m$:

$$\mathbf{H}_{\mathbf{k}}^{\dagger} = [\mathbf{H_{k-1}}|\mathbf{h_k}]^{\dagger} = \begin{pmatrix} \mathbf{H}_{\mathbf{k-1}}^{\dagger} - d_k b_k^{\top} \\ b_k^{\top} \end{pmatrix},$$

where

$$b_k^{\top} = \begin{cases} c_k^{\dagger}, & \text{if } c_k \neq 0; \\ (1 + d_k^{\top}d_k)^{-1}d_k^{\top}\mathbf{H}_{\mathbf{k-1}}^{\dagger}, & \text{if } c_k = 0; \end{cases}$$

Proof : See Theorem 5.5.7 in Ben-Israel and Greville [12].

A more efficient procedure was devised also from Greville for the calculation of $\mathbf{H}^{\dagger}\vec{f}$, without the need for computing $\mathbf{H}^{\dagger}$ first. Consider the augmented matrix $\tilde{\mathbf{H}} = [\mathbf{H}|\vec{f}]$. Then $\mathbf{H}^{\dagger}\tilde{\mathbf{H}} = [\mathbf{H}^{\dagger}\mathbf{H}|\mathbf{H}^{\dagger}\vec{f}]$ and if we can get $\mathbf{H}^{\dagger}\tilde{\mathbf{H}}$ from an iterative procedure, which in the kth–step computes $\mathbf{H}_{\mathbf{k}}^{\dagger}\tilde{\mathbf{H}}$, then the last column of $\mathbf{H}^{\dagger}\tilde{\mathbf{H}}$ is the solution $\vec{m} = \mathbf{H}^{\dagger}\vec{f}$ of our system.

From the previous lemma, for $k = 2,\ldots,m$

$$\mathbf{H}_{\mathbf{k}}^{\dagger}\tilde{\mathbf{H}} = [\mathbf{H_{k-1}}|\mathbf{h_k}]^{\dagger}\tilde{\mathbf{H}} = \begin{pmatrix} \mathbf{H}_{\mathbf{k-1}}^{\dagger}\tilde{\mathbf{H}} - d_k b_k^{\top}\tilde{\mathbf{H}} \\ b_k^{\top}\tilde{\mathbf{H}} \end{pmatrix},$$

and d_k can be seen as the k–th column of $\mathbf{H}_{\mathbf{k-1}}^{\dagger}\tilde{\mathbf{H}}$. Moreover,

$$b_k^{\top}\tilde{\mathbf{H}} = \begin{cases} c_k^{\dagger}\tilde{\mathbf{H}}, & \text{if } c_k \neq 0; \\ (1 + d_k^{\top}d_k)^{-1}d_k^{\top}\mathbf{H}_{\mathbf{k-1}}^{\dagger}\tilde{\mathbf{H}}, & \text{if } c_k = 0\ ; \end{cases}$$

Proof of Lemma 4.1 :

Once J is twice continuously differentiable in its arguments, its Hessian P exists and is a symmetric 2×2 matrix:

$$P = \frac{\partial^2 J}{\partial x^2} = \begin{pmatrix} \frac{\partial^2 J}{\partial x^2} & \frac{\partial^2 J}{\partial x \partial y} \\ \frac{\partial^2 J}{\partial y \partial x} & \frac{\partial^2 J}{\partial y^2} \end{pmatrix} = \begin{pmatrix} 2\alpha & 0 \\ 0 & 2\beta \end{pmatrix}.$$

Consider a nonzero vector $\mathbf{x} = (x, y) \in R^2$.Then:

$$\mathbf{x}^\top P \mathbf{x} = 2(\alpha x^2 + \beta y^2).$$

Under the above hypothesis about α and β, the matrix P is positive definite, thus J is a strictly convex function (Luenberger [48]). Therefore, the minimizer of J is unique and global (Varaiya[72]). Let $\mathbf{x}^* = (x^*, y^*)$ be such a minimizer, i.e.

$$\nabla J(\mathbf{x}^*) = 0.$$

Then $\mathbf{x}^*$ is the unique global minimizer of J and can be computed from: $\nabla J(\mathbf{x}^*) = 0$.

$$\nabla J(x^*, y^*) = \begin{pmatrix} 2\alpha x^* + \epsilon y^* + \gamma \\ 2\beta y^* + \epsilon x^* + \delta \end{pmatrix} = 0 \Rightarrow$$

$$2\alpha x^* + \epsilon y^* = -\gamma, \qquad (C.2)$$

$$\epsilon x^* + 2\beta y^* = -\delta. \qquad (C.3)$$

If $4\alpha\beta - \epsilon^2 \neq 0$, which in general holds in our system, then (C.2) and (C.3) have the unique solution (4.c.2) and (4.c.3).

BIBLIOGRAPHY

[1] D.M.Allen, "The relationship between variable selection and data augmentation and a method for prediction", *Technometrics*, 13, pp. 469–475, 1987.

[2] J.Aloimonos, I.Weiss and A.Bandopadhay, "Active Vision", *Proc. of the 1st International Conference on Computer Vision*, London, 1987.

[3] J.Aloimonos and D.Shulman, "Discontinuous Regularization", *Workshop on Connectionist Models in Computational and Cognitive Science*, University of Maryland, May 1987.

[4] S.Amari, "Invariant Structures of Signal and Feature Spaces in pattern recognition problems", *RAAG Memoirs*, Vol. 4, 1968.

[5] S.Amari, "Feature Spaces wich admit and Detect Invariant Signal Transformations", *Proc. 4th Intl. Joint Conf. on Pattern Recognition*, pp. 452–456, Tokyo, 1978.

[6] V.I.Arnold, *Mathematical Methods of Classical Mechanics*, Springer Verlag, New York, 1978.

[7] R.Bajcsy and L.Lieberman, "Texture gradients as a depth cue", *CGIP*, 5, pp.52–67, 1976.

[8] A.T.Bahill and T.LaRitz, "Why can't Batters Keep their Eyes on the Ball?", *American Scientist*, 72, pp. 219–253, 1984.

[9] A.T.Bahill and J.D.McDonald, "Model Emulates Human Smooth Pursuit System Producing Zero-Latency Target Tracking", *Biol. Cybren.*, 48, pp. 213–222, 1983.

[10] A.Bandopadhay, "A Computational Study of Rigid Motion Perception", Ph. D. Thesis, Department of Computer Science, University of Rochester, 1986.

[11] D.M.Bates, M.J.Lindstrom, G.Wahba and B.S.Yandell, "GCVPACK– Routines for Generalized Cross Validation", *Commun. Statist.-Simula.*, 16(1), pp.263–297, 1987.

[12] A.Ben-Israel and T.N.E.Greville, *Generalized Inverses: Theory and Applications*, John Wiley and Sons, 1974.

[13] H.J.Bernstein, "Costraints in Real-Time Data Acquisition and Control", TR-195, Courant Institute of Mathematical Sciences, NYU, December 1985.

[14] P.J.Besl and R.C.Jain, "Invariant Surface Characteristics for 3-D Object Recognition in Range Images", *GCVIP*, 33, pp. 33-80, 1986.

[15] A.S.Blaivas, "Visual Analysis : Theory of Lie group representations", *Math. Biosciences*, 28, pp. 45-67, 1975.

[16] O.Bottema and B.Roth, *Theoretical Kinematics*, North-Holland Co., 1979.

[17] P.Bouthemy and A.Benveniste, "Modelling of Atmospheric Disturbances in Meteorogical Pictures", *IEEE Trans. PAMI*, PAMI-6, 5, 1984.

[18] M.Brady and B.K.P.Horn, "Rotationally Symmetric Operators for Surface Interpolation", *CVGIP*, 22, pp. 70-94, 1983.

[19] R.W.Brockett, "Gramians, Generalized Inverses and the Least-Squares Approximation of Optical Flow", Proc. IEEE Conf. on Robotics and Automation, pp. 1834-1841, 1987.

[20] I.N.Bronshtein and K.A.Semendyayev, *Handbook of Mathematics*, Van Nostrandt Reinhold Co., 1985.

[21] A.R.Bruss and B.K.P.Horn, "Passive Navigation", *GCVIP*, 21, 3-20, 1983.

[22] F.Cagney and J.Mallon. "Real-Time feature extraction using moment invariants", *SPIE*, Vol. 726, 1986.

[23] N.Cornelius and T.Kanade, "Adapting Optical Flow to measure object motion in reflectance and X-ray image sequences",*Proc. ACM SIGGRAPH/ SIGGART Interdisciplinary Workshop on Motion : Representation and Perception*, Toronto, pp. 50-58, 1983.

[24] K.Cornog, "Smooth Pursuit and Fixation for Robot Vision", M.S.Thesis, MIT AI Laboratory, 1985.

[25] S.A.Dudani, K.B.Breeding and R.B.McGhee, "Aircraft Identification by Moment Invariants", *IEEE Trans. Comp.*, C-26, 1, 1977.

[26] D.H.Foster, "A method for the investigation of those transformations under which the visual recognition of a given object is invariant", *Kybernetik*, 11, pp. 217-222, 1972.

[27] K.S.Fu, "Learning Control Systems and Intelligent Control Systems: An intersection of Artificial Intelligence and Automatic Control", *IEEE Trans. Aut. Contr.*, AC-16, pp. 70-72, 1971.

[28] D.B.Gennery, "Stereo Vision for the Acquisition and Tracking of moving Three-dimensional Objects", in *Techniques for 3-D Machine Perception*, Ed. A.Rosenfeld, North Holland, 1986.

[29] D.B.Gennery, "Tracking Known Three-Dimensional Objects", *Proc. AAAI 2nd Natl. Conf. on A.I.*, Pittsburgh, PA., pp. 13-17, 1982.

[30] J.J.Gibson, *The Perception of the Visual World*, Houghton Mifflin, Boston, 1950.

[31] A.L.Gilbert, "Video Data Conversion and Real-Time Tracking", *IEEE Computer*, pp. 50-56, August 1981.

[32] A.L.Gilbert, M.K.Giles, G.M.Flachs, R.B.Rogers and Y.H.U, "A Real-Time Video Tracking System", *IEEE Trans. PAMI*, PAMI-2, 1, 1980.

[33] H.Goldstein, *Classical Mechanics*, Addison-Wesley Inc., 1980.

[34] G.H.Golub, M.Heath, G.Wahba, "Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter", *Technometrics*, Vol. 21, 2, pp. 215-223, 1979.

[35] E.C.Hildreth, "Computations underlying the measurement of visual motion", *Artificial Intelligence*, 23, pp.309-354, 1984.

[36] J.W.Hilgers, "Non-Iterative Methods for Solving Operator Equations of the First Kind", Ph.D. Thesis, University of Wiskonsin- Madison, 1973.

[37] W.D.Hillis, "The Connection Machine: A computer architecture based on cellular automata", *Physica*, pp.213-218. 1984.

[38] D.D.Hoffman and B.M.Benett, "The Computation of Structure from Fixed-Axis Motion : Rigid Structures", *Biol. Cybern.*, 54, pp.71-83, 1986.

[39] W.C.Hoffman, "The Lie Algebra of Visual Perception", *J. of Math. Psychology*, 3, pp. 65-98, 1966.

[40] B.K.P.Horn, *Robot Vision*, Mc Graw-Hill, 1986.

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

[42] M.K.Hu, "Visual pattern recognition by moment invariants", *IRE Trans.*, IT-8, pp. 179-187, 1962.

[43] A.E.Hunt and A.C.Sanderson, "Vision-Based Predictive Robotic Tracking of a Moving Target", Report of the Robotics Institute, Carnegie-Mellon University.

[44] E.Ito and J.Aloimonos, "Determining Three Dimensional Transformation Parameters from Images: Theory", *Proceedings of the IEEE Conference on Robotics and Automation*, 1987.

[45] R.Kern, U.Kugel, E.Hettlage "Control of a pointing, acquisition and tracking subsystem for Intersatellite Laser Links (ISL^2)", *SPIE*, Vol. 810, Optical Systems for Space Applications, 1987.

[46] G.R.Legters and T.Y.Young, "A Mathematical Model for Computer Image Tracking", *IEEE Trans. PAMI*, PAMI-4, 6, 1982.

[47] H.G.Longuet-Higgins and K.Prazdny, "The interpretation of a moving retinal image", *Proc. R. Soc. Lond.*, B 208, pp. 385-397, 1980.

[48] D.G.Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley Inc., 1984.

[49] D.Marr, *Vision*, W.M.Freeman and Co., 1982.

[50] A.Mitiche and P.Bouthemy, "Tracking Modelled Objects using Binocular Images", *GCVIP*, 32, pp. 384-396, 1986.

[51] S.Nagalia, "Real-Time Acquisition of Objects in Motion", *Center for Automation Rresearch*, CS-TR-1398, 1984.

[52] H.-H.Nagel, "On the Estimation of Optical Flow: Relations between Different Approaches and some New Results", *Artificial Intelligence*, 33, pp.299-324, 1987.

[53] H.-H.Nagel and W.Enkelmann, "An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences", *IEEE Trans. PAMI*, PAMI-8, 5, 1986.

[54] F.O'Sullivan and G.Wahba, "A Cross Validated Bayesian Retrieval Algorithm for Nonlinear Remote Sensing Experiments", *Journal of Computational Physics*, 59, pp.441-455, July 1985.

[55] T.Poggio and C.Koch, "Ill-posed problems in early vision: from computational theory to analog networks", *Proc. R. Soc. Lond.* , B 226, pp. 303-323, 1985.

[56] T.Poggio, V.Torre and C.Koch, "Computer Vision and Regularization Theory", *Nature*, 317, pp. 314-319, 1985.

[57] K.Prazdny, "Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinearly Moving Observer", *CGIP*, 17, pp. 238-248, 1981.

[58] S.A.Rajala, A.N. Riddle and W.L.Snyder, "Application of the One-Dimensional Fourier Transform for Tracking Moving Objects in Noisy Environments", *GCVIP*, 21, pp. 280–293, 1983.

[59] J.W.Roach and J.K.Aggarwal, "Computer Tracking of Objects Moving in Space", *IEEE Trans. PAMI*, PAMI-1, 2, 1979.

[60] A.Rosenfeld, *Picture Processing by Computer*, Academic Press Inc., 1969.

[61] A.Rosenfeld and A.Kak, *Digital Picture Processing*, Academic Press Inc., 1982.

[62] R.L.Savoy, "Contigent Aftereffects and Isoluminance: Psychophysical Evidence for Separation of Color, Orientation and Motion", CVGIP, 37, pp. 3–19, 1987.

[63] G.N.Saridis, "Intelligent Robotic Control", *IEEE Trans. Aut. Contr.*, AC-28, 5, 1983.

[64] R.J.Schalkoff and E.S.McVey, "A Model and Tracking Algorithm for a Class of Video Targets", *IEEE Trans. PAMI*, PAMI-4, 1, 1982.

[65] D.Shulman and J.Aloimonos, "(Non)rigid motion interpretation: A Regularized Approach", *Proc. R. Soc. Lond.*, B 233, pp. 217–234, 1988.

[66] B.G.Schunk, "The Image Flow Constraint Equation", *GCVIP*, 35, pp. 20–46, 1986.

[67] M.R.Teague, "Image Analysis via the general theory of moments", *J. Opt. Soc. Am.*, Vol. 70, No. 3, pp. 920–929, 1980.

[68] D.Terzopoulos, "Multilevel Reconstruction of visual surfaces: Variational Principles and Finite–Element Representation", in *Multiresolution Image Processing and Analysis*, Ed. A.Rosenfeld, Springer–Verlag, pp. 237–310, 1984.

[69] D.Terzopoulos, "Regularization of Inverse Visual Problems Involving Discontinuities", *IEEE Trans. PAMI*, PAMI-8, 4, pp. 413–424, 1986.

[70] A.N.Tikhonov and V.Y.Arsenin, *Solution of Ill-Posed Problems*, Winston, 1977.

[71] S.Ullman, "Analysis of visual motion by biological and computer systems", *IEEE Computer*, 14, (8), pp.57–69, 1981.

[72] P.P.Varaiya, *Notes on Optimization*, Van Nostrand Reinhold Co., 1972.

[73] A.N.Venetsanopoulos and V.Cappellini, "Real–Time Image Processing", in *Multidimensional Systems: Techniques and Applications*, Ed. S.G.Tzafestas, Marcel Dekker Inc., 1986.

CURRICULUM VITAE

Name: Dimitrios P. Tsakiris

Permanent address: 7011A Fordham Court # 2, College Park, Md 20740.

Degree and date to be conferred: M.S., 1988.

Date of birth: May 11, 1963.

Place of birth: Athens, Greece.

Secondary education: Peiramatikon School of the University of Athens, Athens, Greece, 1981.

Collegiate institutions attended:

- 1986-1988: University of Maryland, College Park, Maryland, M.S. in Electrical Engineering, 1988.
- 1982-1986: National Technical University of Athens, Athens, Greece, Eng.Dipl. in Electrical Engineering, 1986.

Professional publications:

- "Visual Tracking Strategies", *M.S. Thesis*, Electrical Engineering Department, University of Maryland, College Park, May 1988.
- "Control Algorithms in Robotics", *Eng. Diploma Thesis*, Electrical Engineering Department, National Technical University of Athens, Athens, Greece, June 1986.

Professional positions held:

- 1986–1988 : Research Assistant, Systems Research Centre, Electrical Engineering Department, University of Maryland, College Park MD 20742.

[74] A.Verri and T.Poggio, "Regularization Theory and Shape Constraints", *MIT AI Memo 916*, 1986.

[75] G.Wahba and J.Wendelberger, "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross-Validation", *Monthly Weather Review*, 108, pp. 36–57, 1980.

[76] T.P.Wallace and O.R.Mitchell, "Analysis of Three–Dimensional Movement Using Fourier Descriptors", *IEEE Trans. PAMI*, PAMI–2, 6, pp. 583–588, 1980.

[77] A.Waxman and K.Wohn, "Contour evolution, neighborhood deformation and global image flow: planar surfaces in motion", *International Journal of Robotics Research*, 4, 3,pp. 95–108, 1985.

[78] G.Westheimer and S.P.McKee, "Visual acuity in the presence of retinal–image motion", *J. Opt. Soc. Am.*, 65, 7, pp. 847–880, 1975.

[79] A.Witkin, "Recovering surface shape and orientation from texture", *Artificial Intelligence*, 17, pp. 17–45, 1981.

[80] B.Wilcox, D.B.Gennery, B.Bon and T.Litwin, "Real–Time model–based vision system for object acquisition and tracking", *SPIE*, Vol. 754, 1987.