

**Spatial and Temporal Grouping
in the Interpretation of
Image Motion**

Harpreet Sawhney

COINS TR92-05

February 1992

This work has been supported in part by the Defense Advanced Research Projects Agency (via TACOM) under Contract Number DAAE07-91-C-R035, and by the National Science Foundation under Grant Number CDA-8922572.

**SPATIAL AND TEMPORAL GROUPING IN THE
INTERPRETATION OF IMAGE MOTION**

A Dissertation Presented

by

HARPREET S. SAWHNEY

**Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of**

DOCTOR OF PHILOSOPHY

February 1992

Department of Computer and Information Science

© Copyright by Harpreet S. Sawhney 1992

All Rights Reserved

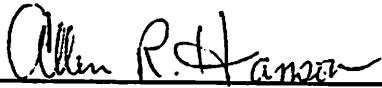
SPATIAL AND TEMPORAL GROUPING IN THE
INTERPRETATION OF IMAGE MOTION

A Dissertation Presented

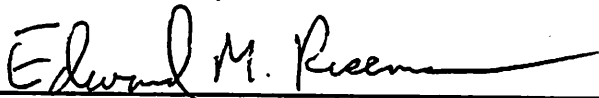
by

HARPREET S. SAWHNEY

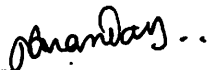
Approved as to style and content by:



Allen R. Hanson, Chair of Committee



Edward M. Riseman, Member



Padmanabhan Anandan, Member



J. MacGregor Smith, Member



W. Richards Adrion, Department Chair
Computer and Information Science

THE UNIVERSITY OF CHICAGO LIBRARY

THE UNIVERSITY OF CHICAGO LIBRARY

to

my

parents

THE UNIVERSITY OF CHICAGO LIBRARY
1215 EAST 58TH STREET
CHICAGO, ILLINOIS 60637
TEL: 773-936-3000
WWW.CHICAGO.LIBRARY.EDU

THE UNIVERSITY OF CHICAGO LIBRARY
1215 EAST 58TH STREET
CHICAGO, ILLINOIS 60637
TEL: 773-936-3000
WWW.CHICAGO.LIBRARY.EDU

...life is an unending quest

and then there is nothingness...

ACKNOWLEDGMENTS

I have gone through many personal ups and downs during the course of this work, but the consistent intellectual and emotional support of Professors Al Hanson and Ed Riseman has been instrumental in bringing me to this stage. My warm thanks to them for their support and for their tireless but friendly prodding — “to be significant in motion vision you have to be *really* significant”. I do not think that I have responded adequately, but then I have learned a lot through the process. Even more gratitude is due to them for their amazing ability to bring together into a research group such a wonderful spectrum of dreamers and hackers as the computer vision group happens to be. I think even Ed and Al themselves are sometimes surprised by the many different possibilities they are led into, but their ability to sustain and nurture a majority of these is commendable. I hope this pluralism flourishes even more. Many thanks to them for largely shielding me from the not always pleasant fund-seeking process. But for the largely non-civilian sources of funding, this setting would almost be ideal.

Without the support and love of my wife, Vindi, this accomplishment would not have been possible. Doing a Ph.D. and starting a family have taken us through some really trying times but it has primarily been the patience and givingness of Vindi that kept life rolling. My loving thanks to her for always being there. And then there is my *mishti*, our Dheerja. Many playful thanks to her for constantly reminding me not to take life always seriously, but play *chutes and ladders* sometimes!

The untiring and selfless efforts of my father Mr. Mohan Singh and my late mother Ms. Darshan Kaur are behind all my accomplishments to this day. They did not spare any effort in providing me a wholesome moral, intellectual and material life. My gratitude and reverence to them for all they have done. My mother's dream

was to see me comfortably settled. Unfortunately, she did not live to see this day, but her ethereal presence will soothe and guide me forever.

Graduate life would have been so much less colorful without the presence of Teddy, Renee and Teddy&Renee. A 'bearful' of thanks to Teddy for sharing so many pleasant and some not-so-pleasant moments of life in the past six years — from understanding the intricacies of the SVD to sharing letters of reference! To Renee for making everything look simple. And to Teddy&Renee for their tremendous support through thick and thin as members of our extended family here. Thanks also to Harmeet and Kathy for their care and support.

My sincere thanks to Anandan for always trying to bring out the best in me and for his unceasing professional support. Thanks also to Prof. Jim Smith for introducing me to optimization and computational geometry.

Lance, the dreamer, helped evolve many ideas in this thesis. Thanks Lance for sharing many 'visions'. Thanks also to Brian for always helping possibilities emerge, to Manmatha for stimulating discussions, and to JohnO for patiently explaining formalisms.

Thanks to Bob Collins, Ross, Bruce, Poornima, JohnD, Chris, Martin, Mike, Inigo, Rabi, Rich Weiss, Mark Snyder, Val and Jonathan for various fruitful discussions. Without the ever available support of that wonderful Bob Heller, I would still be at it!

Thanks to an anonymous reviewer of one of my papers whose one line description of the instability of conic fitting I particularly liked and have used it in a chapter.

Finally, my gratitude to LaurieW and Janet for keeping the contracts rolling and for all kinds of help.

ABSTRACT

SPATIAL AND TEMPORAL GROUPING IN THE INTERPRETATION OF IMAGE MOTION

FEBRUARY 1992

HARPREET S. SAWHNEY

B.TECH., INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

M.TECH., INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

PH.D., UNIVERSITY OF MASSACHUSETTS

Directed by: Professor Allen R. Hanson

Interpretation of the three-dimensional (3D) world from two-dimensional (2D) images is a primary goal of vision. Image motion is an important source of 3D information. The 3D motion (relative to a camera), and the 3D structure of the environment manifest themselves as image motion. The primary focus of this thesis is the reliable derivation of scene structure through an interpretation of motion in monocular images over extended time intervals.

An approach is presented for the integration of spatial models of the 3D world with temporal models of 3D motion in order to robustly derive 3D structure and perform tracking and segmentation. Two specific problems are addressed to illustrate this approach. First, a model of a class of 3D objects is combined with smooth 3D motion to track, identify and reconstruct *shallow* structures in the scene. Second, a specific model of 3D motion along with general spatial constraints is employed for 3D reconstruction of motion trajectories. Both parts rely fundamentally upon the quantitative modeling of *common fate* of a structure in motion.

In many man-made environments, obstacles in the path of a mobile robot can be characterized as *shallow*, i.e. they have relatively small extent in depth compared to the distance from the camera. Shallowness can be quantified as *affine describability*. This is embedded in a tracking system to discriminate between shallow and non-shallow structures based on their *affine trackability*. The temporal evolution of a structure, derived using affine constraints, is used for verifying its identity as a shallow structure and for its 3D reconstruction.

Spatio-temporal analysis and integration is further demonstrated through a *two-stage* technique for the reconstruction of 3D structure and motion of a scene undergoing a rotational motion with respect to the camera. First, the spatio-temporal grouping of discrete point correspondences as a *set* of conic trajectories in the image plane is obtained, by exploiting their common motion. This leads to a description that is reliable when compared to the *independent* fitting of trajectories. The second stage uses a new closed-form solution for computing the 3D trajectories from the computed image trajectories under perspective projection.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS	vi
ABSTRACT	viii
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
 Chapter	
1. INTRODUCTION	1
1.1 Goal 1 : From Image Motion to 3D Structure and Motion	2
1.2 Goal 2 : Tracking and Segmentation	3
1.3 Goals of the Thesis	5
1.4 Accomplishments of the Thesis	6
1.4.1 Scene Reconstruction through 'Shallow' Structures	6
1.4.2 Description and Reconstruction of Rotational Motion through Image Trajectories	9
1.5 Summary	10
1.6 Contributions of the Thesis	12
1.7 Organization of the Presentation	12
2. CURRENT APPROACHES	13
2.1 Goal 1 : Computation of 3D Structure and Motion	15
2.1.1 The Rigidity Constraint	15
2.1.2 Discussion on Two-Frame Motion Analysis	19
2.1.3 Structure Computation	20
2.1.4 Discussion on Multi-Frame Structure Estimation	23
2.1.5 Reconstruction Using Models of 3D Motion	24

2.1.6 Discussion of Motion Analysis with Motion Models	25
2.2 Goal 2 : Object, Boundary and Surface Segmentation	26
2.2.1 Multiple Motion Segmentation	27
2.2.2 Boundary Segmentation	28
2.2.3 Surface Segmentation	28
2.2.4 Discussion	29
2.3 Goal 3 : Feature and Object Tracking	29
2.4 Relationships to Our Work	31
2.4.1 Structure Reconstruction	31
2.4.2 Segmentation and Tracking	32
2.4.3 Motion-model Based Reconstruction	33
3. TRACKING, IDENTIFICATION AND RECONSTRUCTION OF SHALLOW ENVIRONMENTAL STRUCTURE	35
3.1 Overview	38
3.1.1 Affine Describability and Trackability	39
3.1.2 Tracking Algorithm	39
3.1.3 Experimental Results	40
3.2 Relationship to Previous Work	41
3.3 Shallowness as Affine Describability	43
3.4 Does Affine Describability Imply Shallowness ?	45
3.4.1 General 2D Affine Transformation	46
3.5 Solving for Affine Parameters and Their Covariances	47
3.5.1 Modeling Uncertainties in Image Lines	50
3.5.2 Covariances of the Affine Parameters	50
3.6 Aggregate Structure Representation and Matching	52
3.6.1 Representing Lines and Aggregate Structures	53
3.6.2 Model Matching with Measurement and Prediction Errors	54
3.7 Shallowness as Affine Trackability	59
3.7.1 Cycle of Prediction and Matching	60
3.7.2 The Algorithm	65
3.8 Experimental Results	69

3.8.1 Tracking Results	69
3.8.2 Segmentation and Reconstruction Results	83
3.9 Summary	93
4. DESCRIPTION AND RECONSTRUCTION FROM IMAGE TRAJECTORIES OF ROTATIONAL MOTION	101
4.1 Previous Work	102
4.2 Overview of the Approach	105
4.3 The 3D Estimation Problem	107
4.3.1 Formulation	108
4.3.2 Solution	112
4.3.3 Multiple Solutions	115
4.4 Image Trajectory Description	121
4.4.1 Independent Conic Fitting	121
4.4.2 Grouping Algorithm	126
4.5 Experimental Results	133
4.5.1 Trajectory Grouping Results	135
4.5.2 3D Estimation Results	142
4.6 Summary	148
5. SUMMARY AND FUTURE WORK	150
5.1 Major Contributions	150
5.2 Directions for Further Research	152
APPENDICES	
A. SOLUTION FOR THE ROTATIONAL TRAJECTORY WHEN THE AXIS PASSES THROUGH THE ORIGIN	155
B. CONIC CURVE DESCRIPTION	157
B.1 Algebraic Distance Measure	158
B.2 First Order Distance Measure	160
BIBLIOGRAPHY	163

LIST OF TABLES

Table	Page
3.1 Depth results for the <i>cones-seq</i>	97
3.2 Depth results for the <i>room-seq-1</i>	98
3.3 Depth results for the <i>comp-seq</i>	100
4.1 3D distance comparisons for the <i>room-seq-2</i>	143
4.2 Independent vs. grouped fit 3D distances for the <i>room-seq-2</i>	144
4.3 Two-frame vs. trajectory 3D distances for the <i>room-seq-2</i>	145
4.4 Independent vs. grouped fit 3D depths for the <i>box-seq</i>	147
4.5 Two-frame vs. trajectory 3D depths for the <i>box-seq</i>	147

3.13	Tracking of a shallow 4-line structure in the <i>room-seq-1</i>	84
3.14	Motion of a sample structure in the <i>room-seq-1</i>	86
3.15	Tracking of a shallow triple in the <i>room-seq-1</i>	87
3.16	Tracking of an independently moving object	90
3.17	Shallow structures identified in the <i>cones-seq</i>	95
3.18	Shallow structures identified in the <i>room-seq-1</i>	96
3.19	Labelled objects in the <i>cones-seq</i>	97
3.20	Labelled objects in the <i>room-seq-1</i>	98
3.21	Two image frames of the <i>comp-seq</i>	99
3.22	Labelled objects in the <i>comp-seq</i>	100
4.1	The stages of processing of rotational trajectories	106
4.2	Geometry of a point's rotation.	109
4.3	Four solutions corresponding to $+d$	116
4.4	Four solutions corresponding to $-d$	117
4.5	Two distinct solutions for one point.	118
4.6	Unique solution as the common solution of two points	119
4.7	Sample independent conic fits for the <i>box-seq</i>	124
4.8	Sample independent conic fits for the <i>room-seq-2</i>	125

LIST OF FIGURES

Figure		Page
2.1	The Coplanarity constraint	17
2.2	The multi-frame bundle of rays	22
3.1	A Hallway scene with shallow and non-shallow structures	37
3.2	The parallel and perpendicular error components	48
3.3	The model for noise in lines	51
3.4	Bootstrap matching using flow	61
3.5	Four image frames of the <i>cones-seq</i>	70
3.6	Four image frames of the <i>room-seq-1</i>	71
3.7	Motion of the doorway lines in the <i>cones-seq</i>	73
3.8	Tracking the doorway in the <i>cones-seq</i>	74
3.9	Tracking of a shallow triple in the <i>cones-seq</i>	76
3.10	Non-trackability of a non-shallow triple	79
3.11	Longitudinal error in affine projection of a non-shallow structure	81
3.12	Longitudinal error in affine projection of a shallow structure	81

4.9	Conic section parameterization illustrated for an ellipse	127
4.10	Case of small overlap along projections	130
4.11	Case of large overlap along projections	130
4.12	Case of no overlap and small gap	131
4.13	Case of no overlap but a large gap	131
4.14	Two frames of the <i>room-seq-2</i>	136
4.15	Displacement field for the <i>room-seq-2</i>	136
4.16	Two frames of the <i>box-seq</i>	137
4.17	Displacement field for the <i>box-seq</i>	137
4.18	Tracked lines for the <i>room-seq-2</i>	138
4.19	Sample point tracks for the <i>room-seq-2</i>	138
4.20	Tracked lines for the <i>box-seq</i>	139
4.21	Sample point tracks for the <i>box-seq</i>	139
4.22	Combined conic fits for the <i>room-seq-2</i>	140
4.23	Combined conic fits for the <i>box-seq</i>	141
B.1	Hyperbolic and elliptic level curves	161

CHAPTER 1

INTRODUCTION

The future success of autonomous robots depends on the intelligent use of vision. A primary goal of vision is to derive interpretations of the three-dimensional (3D) world from two-dimensional (2D) images. Variations in the 2D images embody information about the structure of the 3D world. Motion of the imaging sensor and of objects in the environment is an important source of these variations. Deriving descriptions of 3D relationships in the environment through the interpretation of image motion is an important problem in robot vision. The primary focus of this thesis is on the reliable derivation of scene structure through an interpretation of motion in monocular images over extended time intervals. Structure is defined in terms of spatial and temporal models; the output descriptions are in terms of these models.

Given a single static image, an infinite number of plausible inferences about the imaged 3D environment can be drawn. Work in static image understanding has concentrated on inventing a plausible set of constraints to impose on the world and the imaging process to make the problem well-posed. However, metric relationships between the sensor and the scene cannot be derived without resorting to specific domain knowledge about the scene. Furthermore, certain coincidences of viewpoint cannot be resolved from a single view alone. The availability of a second view of the same scene from a different vantage point can dramatically reduce the number of solutions to a small finite set. Multiple views (more than two) can further aid in deriving the 3D information. Moreover, both quantitative and qualitative 3D

information can be derived from these motion-based techniques without requiring excessive domain knowledge about the scene.

In spite of the importance of three-dimensional interpretation from motion, deriving reliable scene structure from image motion has proven to be an elusive goal, due to a variety of practical factors. The problem occurs both in defining the goals for the 3D interpretation and in choosing a set of constraints and solution methods to meet the goals. Some of the important goals for image motion interpretation are discussed briefly in the following sections.

1.1 Goal 1 : From Image Motion to 3D Structure and Motion

One popular goal for the interpretation of image motion has been the 3D reconstruction of features in the scene based on their measured motion in the image plane. Within this problem definition, a large effort has been expended on what has come to be known as the *two-frame* structure-from-motion problem. That is, given two views of a scene from different vantage points, the goal is to compute the relative motion between the scene and the camera, and the 3D coordinates of the scene features. It is assumed that the correspondences between pairs of image features for the same scene feature in the two views have already been established, and that there is a single relative rigid motion between the camera and the scene. Thus, the transformation between the two views is constrained to be a rigid body transformation. In fact, rigidity is the minimal constraint that can be imposed to reduce the number of 3D solutions to a finite set given two views of a scene.

The rigidity constraint is attractive because it potentially allows 3D reconstruction without apparently resorting to any more knowledge about the scene and the

motion. However, algorithms based on this constraint alone, lack robustness in practical situations where:

- the field-of-view (FOV) and resolution of the imaging system are limited,
- the translational motion between views is limited, and
- the image measurements are corrupted with noise.

The computation of the 3D motion parameters, independent of the particular algorithm used, is inherently ambiguous ([3, 94]). This has limited the practical application of the two-frame techniques.

The rigidity constraint, coupled with specific models of smooth motion, provides another framework for the goal of 3D reconstruction. This approach has been applied to many frames of image data related through the motion models. This idea has had relatively more success because smoothness can be used to relate motion measurements over time. Many viewpoints of the same scene provide robustness in the presence of noise. However, the problems of validation of the assumed models, and of reliably tracking objects over time have not been addressed adequately.

1.2 Goal 2 : Tracking and Segmentation

The goals of segmenting a scene into semantically useful object hypotheses, handling multiple motions, and maintaining correspondences of objects over time are also important for an autonomous robotic system. As was discussed earlier, in addressing the problem of reconstruction of 3D motion and structure from image motion, it has largely been *assumed* that these segmentation and correspondence problems have been solved. The reconstruction problem is hard enough on its own; hence relatively little research has been done towards integrating segmentation, tracking and reconstruction.

Segmentation of a scene into surfaces and objects on the basis of similarity of their range from the camera, or other common structural properties, is useful for many robotic tasks. This has generally been viewed as a post-processing step which comes after the 3D coordinates of primitive features like points and lines have been computed. The need for this grouping cannot be overlooked for tasks like obstacle avoidance and grasping, where it might be necessary to represent the 3D structure in terms of surfaces, volumes and their boundaries. Given that the first step of recovering and reconstructing features, such as points and lines, can be quite erroneous, constructing aggregate structures (surfaces/objects) out of these will be difficult, to say the least. On the other hand, if the process of scene reconstruction from image motion is constrained early on by using general structural properties of scenes, it may be possible to achieve robustness.

In dynamic scenarios, it is inefficient (assuming it could even be done reliably) to segment objects and surfaces from frame to frame, independent of each other. Smoothness of motion in most practical situations provides a natural dynamic constraint for tracking objects over time and hence maintaining their identity. Most approaches for this task either use 3D-3D constraints [99] or 2D-2D [27, 30] constraints. That is, either the reconstructed 3D structure is tracked under 3D motion constraints or 2D image features are tracked with purely 2D constraints on their motion. In the former case, stereo data or some other knowledge of 3D structure is present (e.g. range data). For image plane tracking, heuristics on the uniformity of image motion are employed.

Finally, when using specific models of motion, approaches in the past have directly computed the 3D structure and motion parameters from primitive image measurements (point/line correspondences) [19, 81]. The work described here presents an alternative approach by building descriptions of image motion based on the assumed

model of motion, verifying the model as the descriptions are built. Potentially, departures from the assumed model, either due to multiple object motions or due to the presence of gross outlying errors in the data, can be handled within this alternative framework.

In our work, we show how general constraints on 3D structure and motion can be employed to track and segment objects based on expectations of their coherent behavior in the image plane. Generic 3D constraints are used to model image motion of objects for tracking, which in turn is used as a verification for the validity of the constraints.

1.3 Goals of the Thesis

The goal of our work is to develop robust algorithms for the derivation of scene structure from image motion in a way that is useful for tasks like obstacle avoidance and 3D model building. Towards achieving that end, we demonstrate the strength of combining both spatial and temporal constraints in dynamic images to obtain grouping of image information into coherent 3D structures. With few exceptions, approaches in the literature have addressed each of the problems related to 3D interpretation of image motion, that were outlined in the previous sections, in isolation. An objective of this work is to develop techniques that provide an integrated framework for robust tracking, segmentation and 3D reconstruction.

Specifically, this thesis shows how spatial models of the 3D world and temporal models of 3D motion can be employed to compute:

- robust 3D reconstruction of *aggregate structures* rather than just individual point or line features,
- robust correspondence of aggregate structures over time,

- a semantically useful segmentation of the scene in terms of 3D structures, and
- 3D trajectories of features from image trajectories of rotational motion.

It is to be emphasized that the models being referred to are not object-specific models or models for a specific scene. The models are of structure and of motion that are valid for a wide variety of scenes.

Inherent in the specific algorithms developed here are mechanisms for deciding the validity of the constraints. These algorithms demonstrate that posing the problem of 3D reconstruction as an integrated grouping and parameter estimation problem addresses the issues of both model validation and reliable reconstruction within the constraints of the validated model.

1.4 Accomplishments of the Thesis

We have selected two demonstrative problems in scene reconstruction to highlight the general approach:

1. Scene reconstruction through 'shallow' structures.
2. Description and reconstruction of rotational motion through image trajectories.

1.4.1 Scene Reconstruction through 'Shallow' Structures

One key problem in computer vision is that of segmenting an 'object' from its 'background'; this is known as the figure-ground segregation problem. Objects have been variously defined in terms of the uniformity of texture, surface smoothness and similarity of image motion. We observe that in man-made environments, many potential obstacles can be characterized as being *compact*; that is, the features in an object are close to a nominal central point or an axis in contrast with larger scale

structures in the background. This definition can be used to represent potential obstacles as compact structures in the path of a robot. This property can be quantified by observing that such objects have limited extent along the direction of view compared to the distances at which an autonomous robot should be capturing them in its internal representation, that is, these objects are 'shallow'. Thus, we wish to achieve the representation of a scene for a mobile robot in terms of shallow structures in the environment that are distinct from the background. In other words, the assumption is that for a mobile robot heading directly into the environment, structures that face it head-on are relevant for obstacle avoidance and path planning. The approach presented here makes this assumption explicit and uses it quantitatively.

Tracking Shallow Structures

The image projections of a shallow structure in motion relative to the camera, when sampled closely in time¹, are related through a 2D affine transformation. The motion could be due either to camera motion and/or to object motion. The 3D location and the image motion of a shallow structure is completely characterized by the affine transformation and the corresponding 'average' 3D depth of the structure. This property can be exploited to track, segment and reconstruct shallow structures. If the 3D motion is assumed to be smooth, the temporal evolution of shallow structures can be used for their segmentation and 3D reconstruction.

We have embedded a model of shallow structures undergoing smooth motion in a dynamic tracking framework. The image motion of hypothesized structures is instantiated and predictions of these structures are matched to newly acquired data. Tracking has to account for measurement errors in the image data and modeling

¹Relative to the magnitude of motion.

errors due to departures from the assumed models of 3D motion and structure. Two techniques in the literature have been adapted for this tracking:

1. **Prediction:** Kalman filtering is used for recursive state and covariance prediction, and estimation of the dynamically changing state of a hypothesized structure, and
2. **Matching:** the covariance normalized Euclidean distance (Mahalanobis distance) is used to compute the error between the predictions of a structure and its potential matches in the data of a newly acquired image frame.

These techniques also handle modeling errors in the dynamic models and measurement errors in the data.

Tracking Aids Segmentation and Reconstruction

Consistency of the temporal predictions and the 3D description as determined by tracking are used as criteria for labeling a hypothesized structure as shallow or otherwise. In other words, maintenance of an object's identity over time, within the constraints defined by the model of shallowness, is used for inferential leverage in identifying the object as shallow. The essential idea is that if a hypothesized structure can be consistently tracked and its 3D depth over time is consistent with a shallow structure model, then the structure is identified as shallow, otherwise it is labeled non-shallow. Thus, *affine trackability* leads to segmentation and 3D reconstruction in this demonstration of the integration of a general spatial model and its temporal continuity derived from the assumption of smoothness of its 3D motion.

The output representation for each shallow object is as a fronto-parallel plane (*cardboard-cutout*) placed at the computed 3D location. Such a representation of aggregate structures, although only an approximation of an object's structure, might

still be adequate for path planning and obstacle avoidance. Robustness in the computed 3D structure is a result of the use of an aggregate structure (e.g. a surface or a group of surfaces), rather than the traditional use of more primitive features like individual points and lines. Furthermore, an explicit step of the computation of 3D motion as a first step towards the computation of 3D structure is not required, thus avoiding some of the inherent instabilities in the general structure-from-motion problem that will be discussed in the next chapter.

1.4.2 Description and Reconstruction of Rotational Motion through Image Trajectories

The second demonstration of the use of spatial and temporal grouping is in the reconstruction of 3D trajectories of features in a scene whose motion is induced either by a fixed-axis rotation of the camera or of objects in the scene. This is motivated by the problem of acquiring a 3D model of a scene through a constrained motion of a robotic hand/head carrying a camera.

The reconstruction process consists of two stages:

1. Description of the motion of image features as curved trajectories, and
2. Reconstruction of the trajectories in 3D from the 2D image trajectories.

The 3D parameters are computed from the continuous trajectories that describe the image motion of discrete image features. This is in contrast with methods that directly compute the 3D parameters from discrete feature correspondences. The image trajectories are intermediate descriptors of the 3D motion. The similarity of the motion parameters across many features is made explicit by the trajectories. An explicit step of describing the motion of discrete image features as curved trajectories (conics) in the image plane is potentially useful for segmentation of multiple rotations and rejection of gross errors in the image data, in addition to robust 3D reconstruction.

Image Trajectories

Describing conic curves, especially when only short segments of the underlying points of the 3D trajectory are imaged, is known to be unstable. In general, conics with widely varying parameters can be fit to these short segments, each of which describes the image data equally well. We address this problem using a grouping constraint in an algorithm that exploits spatial proximity of features and the common motion constraint. The grouping constraint relates parameters of different conics based on a locally orthographic assumption. The grouping algorithm embeds this constraint in an incremental hypothesize-and-verify technique to fit the conics. The algorithm achieves a combined description of a *set* of the tracked image features rather than for each of the features independently.

3D Trajectories

A new closed-form solution, under perspective projection, has been developed to compute the 3D trajectories in the scene from the computed image trajectories. The particular choice of parameterization of the 3D geometry and motion leads to a solution which is simpler than those existing in the literature.

In summary, spatial proximity of points and their temporal evolution under rotational motion is translated into a grouping constraint for the description of the corresponding curved trajectories in the image plane. The 3D description is then derived from the image trajectories.

1.5 Summary

The previous sections have summarized two demonstrations of our spatial and temporal grouping approach to the 3D interpretation of image motion. The solution

to the problems addressed above arise from the application of a general paradigm for the 3D interpretation of image motion using both spatial and temporal grouping. The two are related in a common conceptual framework. The endeavor has been to formalize the notion of *common fate* of a structure under motion. Common fate as a cue to various kinds of segmentations has been used both in computational vision and psychophysics. In our case, shallow structures and the resulting coherence in their image motion, and the coherence of proximal image features under rotational motion, are used as common fate constraints for grouping and reconstruction.

The two algorithms which we present can be potentially combined in a single application. Consider a scenario where a eye-on-hand robot is to steer through a cluttered environment and then grab a specific object whose detailed 3D model is not available. A coarse representation of the environment in terms of shallow structures can facilitate steering without hitting obstacles. Once the vicinity of the goal is reached, a 3D model of the object to be grasped can be acquired through a constrained rotational motion. The advantage of our approach is that even when the object is observed from a small number of viewpoints due to limited availability of free space, reliable structure can still be derived through grouped trajectories. Rotational motion provides the added advantage of being able to observe an object from a relatively large set of disparate viewpoints even when free space is constrained, thus effectively providing a large baseline. However, combining the two algorithms into a working system demands further research into areas of active vision and goal-directed model building.

1.6 Contributions of the Thesis

The following summarize the major contributions of this thesis:

1. Robust scene reconstruction as aggregate spatio-temporal structures – shallow structures and 3D trajectories – using spatial and motion models.
2. Object tracking motivated by 3D motion and structure in contrast with traditional multi-frame token tracking which utilizes heuristics about image motion.
3. Aggregate structure matching in contrast with point/line matching provides figural constraint for relatively unambiguous matching.
4. Tracking as an aid to segmentation and reconstruction.
5. A grouping algorithm for robust description of combined conic trajectories.
6. A new closed-form solution for the problem of deriving a 3D rotational trajectory from a 2D conic obtained under perspective projection.

1.7 Organization of the Presentation

The next chapter is devoted to a critical analysis of major approaches to 3D reconstruction and segmentation based on monocular motion. In addition to the analysis, significant points of departure and a justification of the approach adopted for the problems addressed in this thesis are also presented. Chapter 3 presents a description of tracking and identification of shallow structures. Chapter 4 details the work on image description and 3D interpretation of rotational motion through image trajectories. Finally, in Chapter 5 we reiterate the salient contributions and the lessons learned through this work, and then go on to present directions for future research.

CHAPTER 2

CURRENT APPROACHES

There is a vast body of literature that addresses problems of three-dimensional (3D) interpretation from image motion. A review of this entire body of work will not be attempted here. We will briefly categorize various approaches in terms of the issues addressed and the techniques and inputs used. A few commonly addressed problems and the constraints used to solve them will be presented to bring out their relationship to the issues and the techniques adopted in this thesis. For a more comprehensive overview of the literature, see Barron's [11] and Aggarwal and Nandhakumar's [5] surveys.

There are three broad goals for the interpretation of three-dimensional (3D) scenes from image motion:

1. Computation of the relative 3D motion between the camera and the environment, and the 3D structure of the environment.
2. Delineation of objects and surfaces on the basis of similarity of their motion and structure.
3. Tracking of objects and other entities in the scene which might be in relative motion with respect to the camera due to ego-motion and/or independent motion.

This review bases its classification of various approaches according to the goals they address. A large amount of work in motion understanding has been devoted to the

recovery of 3D motion and the reconstruction of 3D structure, whereas the work on tracking and on segmentation of semantically important events is scant. With few exceptions, approaches in the literature have addressed each of the above problems in isolation. One goal of this review is to motivate the development of techniques in which tracking, segmentation and reconstruction could be made robust through an integrated framework.

In addition to a goal-based classification, the different approaches can also be characterized in terms of:

1. the type of image measurements or image data used,
2. the number of frames used, and
3. the constraints employed to relate the measurements spatially and temporally.

Most approaches use one of the following three inputs as the measurements of image motion:

1. optical flow;
2. spatio-temporal derivatives of image intensities (the direct approach);
3. correspondence of symbolic tokens, such as lines and points, over time.

The first two measurements use temporal derivatives of image intensities and hence require image frames which are sampled closely in time. In this thesis, symbolic tokens and their image motion have been chosen as the measurements. Consequently, this review is largely limited to those which use token correspondences. For a review of approaches involving other types of measurements, refer to [36].

2.1 Goal 1 : Computation of 3D Structure and Motion

The goal within the 3D reconstruction paradigm is to reliably compute the 3D structure and motion of the scene from the measured image motion. Most approaches assume that image motion measurements (feature correspondences or flow fields) along with their associated confidence measures are already available.

In order to derive the three-dimensional (3D) information from the measured two-dimensional (2D) image motion, constraints relating the image motion to the 3D motion and structure have to be imposed. The constraints are used to derive equations that govern the relationship of the motion of image features to the 3D motion and structure parameters. Error measures are then derived from the governing equations. Optimization of the error measures using parameter estimation or search techniques leads to the solution for the 3D parameters. Various approaches can be grouped together into two broad categories depending on the type of constraints used:

1. The Rigidity constraint.
2. Models of 3D motion.

2.1.1 The Rigidity Constraint

The rigidity constraint assumes that the image motion is a result of a rigid-body motion between the camera and the scene across two time instants. The 3D motion that relates the 2D measurements at any two time instants can be represented as a rotation around an axis that passes through the origin and a translation in the camera coordinate system at one of the instants. It is assumed that this constraint is valid over large patches in the image (or over all of the image). Image measurements in these patches are then used to constrain the solution for the underlying 3D motion parameters. Once the motion parameters have been computed, the scene features

corresponding to the measured image features can be reconstructed if the translation is non-zero. It is to be noted that for most of the approaches based on the rigidity constraint alone, the "3D structure" means the 3D coordinates of points and lines in a suitably chosen coordinate system. Usually no surface or object level properties are represented.

Motion Parameters

The rigidity constraint is used in various ways to formulate error measures. Three representative approaches have been chosen for this review because they highlight the trade-off between computational convenience and reliability of the output parameters.

The E Matrix Approach

Given that a point in the scene is observed from two different vantage points, the view rays from the two views and the translation vector (baseline) joining the origins of the two camera positions are coplanar as shown in Figure 2.1. This is called the *coplanarity constraint*. It can be expressed as:

$$\mathbf{t} \bullet (R\mathbf{r}_1 \times \mathbf{r}_2) = 0 \quad (2.1)$$

where \mathbf{t} is the translation vector, R is the rotation matrix, \mathbf{r}_1 is a view ray in the first frame and \mathbf{r}_2 the corresponding ray in the second frame. The above equation can also be written as:

$$\mathbf{r}_2^T T_X R \mathbf{r}_1 = 0 \quad (2.2)$$

where T_X is the matrix operator for the cross-product operation with the vector \mathbf{t} . The 3×3 product matrix $T_X R$ is called the *Essential Matrix* or the E-matrix.

Tsai and Huang [90] first compute the E-matrix using eight or more image correspondences. The rotation and translation parameters are then derived using

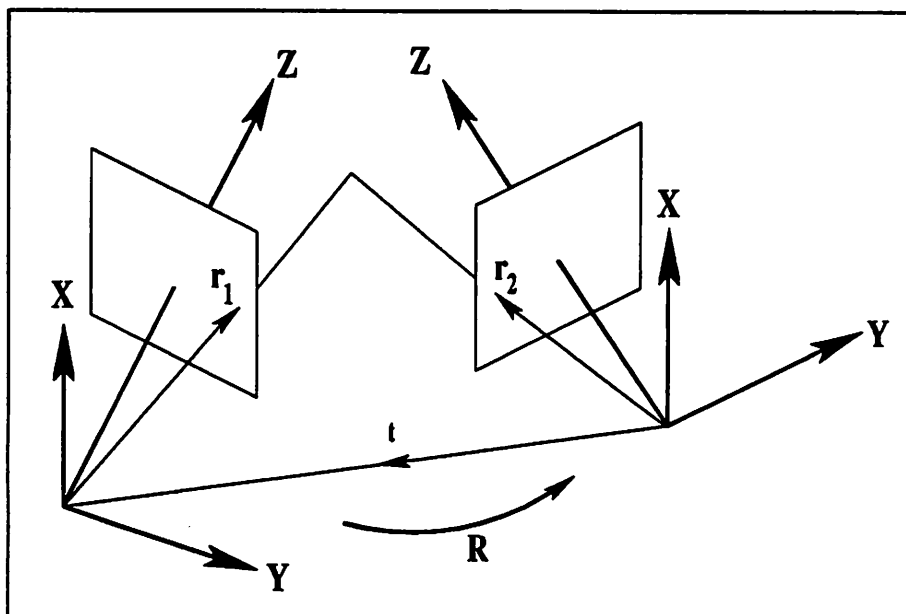


Figure 2.1: The Coplanarity constraint.

a Singular Value Decomposition (SVD) of the E -matrix. Longuet-Higgins [56], Faugeras et al. [33], Zhuang et al. [100], and Weng et al. [94] also first compute the E -matrix but subsequently use different methods for its decomposition into the motion parameters.

The methods based on the E -matrix are efficient in that they are non-iterative, but the computed motion parameters are very sensitive to errors in the input point correspondences [39, 82]. This is due to a fundamental flaw in the E -matrix approach. When solving for the E -matrix, all its components are assumed independent. This results in a least-squares problem which is linear in the matrix components. But there exist dependencies between the constituent terms of the E -matrix because it is the product of an anti-symmetric matrix and an orthonormal rotation matrix. Weng et al. [92] show that a 3×3 matrix, E , satisfies the requirements for being an essential matrix if and only if one of the eigenvalues of EE^T is zero and the other two are equal.

When the E-matrix is computed by ignoring this constraint, another matrix has to be found which is nearest to the computed matrix and also satisfies the constraints [42]. The rotation and translation corresponding to this constrained matrix can be far from the true ones even with moderate amounts of noise in the image measurements. Thus, the E-matrix approaches have been largely of theoretical interest only.

Solution of R and t Directly

Horn [43] directly solves for the translation and rotation using the departure from coplanarity (Figure 2.1) as an error measure. The quaternion representation for rotations is used. Starting from an initial guess, the resulting constrained¹ non-linear least squares problem is solved using the Gauss-Newton [38] method. It is suggested that initial guesses can be automatically generated by sampling the space of rotations, and the solution(s) with the least residual error can be chosen.

Bruss and Horn [21] and Adiv [2] compute the motion parameters differently. Assuming small rotations between two time instants, the displacement vectors in the image plane are represented as a function of the motion parameters and the image coordinates²:

$$\begin{bmatrix} X' - X \\ Y' - Y \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} -\frac{X'Y}{f} & (f + \frac{XX'}{f}) & -Y \\ -(f + \frac{YY'}{f}) & \frac{X'Y}{f} & X \end{bmatrix} \Omega + \frac{1}{z} \begin{bmatrix} f & 0 & -X' \\ 0 & f & -Y' \end{bmatrix} t \quad (2.3)$$

where α and β are the x and y components of the displacement vectors, f is the effective focal length, (X, Y) and (X', Y') are the two image projections of the same point in the scene whose depth is z , and Ω and t are the rotation and translation vectors. The z -axis points along the optical axis of the camera into the scene and the x and y axes are parallel to the image plane.

¹The translation vector is constrained to be of unit magnitude due to the scale ambiguity between its magnitude and the range of points in the scene.

²These equations are a modified form [98] of the authors' image velocity equations.

The motion parameters, Ω and t , are computed by minimizing the sum of squared errors between the measured displacement vectors and the analytical ones given by Equation 2.3. First, the depth z is eliminated. The resulting error measure is linear in the rotation parameters if the translation is known. Bruss and Horn propose an iterative gradient-descent technique for the minimization, whereas Adiv employs a systematic search of the two-dimensional space of unit translation vectors to find the motion parameters which correspond to the least residual error. Heeger and Jepson [39] use the same error measure as well as a sampling of the space of unit translations. However, they add the residual errors obtained from many small patches all over the image instead of computing them globally. The advantage is that for each small patch and each direction of translation, some computations can be precomputed and the rest can be done in parallel.

Longuet-Higgins and Prazdny [57], and Koenderink and van Doorn [49] were among the first to derive the differential form of Equation 2.3 for image velocities. They analyzed the structure of the image flow fields at length and showed how the 3D motion and structure parameters could be derived using motion parallax ([72]), and also using the derivatives of the flow field.

The techniques described above lead to an improvement in the computed motion parameters compared to the E-matrix methods [39, 82]. However, limitations due to certain inherent ambiguities cannot be overcome by any of these approaches; this is discussed presently.

2.1.2 Discussion on Two-Frame Motion Analysis

The two-frame techniques constrain the solution of the rigid-body rotation and the translation using motion measurements from the whole image or a substantial part of it. However, the success of these depends critically on having a large

field-of-view camera, a significant amount of translation and significant variations of depth in the scene [2].

Spetsakis and Aloimonos [82] show that techniques which use the coplanarity constraint as an error measure encounter a local minimum. The resulting translation vector points approximately towards the centroid of the image region under consideration for any arbitrary true translational vector. The residual for this local minimum can be as low or lower than that of the minimum corresponding to the correct solution [78]. However, this problem can be handled by normalizing the error measure appropriately ([2, 29, 82]).

A more severe problem is the ambiguity between translations parallel to the image plane and rotations in depth. This problem is severe when the translations are parallel to the image plane, there is noise in the image data and the field-of-view is small (less than about 40°) [79, 94]. In such situations, differences in the image motion due to the rotational and the translational components of the 3D motion are small almost everywhere in the image plane. This problem could be alleviated to a certain extent by increasing the resolution of the imaging system. However, within the current practical limitations, it is a serious problem.

2.1.3 Structure Computation

The unreliability in the computation of motion parameters, at least for the cases of inherent ambiguities, can seriously affect the estimation of the 3D structure of the scene. Each image correspondence or displacement vector along with the computed motion parameters leads to its own depth measurement. This estimate can be very erroneous, in general, because not only does it depend on a single image measurement but also on the unreliable estimate of motion. Thus, a natural extension of these techniques is to improve the structure estimate by considering more image frames in which a large part of the scene remains visible over the sequence.

Multi-frame Structure

In principle, multi-frame estimation of structure would incorporate tracking of the scene features over time, in order to maintain their correspondence. However, reliable solutions of scene structure, even with the correspondences assumed, has proven to be hard. The integration of structure estimation and tracking in the domain of monocular motion analysis has rarely been attempted.

One way to use multiple frames is within a dynamic refinement of the estimate of structure. Cui and Weng [28] use both the current estimate of structure and the new frame of image measurements to find the best motion parameters which relate the most recent pair of frames. Subsequently, the structure estimate for the new frame is refined. They use a recursive Kalman filter technique for the dynamic estimation of structure and motion. In every new frame, the current estimate of structure is used to compute the motion parameters for all the feature measurements. The new motion parameters are then used to obtain a combined estimate of structure by fusing the current estimate and the new one. Oliensis and Thomas [67] observe that Cui and Weng ignore the correlation of errors between the 3D estimates of all the scene features, thus weighting certain directions in the structure parameter space incorrectly. These correlations are induced by the errors in the computed motion parameters. They incorporate these correlations into their work, although doing so results in a significantly greater computational cost for updating the structure parameters. However, unlike Cui and Weng, Oliensis and Thomas do not constrain the solution of the motion parameters for every new frame by the current structure estimate. Thus, they treat each motion estimation problem independently. These techniques have been shown to improve the estimates of structure, but only in cases where the inherent ambiguities in the embedded two-frame motion estimation are not severe. The two-frame motion ambiguities can bias the solution for the motion

parameters away from the true solution. Even the covariance-based error modeling of the resulting structure estimates using Kalman filtering is unable to handle such errors.

Spetsakis and Aloimonos [82] formulate a batch estimation problem for both the structure and motion parameters over many frames in contrast with the recursive technique of [28, 67]. Given that m points in the scene are imaged from n different vantage points, they formulate the constraint that for every pair of frames, the corresponding rotation R and translation t should be such that the bundle of rays for each observed point in the scene is coincident in the scene (Figure 2.2 from [82]). All

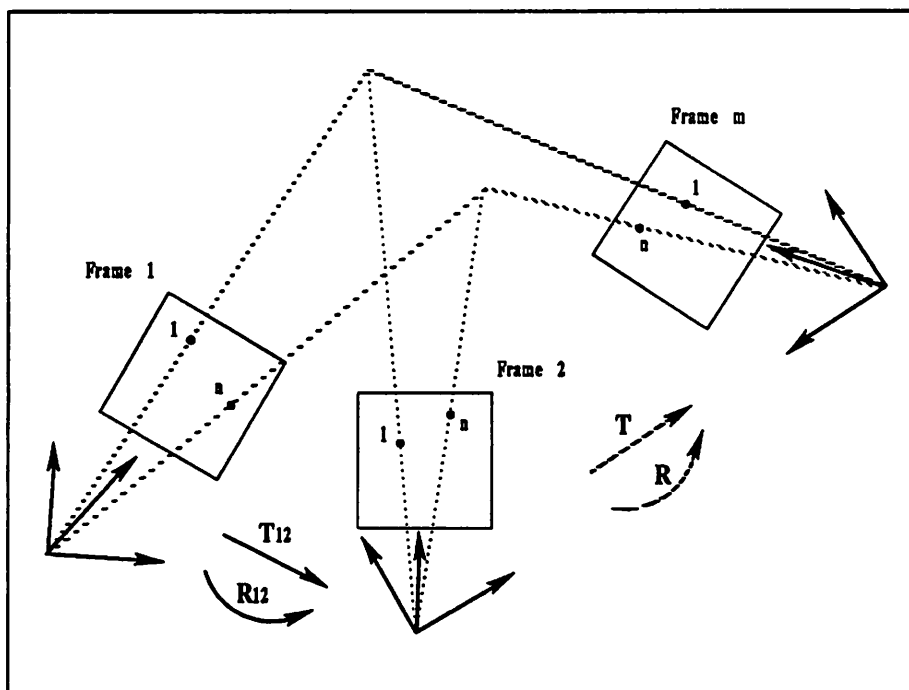


Figure 2.2: The multi-frame bundle of rays.

the pairwise R 's and t 's, with nested constraints amongst triples, can be computed by minimizing an error measure which measures the departure of the measurements

from this constraint. However, the resulting optimization problem is complex and no good techniques are presented for its solution.

In a recent work [89] on multi-frame structure, Tomasi and Kanade also combine the solution for the motion and the structure parameters in a single batch estimation problem. The measured image correspondences over many frames are combined into a matrix whose factorization into a singular value decomposition with appropriate constraints leads to both the 3D shape and the motion parameters. This technique is simple and elegant. All the spatial and temporal measurements are accounted for by a single least squares solution in which the singular value decomposition leads to the optimal motion and structure parameters [31]. However, the work in its current form is limited to orthographic projection and has been applied to real data with large focal length cameras and for objects far way from the camera. The performance under perspective projection with wide variations in the depth of objects in a scene is not known.

There have been many efforts for the incremental estimation of 3D motion using *known* structure or stereo cameras ([61, 99]), or for the refinement of 3D structure with completely known motion ([15, 62]) or a partially known 3D scene ([51]). These have not been reviewed because the interest here is in those approaches which do not assume *specific* models of the scene or knowledge of the motion parameters, or use binocular/multi-ocular vision.

2.1.4 Discussion on Multi-Frame Structure Estimation

The approaches discussed here for multi-frame structure estimation assume no models for 3D structure and motion apart from the rigidity constraint. Structure estimates are obtained by triangulation between successive frames after the computation of the 3D motion parameters. The experimental results presented in the

approaches discussed indicate that to achieve a significant improvement in the estimation of 3D shape from general motion under perspective projection, disparate viewpoints are necessary. It is well known that reconstruction of 3D points using triangulation over two frames introduces errors largely in the depth component for a given error in the image measurements. Furthermore, Oliensis and Thomas have shown that errors in the computed motion parameters also introduce correlated errors across all the 3D points. This suggests that if new viewpoints of the same scene are chosen so that these induce correlated errors in directions which are nearly orthogonal to the directions of existing errors, then the shape estimates might improve. More analytical and experimental work needs to be done to conclusively show this.

2.1.5 Reconstruction Using Models of 3D Motion

Many approaches to 3D reconstruction assume specific models of 3D motion in addition to the rigidity assumption. In general, these approaches assume that many point correspondences over a number of image frames are available. It is also assumed that these points are images of 3D points lying on an object moving with the modeled motion.

The essential idea behind motion model based reconstruction is that image motion of features over extended time frames are expected to conform to projections of an object with unknown structure undergoing the modeled 3D motion. This idea can be applied in situations where either the camera is allowed to move only within the constraints of the modeled motion, or when both the rotational and translational components of the camera motion are smooth enough that the motion can be well approximated by only a few low order terms of its Taylor series expansion. The model of motion is enforced on either a set of intermediate motion parameters derived on a two-frame basis, or directly on the 2D feature correspondences.

Weng et al. [93] compute a series of two-frame transformations (sets of rotations and translations) and then fit a general precessional model of motion which they call the Locally Constant Angular Momentum (LCAM) model. Asada et al. [9] also fit a smooth global model of motion over the pairwise computed parameters. The robustness of these approaches is questionable because the two-frame computations themselves can be highly unstable as discussed earlier.

Shariat [81] uses models of uniform translational and rotational motion with the additional constraint that the frames are sampled uniformly in time. Constraints are separately formulated not only for each type of motion, but also for each type of data set. For each situation, the minimum set of necessary constraints are formulated and the parameters are computed by exact solutions of the resulting non-linear equations.

Broida [18] models the motion of a smoothly moving object as the translation of the origin of an object-centered coordinate system and the rotation of the object in this system. The time-varying position of the object is expressed as a Taylor series whose coefficients represent variations of arbitrary order. Points in 3D are assumed to move with a motion corresponding to derivatives of a reasonable order (typically uniform velocity). Then, the square of the difference between the expected image projections of these points and their measured locations in a number of image frames is formulated as an error measure. This error measure is optimized to find those motion and structure parameters which correspond to a minimum. Both recursive Kalman estimation techniques and batch estimation methods have been used. Superior results are reported with the batch estimation methods.

2.1.6 Discussion of Motion Analysis with Motion Models

The advantage of imposing specific models of motion on the observed motion of smoothly moving objects is that the number of unknown motion parameters does not grow with the number of frames. In contrast, in multi-frame reconstruction

from two-frame motion estimates, every new frame introduces a new set of motion parameters. When using models of motion, a fixed set of parameters is assumed to describe the complete trajectory of a moving object independent of the number of frames observed. Thus, the larger the number of frames considered, the more overconstrained the problem becomes, and hence the effect of random errors in the observations over time can be effectively counteracted.

A second motivation for fitting models of smooth motion to the observed image data is the possibility of describing object motion in its natural frame of reference rather than in some arbitrarily chosen coordinate system. Consider a ball spinning around its own axis. The natural description of such a motion is an axis of rotation passing through the origin of an object-centered coordinate system and the rotation of points around this axis. However, methods based on the computation of motion on a two-frame basis involve first computing a number of rotations and translations in a camera-centered system, and then discovering that a rotation around a fixed axis with no translation is a more succinct description.

In the approaches discussed above, the parameterization of the motion models inherently is a natural parameterization. Thus, if the imaged object undergoes a motion that is describable with a natural set of parameters, then that description is achieved. However, the question of validity of the assumed model in general situations has not been addressed.

2.2 Goal 2 : Object, Boundary and Surface Segmentation

Delineation of regions in the image which correspond to a single rigid motion is necessary before the techniques discussed above for 3D reconstruction can be applied. Segmenting independently moving objects, even without the subsequent goal of their 3D reconstruction, can be useful for tracking and servoing. When only the camera is

moving through a static scene, there is a single rigid motion between the entire scene and the camera. However, even in this situation it is useful to obtain a segmentation of the scene in terms of planar patches and other surface descriptors.

Image motion can also be used for the identification of depth (occluding boundaries) and orientation discontinuities. Thus, interpreting image motion as the projection of the motion of independently moving objects or of surface patches and their boundaries is an important goal.

2.2.1 Multiple Motion Segmentation

Fennema and Thompson [34], in one of the earliest works on motion segmentation, use the brightness constraint equation to cluster image pixels with similar image velocities. Connected components of image pixels whose spatio-temporal intensity gradients correspond to a cluster of x and y velocity components are labeled as coherently moving regions. This method can segment 2D objects translating parallel to the image plane (parallel to the xy -plane).

Adiv [1] segments a displacement field computed over two-frames into regions, each of which corresponds to a rigid motion of a single object/surface patch in the scene. The algorithm is divided into two stages. First, a Hough transform is used to segment the image motion into patches, with each patch describing an affine approximation of a rigid planar motion. In general, this leads to an oversegmentation of the flow field but keeps the computational complexity relatively low. Subsequently, a hypothesize-and-test algorithm is used to group together connected components of these patches into regions, each of which corresponds to a general rigid motion. The constraints used for the second step are those which were presented in Section 2.1.1 on 3D reconstruction.

Both the above techniques are velocity (displacement) based and use only two frames of information.

2.2.2 Boundary Segmentation

Clocksini [25] presented a theoretical work on the labeling of various boundary types (viz. convex, concave, occluding etc.) in flow fields on a spherical imaging surface. Signatures of range profiles across the various types of boundaries were related to the profiles in the spherical flow fields. Thus, in this work 3D boundary types are derived from the image flow.

Boundaries can also be defined as discontinuities in the 2D flow. These possibly correspond to depth and orientation discontinuities or object motion boundaries. Hildreth [40] proposed the use of discontinuities in the normal flow of similarly oriented zero-crossings for labeling image regions as boundaries. Similarly, Thompson et al. [87] use zero-crossing detectors to identify changes in direction and/or magnitude in the vector-valued flow fields. Spoerri and Ullman [83] use statistical tests on the local velocity histograms to detect motion boundaries. In all these approaches, it is assumed that the image flow varies smoothly within patches which are projections of smooth surfaces moving rigidly, and that it changes abruptly across occlusion boundaries or boundaries between independently moving objects.

Black and Anandan [12] use robust estimation techniques to model departures from spatial and temporal continuity of the flow. They compute the optical flow dynamically over a sequence of images and label occluding boundaries in the process. Theirs is the only work in motion boundary detection which uses multiple frames.

2.2.3 Surface Segmentation

The work on surface segmentation using motion is largely limited to planar patch segmentation. Murray and Buxton [64] perform a global optimization based on a maximum a posteriori (MAP) estimate of a planar facet model. Adiv [1] also segments a flow field into patches which correspond to a planar patch flow. This is a step towards general rigid body segmentation in his work.

Faugeras and Lustman [32] use point and line tokens to divide a scene into token sets. Each set represents a planar patch moving over two frames. A seed set of four tokens initiates a planar patch hypothesis which is then extended by a hypothesize-and-test algorithm.

All the above algorithms use two frames of information only. Extensions to dynamic segmentation and estimation over a sequence of images has not been attempted.

2.2.4 Discussion

Delineation of objects from their background, labeling depth and orientation discontinuities and segmenting 3D surfaces from image motion have proven to be hard problems in their generality. In order to pose the figure-ground segregation problem in motion analysis, objects have to be characterized as distinct from their background. Similarity of image motion and that of 3D motion have been used as criteria for delineating multiple motions. However, the use of extended temporal information has not been attempted. This is largely the case with boundary and surface segmentation as well. In other words, use of 3D motion and structure constraints in addressing motion-based segmentation tasks as a dynamic analysis has largely been left unexplored.

2.3 Goal 3 : Feature and Object Tracking

For an autonomous intelligent system, tracking features and objects in a dynamically changing environment is necessary for maintaining identity (correspondence) of objects over time. Maintaining identity is important because many observations of the same entity can be related over time to obtain reliable estimates of its structure and motion, and also to predict its future course. Dynamic identity maintenance with

closely sampled image frames can substantially reduce errors in correspondence while keeping the computational cost of search and matching low.

Sethi and Jain [80] find trajectories of point tokens in an image sequence. Smoothness of image motion over time is used as a heuristic. They formulate a path coherence function which measures the similarity of motion of the point features across two consecutive time instants. This function is embedded in an iterative optimization algorithm which establishes correspondence of m points over n frames by maximizing the path coherence for the whole sequence. Rangarajan and Shah [70] present a non-iterative algorithm for finding point trajectories in images. They minimize a *proximal uniformity function* which measures the deviations from uniformity of velocity, and proximity of position of a set of points over successive time instants.

Rehg and Witkin [71] formulate the 2D tracking problem with a deformation model for the 2D motion and a set of energy-based match criteria for matching the image features. A model of uniform 2D translation provides the temporal prediction.

Papanikolopoulos et al. [68] control a robotic arm to track 2D objects which lie and move in a plane parallel to the image plane. At each time instant, optical flow computed by the method of sum-of-squared-differences (SSD) [7] is used to solve for the 2D motion of an object. This motion serves as an error signal for the controller which corrects for the motion bringing the object back to the desired position in the image plane.

Crowley et al. [27] and Deriche and Faugeras [30] use a locally-constant acceleration model of image motion for tracking individual line segments over a sequence of frames. A dynamic model is maintained for each line segment. The segment's position is predicted and matched dynamically to the newly acquired data for every frame.

Burt et al. [22] suggest a hierarchical framework for tracking and segmentation. An affine transformation is used to describe the motion of planar patches in rigid

motion. Decomposition into multiple motions is done using the mechanism of locking onto a dominant motion and then subtracting out the other components. A frequency domain analysis of the mechanisms underlying this one-component-at-a-time decomposition is presented in [23].

Discussion

All the approaches reviewed above use 2D transformations to compute image motion and subsequently use heuristics for the smoothness of image motion to track image entities over time. There is no attempt to relate tracking constraints to 3D structure and motion within the tracking framework.

2.4 Relationships to Our Work

The primary objective of this work also is the computation of reliable scene structure from dynamic images. The goals of tracking, segmentation and reconstruction are all addressed but with some differences in the problems posed and the methods adopted for their solution.

2.4.1 Structure Reconstruction

First, in contrast with the 3D reconstruction approaches discussed in Section 2.1, we endeavor to capture the definition of an *aggregate* 3D structure defined in terms of primitive features like lines and points. We observe that for tasks like obstacle avoidance for autonomous navigation in man-made environments, it may not be necessary to compute the 3D structure of the scene in terms of the coordinates of most points in the scene. It is also reasonable to assume that scenes in such environments consist of small scale structures, which could be potential obstacles, along with extended structures which fill the background. This work poses the reconstruction problem

as that of direct reconstruction of these aggregate structures and not of individual points/lines. In principle, it is possible to group points and lines into aggregate structures for objects and surfaces and their background. However, as discussed above, the task of reliably deriving 3D coordinates of points and lines has proven to be difficult even when it is assumed that correspondences over time are available and the regions of the image under consideration contain measurements only from a single rigid body motion. In contrast, an advantage in our approach is that it integrates dynamic tracking and robust 3D reconstruction without the need for an explicit computation of the 3D motion parameters.

2.4.2 Segmentation and Tracking

Second, a recurring theme in 3D interpretation from images is that of identifying what an 'object' is, as distinct from its background. Gestalt psychologists tried to capture this notion in dynamic scenes using the notion of *common fate*. The idea of common fate in terms of the similarities of image velocities has been quantified in the segmentation work of Fennema and Thompson [34] and Hildreth [40]. However, similarity of image velocities is inadequate to capture the notion of common fate of objects/surfaces that undergo the same 3D motion. Adiv[2] in his work on segmentation of multiple object motions did characterize regions of images undergoing common 3D motion. However, his work was limited to using two frames of information whereas delineation of such regions and the maintenance of the identity of the corresponding 3D object over time should both be addressed for a practical system.

In defining structure as an aggregate of primitive features, the objective is to define the common fate of an object in terms of a coherent 3D spatial structure which manifests itself as a coherent spatio-temporal image structure under the assumption of smooth motion. The goals are to identify objects which have a common fate in

the above sense, to reconstruct their 3D structure, and to dynamically maintain their identity over time.

In order to achieve these goals, we use general models of 3D structure like *compactness* and *planarity*, and combine these with a general model of smooth motion. The evolution of the spatial model under motion is *dynamically* tracked, leading to a segmentation and reconstruction of the scene in terms of these models. This framework integrates the problems of tracking, segmentation and reconstruction in a framework of spatial and temporal grouping in dynamic images. The ability of a structure to be predicted and tracked consistently under its assumed model becomes a basis for its segmentation and reconstruction as the corresponding 3D structure. Thus, the output of our system is a scene description in terms of segmented compact and/or planar structures at their respective locations with respect to the camera.

In contrast, none of the approaches discussed above for planar patch segmentation uses dynamic multi-frame techniques. The other techniques discussed for tracking use image plane constraints only and hence are useful for 3D segmentation and tracking of objects only after the correspondences have been established.

2.4.3 Motion-model Based Reconstruction

Third, this work provides an alternative approach to 3D reconstruction using a model of rotational motion. In general, it is expected that 3D reconstruction based on models of motion using many image frames will perform better than two-frame motion based methods. This is borne out by the results of Broida [18] and the results discussed in Chapter 4. However, none of the approaches discussed in Section 2.1.5, address the question of validating the assumed model of motion from the observed data. The observed image data is expected to correspond to the assumed model and based on this expectation, the model parameters are derived from the data. This is what we call *direct 3D parameter estimation*. Direct parameter estimation approaches

have to pose complex non-linear optimization problems. These, in general, tend to be sensitive to initial guesses and noise in the data. It is not clear how these approaches will handle gross errors in the data (outliers) or multiple instances of the same motion or instances of multiple models of motion.

In our work on rotational trajectories, an alternative to the direct 3D parameter estimation approach has been adopted. The stress is on both the robust derivation of scene structure and motion and on building a description of image motion which can potentially address the problems discussed above. Spatial and temporal grouping is used to build a description of a discrete set of point correspondences in the image plane in terms of continuous curves or image trajectories. The image trajectories can be used for both a qualitative interpretation of motion for grouping similar motions, and for quantitative estimation of the parameters of the corresponding 3D trajectories.

CHAPTER 3

TRACKING, IDENTIFICATION AND RECONSTRUCTION OF SHALLOW ENVIRONMENTAL STRUCTURE

Detecting obstacles in the path of a mobile robot is an important, and yet generally unsolved, problem in the area of autonomous visual navigation. Autonomous navigation would be greatly benefited if 3D representations of surfaces could be derived using vision. This work deals with 3D reconstruction from monocular vision. Much of the work in recovering scene structure from monocular vision has concentrated on deriving depths of points, lines or pixels but has, unfortunately, achieved only limited success. Both the motion and structure computations suffer from inherent ambiguities [3] in many realistic scenarios and also are very sensitive to noise in correspondences or flow extraction [90]. The recovery of aggregate 3D structures is generally left to some later stage in which features are grouped into surface patches. In this chapter, we demonstrate that useful quantitative inferences about the scene structure can be derived if descriptions are based on generic assumptions about the world over and above rigidity of motion. The motion could be due either to camera motion and/or to object motion.

A major line of research has been to track lines or points over two or more frames, followed by the application of a structure from motion technique to the resulting correspondences [33, 43, 94]. The tracking of image tokens over time has been largely based on heuristics about the motion of these tokens in the image plane. For instance,

locally constant acceleration in time and similarity measures over image tokens have been employed [27, 30]. As the 3D structure of the scene and the motion of the camera are both confounded in the motion of the image tokens, heuristics about image motion can easily break down. The approach described here employs generic assumptions about the 3D motion and structure to compute descriptions of aggregate structures in the imaged scene. This is a significant difference from the tracking algorithms developed by Crowley et al. [27] and Deriche and Faugeras [30] who employ only partially valid heuristics involving 2D motion. In our work, generic model-matching, common fate grouping and prediction-based tracking are integrated in a single dynamic framework for describing scene structure over time.

Furthermore, an advantage of the approach described here is that 3D structure information is derived reliably without the intermediate step of explicit computation of the 3D motion parameters. Recall that the well-known inherent ambiguities ([3, 94], Chapter 2) in the process of decomposing the image motion into a 3D rotation and a translation can lead to large errors in the 3D structure estimation.

The goal here is to discover aggregate structures in the imaged scene which can be characterized as *shallow structures*. Shallow structures are 3D structures with the property that the difference in depth within the whole structure is small compared to its distance from the camera. Figure 3.1 shows an image of a hallway. This scene consists of compact structures like the cones and the trash can, and extended structures like the walls, the floor and the ceiling. When viewed from distances at which it might be desirable for a mobile robot to represent these internally, the variation in depth within these structures is small compared to their average distances from the camera. That is, the structures can be characterized as shallow at distances where the path planner for the robot might need an internal representation of the structures.



Figure 3.1: A Hallway scene with shallow and non-shallow structures.

In this work, constraints derived from the shallowness property are employed to identify shallow structures among the larger scale structures in the background. A general formulation is developed and a dynamic algorithm is presented which works over a sequence of images captured by a camera undergoing smooth motion. Hypothesized shallow structures are dynamically tracked under the shallowness assumption. Within a temporal window of a few frames, true shallow structures are extracted from the set of hypothesized aggregate structures on the basis of both the consistency of predictions in tracking and the depth of the structure. In other words, temporal evolution of a hypothesized structure is used to verify its consistency within the constraints of spatial shallowness.

Shallow structures are shown to be *affine describable* over time. Instead of clustering image features into shallow structures on the basis of this property applied over only two frames, the idea of *affine trackability* is applied dynamically to each hypothesized shallow structure. The key idea in this work is that affine trackability can be used to segment shallow structures in a scene and to reconstruct these in 3D. Two important insights have been developed within an estimation theoretic framework for the problem of robust shallow structure tracking. First, it is observed that matching of an aggregate structure as a whole is generally unambiguous in comparison with independent matching of features within the structure. Representation of a structure as a state vector along with the associated covariance matrix that allows for uncertainties in modeling and measurements, provides a natural representation for the aggregate structure as a whole that is suitable for model matching. Second, in order to circumvent the high dimensionality of this representation in matching, a nice decoupling of the structure parameters is shown to lead to a matching problem of less complexity.

The 3D location and the dynamics of the entire aggregate structure are directly represented instead of the depth of more primitive tokens like points and individual lines. The derived description of the scene can be viewed as a set of fronto-parallel planes (*cardboard cut-out* surfaces) of constant depth, one for each shallow object in the scene.

3.1 Overview

This presentation is broadly divided into three parts.

1. A Formulation of Affine Describability and Trackability of Shallow Structures.

2. An Algorithm for Shallow Structure delineation.

3. Experimental Results.

3.1.1 Affine Describability and Trackability

First, it is shown that if the rotations of the camera between two image frames are small, then the image projections of a shallow structure can be approximated by a four-parameter affine transformation — scale change, rotation around the optical axis and translation in the plane [86]. This is called *affine describability*. A model of errors for 2D line tokens is developed which is used to compute the affine parameters and their covariances. These computed affine parameters with their associated covariances represent the 2D dynamic parameters of a hypothesized shallow structure.

Second, affine motion of a shallow structure in the image plane is used to maintain its identity over time by tracking the structure dynamically. This is called *affine trackability*. Tracking involves both predicting the structure's appearance at every instant and finding its match in the newly acquired image data. Matching the prediction to the data for the structure as a *whole* is shown to be resilient to errors in prediction and modeling. These errors are a result of approximations in modeling the motion and structure, and measurement errors in the image data.

3.1.2 Tracking Algorithm

The goal is to declare a hypothesized structure as shallow or non-shallow on the basis of its trackability as an affine structure. Tracking is done within a framework of Kalman filtering and model matching. The affine parameters are used in a dynamic model of the image motion of a shallow 3D structure. Matching is done using the predictions from this model and the actual data. Newly matched data is then used to refine the estimates of the model. The tracking algorithm consists of two phases.

In an initial *bootstrap phase*, each line in a newly hypothesized structure in frame 1 is matched to a frame 2 line using the method described in [96]. Using this two frame correspondence, two state vectors are instantiated for the hypothesized structure — one is a vector of the location and the second of the affine motion parameters. The location vector specifies the image location of the structure in terms of its constituent lines and the current estimated depth.

The *tracking phase* consists of prediction, matching and updating. Once a model of the hypothesized structure has been instantiated, its affine motion state at frame t is used to predict the motion between time t and $t+1$. The predicted motion is used to predict the location at $t+1$. Using this predicted location, a search is performed in frame $t+1$ to obtain candidate matches for the structure as a whole. For each candidate match, its *Mahalanobis distance* [30, 58] is computed with respect to the *predicted model as a whole* and the best match above a threshold is selected. This match is then used to update its state.

The prediction of motion parameters is based on an assumption of *approximately uniform* motion between successive frames. To model departures from uniformity, modeling error is added to the predicted motion parameters with the result that the covariances of the predicted model effectively encode perturbations around the predicted parameters.

The algorithm incorporates the notion of model persistence over time [27]. Occlusions, deocclusions, overgrouping and undergrouping of lines are very frequent occurrences in a sequence of images. To handle these effectively, the model remains instantiated even when an adequate match is not found for a few frames.

3.1.3 Experimental Results

Results of shallow-structure tracking and delineation are presented on image sequences which were captured with a camera mounted on a Denning robot vehicle. The

sequences illustrate the power of the incremental tracking approach. For instance, the power of aggregate model-matching as opposed to individual line matching is demonstrated. In addition, it is shown that the algorithm can be applied to independently moving objects as well, if they satisfy the shallow structure constraint. This follows from the nature of the assumptions and models. A scene is modeled as a collection of locally shallow structures which potentially move independently with a rigid motion relative to the camera. The motion could be due either to camera motion and/or to object motion.

3.2 Relationship to Previous Work

The approach developed in this work has been inspired by the work of Crowley et al. [27] and Deriche and Faugeras [30] on multi-frame tracking of line segments in images, and by the structure from looming idea of Williams and Hanson [96], but goes beyond the framework developed by either of them. In [27] and [30], a locally-constant acceleration model is used for tracking of individual 2D line segments over a sequence of frames. Each model represents the location and dynamics of a single line segment and is kept current using a prediction and matching technique within the Kalman filtering framework. This model can lead to tracking errors even for simple cases of motion, such as a uniform translation in depth, especially in images where more than one line of similar orientation appears proximally in the image plane¹. In contrast, the model defined here assumes both a property of 3D structure (shallowness) and smooth motion of the camera. The affine motion of a shallow structure provides a more exact trackability constraint. A shallow structure, being a collection of primitive tokens (lines and points), provides implicit figural context for more

¹A situation which is not uncommon in buildings and hallways.

robust matching than just the primitive tokens. In addition, tracking in our work is used for segmentation and 3D reconstruction.

Williams and Hanson [96], in their work on flow-predicted line correspondences, have demonstrated that for translations in depth, reliable depth can be computed by measuring the temporal magnification (looming) of lengths and regions at approximately constant depth. Their method was demonstrated on manually selected virtual line segments and regions in the image. Automatic segmentation and temporal-persistence in tracking was not addressed. For instance, their system had limited ability to recover from undergrouping/overgrouping errors, and no ability to handle occlusions. Further, it did not assume or utilize motion continuity over time for tracking. Since it was based directly on the computed image displacements, it handles fairly arbitrary kinds of motions if the displacement fields are sound. The work presented here differs in that the notion of shallowness of depth of a structure has been formalized into a constraint which is utilized to automatically identify shallow structures in the scene. For motion with a significant component in depth, reliable depth of the structure can be computed from its scale parameter, which is related to looming and divergence.

A different approach for representing the scene as image regions corresponding to surfaces at different depths has been developed by Nelson and Aloimonos [66]. The divergence of the flow field between a pair of frames is used to divide various regions in the image into surfaces at different depths with respect to the camera. Reliable computation of flow and its divergence requires textured surfaces. In many real-world navigation scenarios, like a robot moving down a hallway, most surfaces are smooth and featureless with only a few reflectance edges. In such situations, occluding contours and significant reflectance edges are a reliable source of geometric cues. Our work uses the temporal evolution of such geometric image tokens. Furthermore,

figural cues can be very naturally integrated within the framework of tracking of aggregate structures consisting of line and/or point tokens.

One of the earliest attempts at describing the scene as planar patches and its subsequent segmentation into multiple object motions was that of Adiv [1]. His approach employed the constraints on image flow from the rigid motion of a planar patch to group image regions, each region corresponding to such a motion. The input used was sparse or dense image flow and the associated confidence measures between a pair of images [8]. Again, since the method is based on image flow, it is not very reliable when the scene is composed primarily of textureless surfaces. Furthermore, Adiv's approach was limited to descriptions based on only two image frames and extensions to multiple frames has not been proposed.

Faugeras and Lustman [32] also suggest an approach for reconstructing the scene as planar patches based on line tokens. The relationship between a pair of image projections of a set of lines on a plane is derived as a projective transformation involving the plane and motion parameters. However, no clear algorithm is given for using this constraint to obtain the desired segmentation.

3.3 Shallowness as Affine Describability

In this and the next two sections, we show how projections of shallow structures are affine transformable over time, and present the solution for their affine parameters. Furthermore, a match measure is developed for matching predictions against the data while accounting for measurement and prediction errors.

Given a set of 3D points whose extent in depth δZ about a nominal point Z_0 is small compared to Z_0 and assuming that the rotations between two image frames are small, then the transformation of the projections of the point sets between the

two time instants can be accurately approximated by a four-parameter affine transformation. Subscript i for the i th point is dropped in the derivation for notational convenience. A camera-centered coordinate system is chosen in which the XY -axes are in the image plane and the Z -axis points into the scene along the optical axis of the camera. The origin is the center of projection and lies on the optical axis with the image plane a focal length away from it along the positive Z -axis. The following notation is adopted:

\mathbf{P}, \mathbf{p} : 3D vector $[X, Y, Z]$ of an imaged point at t and the corresponding 2D image vector (x, y) .

\mathbf{P}', \mathbf{p}' : The 3D vector $[X', Y', Z']$ at $t + 1$ and the corresponding 2D image vector $[x', y']$.

Z_0, Z'_0 : The depths of the 3D centroids of the point set at t and $t + 1$.

δZ : Extent in depth around the centroid.

s : Scale defined as Z_0/Z'_0 .

R : The small angle approximation to the 3×3 rotation matrix formed out of $[\omega_x, \omega_y, \omega_z]$.

R_z : The 2×2 rotation matrix for rotations around the z - axis.

\mathbf{T} : The 3D translation vector $[T_x, T_y, T_z]$.

Ω, \mathbf{T}_{2D} : The 2D vectors $[\omega_y, -\omega_x]$ and $[T_x, T_y]$, respectively.

f : The effective focal length of the camera given a square image.

The weak perspective projection equation for a shallow structure, approximated to the first order, can be written as,

$$\frac{1}{f}\mathbf{p} \approx \frac{1}{Z_0}\left(1 - \frac{\delta Z}{Z_0}\right)\mathbf{P} \quad (3.1)$$

The rigid body transformation between the two 3D vectors is:

$$\mathbf{P}' = R\mathbf{P} + \mathbf{T} \quad (3.2)$$

Using these two equations, the relationship between the projections at the two instants can be written as:

$$\frac{1}{f}\mathbf{P}' = \frac{Z_0}{Z'_0}\left(1 - \frac{\delta Z}{Z'_0} + \frac{\delta Z}{Z_0}\right)\frac{1}{f}R_z\mathbf{P} + \frac{Z_0}{Z'_0}\left(1 - \frac{\delta Z}{Z'_0} + \frac{\delta Z}{Z_0}\right)\Omega + \left(1 - \frac{\delta Z}{Z'_0}\right)\frac{1}{Z'_0}\mathbf{T}_{2D} \quad (3.3)$$

Since our assumption is that the rotations, field-of-view and $\frac{1}{Z'_0}[T_x, T_y]^T$ are small, the second and higher order terms can be ignored and this transformation can be approximated as follows:

$$\frac{1}{f}\mathbf{P}' \approx \frac{1}{f}sR_z\mathbf{P} + \mathbf{t}, \quad \mathbf{t} = s\Omega + \frac{1}{Z'_0}\mathbf{T}_{2D} \quad (3.4)$$

which is a four-parameter affine transformation (also called a *similarity transformation*). We emphasize that these assumptions are easily satisfied in most visual motion scenarios using commonly available CCD cameras. For instance, rotations up to 0.1 radians (about 5 degrees), FOVs of up to 25 degrees (maximum $\frac{X}{f}$ of about 0.2) and translations in the X and Y directions of up to 1 unit for objects as close as 10 units, satisfy these assumptions. Similarly, structures possessing a $\frac{\delta Z}{Z_0}$ ratio of 0.1 or less can be reasonably characterized shallow and therefore affine describable over time.

3.4 Does Affine Describability Imply Shallowness ?

The above formulation shows that if, for a structure in 3D, a fronto-parallel plane (parallel to the image plane) is a good approximation, and if the motion between two image frames is small, then its motion in the image plane can be approximated by a four-parameter affine transformation. The question in the context of 3D reconstruction is whether this transformation is a sufficient condition too, that is, whether affine transformable patterns in the image plane correspond to shallow structures.

For the four-parameter transformation derived above, the answer to the question of existence and uniqueness is straightforward. There is always a unique fronto-parallel plane at a distance given by the scale parameter whose projections are the image patterns. However, there is no unique rigid motion which can be derived because the 2D translation parameters are a combination of the 3D rotation and translation parameters.

In addition to the issue of uniqueness, we have to say how well the reconstructed structure corresponds to some real structure. Unfortunately, there is one configuration of points for which the reconstruction can have an arbitrarily large error. For a purely translational motion, consider the point at the focus of expansion/contraction (FOE/FOC) and any other set of points which are projections of 3D points lying at an arbitrary constant depth. For this total configuration of points, the transformation is affine even though there may be an arbitrarily large difference between the depth of the point at the FOE/FOC and the remaining points. However, this is a degenerate case and can generally be avoided.

3.4.1 General 2D Affine Transformation

In the above formulation, we have chosen to approximate a 3D shallow structure by a fronto-parallel plane. What is the resulting description if a plane of arbitrary orientation is chosen to approximate a shallow structure? It can be shown that the four-parameter 2D affine description generalizes to a six-parameter 2D transformation for the approximation with an arbitrarily oriented plane. It is well known that the object-plane-to-image-plane transformation for a planar object under weak perspective projection is a six-parameter 2D affine transformation [52]. In other words, if a shallow structure is approximated by an arbitrary plane and not by a fronto-parallel plane, then the transformation from the object plane coordinate system to the image coordinate system is a general 2D affine transformation. Further, under rigid

motion, projections of this structure over two time instants are also related through an affine transformation. Thus, projections of planar approximations of arbitrary shallow structures can be related through a general 2D affine transformation.

Given a general 2D affine transformation, what can be said about the corresponding 3D motion and planar parameters? In the context of object recognition, Huttenlocher [44] has shown that given a 2D affine transformation between a model plane and the image plane, a 3D similarity transformation (up to a reflection) that relates the model plane and its image can be recovered. In other words, the relative orientation (up to a reflection), translation (parallel to the image plane) and distance along the optical axis (inverse scale) of the model plane with respect to the image plane can be recovered. However, this result is not directly useful for the case of motion where the model plane is not available and the affine transformation relates two image projections of an unknown plane. For shape from textures, Kanade and Kender [48] showed that given the affine transformation between the image projections of two planar patches, the relative orientation can be recovered only if the absolute orientation of one of the patches in the camera coordinate system is known. The scale can be recovered only if the slant of one patch is known or if the slants for both the patches are equal. Extending this to the case of motion, it is evident that the relative orientation and scale cannot be recovered in general from the six-parameter affine transformation.

3.5 Solving for Affine Parameters and Their Covariances

Given a set of line correspondences in two frames, we wish to compute their affine motion parameters. Although the following derivation is for lines, it is easy to generalize it for a combined set of lines and points. The error measure is general enough to support a range of image measurement models — from strict line segments

with absolutely reliable endpoints (equivalent to point tokens) to lines with infinite extent (absolute uncertainty in the longitudinal location of endpoints). As shown in Figure 3.2, the error measure is a sum of the parallel and perpendicular components

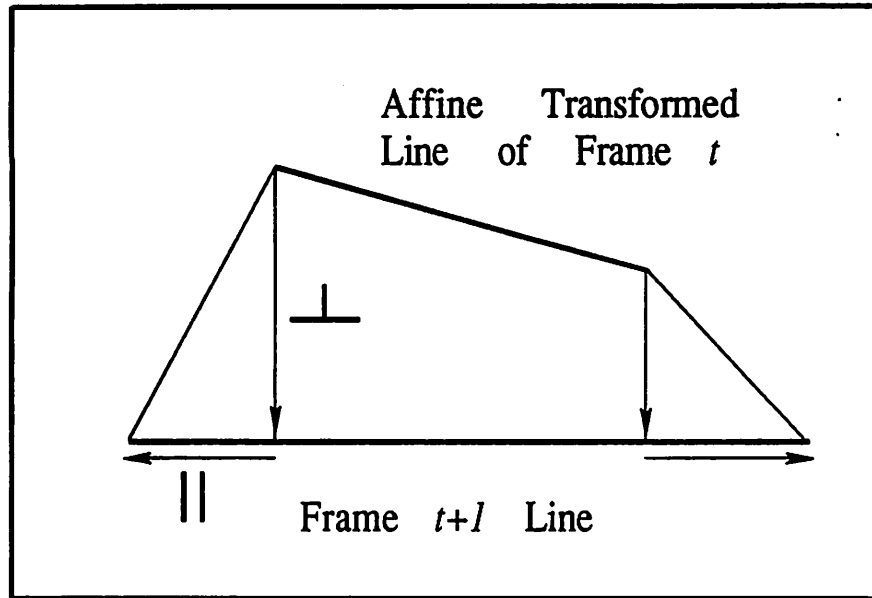


Figure 3.2: The parallel and perpendicular error components.

of the vectors joining the corresponding endpoints of the line in frame $t + 1$ and the affine transformed line in frame t [4]. The parallel and perpendicular directions are defined with respect to the line in frame $t + 1$.

Equation 3.4 can be rewritten in pixel coordinates as follows:

$$\mathbf{p}' = D\mathbf{r}_s + \mathbf{t} \quad (3.5)$$

where the matrix $D = \begin{bmatrix} x & -y \\ y & x \end{bmatrix}$ is the data matrix which is constructed using the point $\mathbf{p} = [x \ y]^T$ in frame t . Vector $\mathbf{r}_s = [s \cos \omega_z \ s \sin \omega_z]^T$ is the product of scale s and rotation, ω_z , around the optical axis. With this simplification, the error

measure, for a pair of corresponding lines i , is,

$$E_i = \sum_{j=1}^2 w_{\perp i} [(D_{ij} \mathbf{r}_s + \mathbf{t} - \mathbf{p}'_{ij}) \cdot \mathbf{n}'_i]^2 + w_{\parallel i} [(D_{ij} \mathbf{r}_s + \mathbf{t} - \mathbf{p}'_{ij}) \cdot \mathbf{l}'_i]^2 \quad (3.6)$$

Here j refers to endpoint 1 or 2, $w_{\perp i}$ and $w_{\parallel i}$ are the weights for the perpendicular and parallel error components, respectively, and \mathbf{n}'_i and \mathbf{l}'_i are the unit normal and direction, respectively, of the line in frame $t + 1$. It is clear from Figure 3.2 that the first term in the above equation is the weighted perpendicular distance between the affine transformed endpoint of a line at t to the corresponding line in the next frame. The second term is the weighted longitudinal distance. The weights associated with each of the error components can be chosen appropriately for both points and lines extracted from the image data. In order to model a circular uncertainty region associated with an extracted point token, $w_{\perp i}$ can be set equal to $w_{\parallel i}$. If $w_{\parallel i}$ is set to 0, then the error measure captures the error model for lines represented as infinite lines. Similarly, measurement errors for line segments can be represented by appropriate choices of the two weights. For example, for lines typically $w_{\perp i}$ is much larger than $w_{\parallel i}$, reflecting the known noise characteristics of most line extraction algorithms. In general, the weights can be suitably chosen depending on the type of token used and the associated noise model of the extraction process.

For a set of token correspondences, the unknown parameters \mathbf{r}_s and \mathbf{t} can be found by minimizing $\sum_i E_i$. Through a series of simple algebraic manipulations it can be shown that the following linear system gives the solution:

$$\begin{bmatrix} M_{24} & M_{13} \\ M_{13}^T & M_{56} \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{r}_s \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{12} \\ \mathbf{v}_{34} \end{bmatrix} \quad (3.7)$$

where²

²In the following we drop the subscript j for the endpoints and assume the error term for each line includes both endpoints.

$$\begin{aligned}
M_{13} &= M_1 + M_3, & M_1 &= \sum_i w_{\perp i} \mathbf{n}'_i \mathbf{n}_i^T D_i, & M_3 &= \sum_i w_{\parallel i} \mathbf{l}'_i \mathbf{l}_i^T D_i; \\
M_{24} &= M_2 + M_4, & M_2 &= \sum_i w_{\perp i} \mathbf{n}'_i \mathbf{n}_i^T, & M_4 &= \sum_i w_{\parallel i} \mathbf{l}'_i \mathbf{l}_i^T; \\
M_{56} &= M_5 + M_6, & M_5 &= \sum_i w_{\perp i} D_i^T \mathbf{n}'_i \mathbf{n}_i^T D_i, & M_6 &= \sum_i w_{\parallel i} D_i^T \mathbf{l}'_i \mathbf{l}_i^T D_i; \\
\mathbf{v}_{12} &= \mathbf{v}_1 + \mathbf{v}_2, & \mathbf{v}_1 &= \sum_i w_{\perp i} \mathbf{n}'_i \mathbf{n}_i^T \mathbf{p}'_i, & \mathbf{v}_2 &= \sum_i w_{\parallel i} \mathbf{l}'_i \mathbf{l}_i^T \mathbf{p}'_i; \\
\mathbf{v}_{34} &= \mathbf{v}_3 + \mathbf{v}_4, & \mathbf{v}_3 &= \sum_i w_{\perp i} D_i^T \mathbf{n}'_i \mathbf{n}_i^T \mathbf{p}'_i, & \mathbf{v}_4 &= \sum_i w_{\parallel i} D_i^T \mathbf{l}'_i \mathbf{l}_i^T \mathbf{p}'_i.
\end{aligned}$$

The vector \mathbf{r} , computed from Equation 3.7 can be further decomposed into the two parameters s and ω_z .

3.5.1 Modeling Uncertainties in Image Lines

Lines extracted in images are more reliable in their lateral than in their longitudinal locations. The unreliability of their endpoints is, in general, due to overgrouping/undergrouping, occlusions/deocclusions and corner effects of the intensity surface. The uncertainties in the endpoints of a line can be modeled as variances, σ_{\parallel}^2 and σ_{\perp}^2 which are the parallel and perpendicular uncertainties respectively in a coordinate system aligned with the line as shown in Figure 3.3. If the orientation of the line in the image coordinate system xy is θ , then the corresponding uncertainties in any endpoint can be expressed as [30]:

$$\Lambda_{xy} = \begin{bmatrix} \sigma_{\parallel}^2 \cos^2 \theta + \sigma_{\perp}^2 \sin^2 \theta & (\sigma_{\parallel}^2 - \sigma_{\perp}^2) \cos \theta \sin \theta \\ (\sigma_{\parallel}^2 - \sigma_{\perp}^2) \cos \theta \sin \theta & \sigma_{\parallel}^2 \sin^2 \theta + \sigma_{\perp}^2 \cos^2 \theta \end{bmatrix} \quad (3.8)$$

3.5.2 Covariances of the Affine Parameters

If $w_{\perp i}$ and $w_{\parallel i}$ are chosen to be the reciprocal of the perpendicular and parallel variances (σ_{\perp}^2 and σ_{\parallel}^2), then Equation 3.6 represents a standard weighted linear least squares problem. Its solution, given in Equation 3.7, can be written concisely as $M_{tot} \mathbf{v}_{aff} = \mathbf{v}_{tot}$, where the new symbols have the obvious correspondence with their

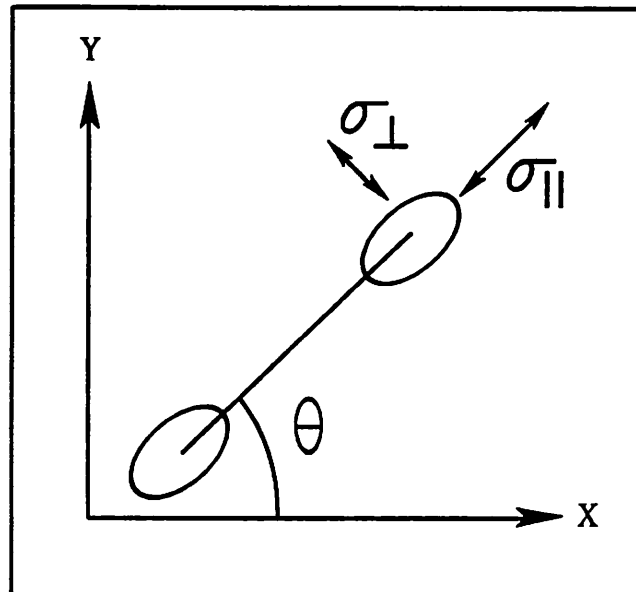


Figure 3.3: The model for noise in lines. Parallel and perpendicular endpoint uncertainties.

expansions in the equation. Using a standard result for the covariances of the output parameters of a least squares problem [85], the covariances of the affine parameters can be written as

$$\Lambda_{\mathbf{r}, \mathbf{t}} = M_{tot}^{-1} \quad (3.9)$$

where $\Lambda_{\mathbf{r}, \mathbf{t}}$ is the 4×4 covariance matrix of the affine parameters \mathbf{r} , and \mathbf{t} .

This completes the discussion of the estimation of affine parameters and their covariances, given correspondences between noisy measurements of a set of line pairs in two frames. In the next section, a representation for aggregate structures is developed and a match measure for comparing two such structures is derived.

3.6 Aggregate Structure Representation and Matching

It is emphasized that an aggregate structure refers here to any set of lines (and/or points), shallow or non-shallow, and hence is called a hypothesized structure. The constituent lines of such a structure are used in two distinct ways by the algorithm — as infinite lines for motion computation and as line segments for prediction and matching. Each use imposes its own requirements on the representation of a line and consequently, on the representation of a hypothesized aggregate structure.

Given a set of correspondences obtained by matching the prediction of an aggregate structure with its appearance in a newly acquired frame, the affine motion parameters are solved for by treating the lines as infinite lines, that is, $w_{\perp i} = 1$ and $w_{\parallel i} = 0$ in Equations 3.6 and 3.7. This leads to the most accurate affine parameters possible for a set of lines even when lines break or grow, or become partially occluded, since only the transverse position of the lines needs to be accurate.

For prediction and matching, this is not sufficient, especially when a model includes only a small number of lines. In particular, it can be shown that for small line sets, the longitudinal image location of the affine projected lines in frame $t + 1$ can be quite erroneous with respect to the data lines even when the residual error for the affine solution based on the perpendicular error is small (Section 3.8). Thus, if the idea of shallowness is to be fully exploited for matching, then lines with infinite extent cannot be used. Moreover, although the extent and location of a line is relatively unreliable, this information still imposes a strong constraint on its motion when the uncertainties are modeled correctly.

Thus, we employ both requirements in different phases of the algorithm to achieve a representation appropriate for both aggregate structures and their constituent lines. It is easily shown that if lines are treated as infinite when solving for the affine parameters, then a minimum of three lines not intersecting at a single point overconstrain

the solution. In fact, any set of parallel lines or any set of lines all intersecting in a single point lead to an infinite number of solutions. Consequently, a primitive aggregate structure is defined as a set of three (or four) lines. The degenerate configurations are automatically detected when solving for the affine parameters. We emphasize here that the algorithm and its implementation are not restricted to sets of only three lines. However, this is the minimum number sufficient for simple shallow structure segmentation while keeping the complexity of matching to a minimum.

3.6.1 Representing Lines and Aggregate Structures

Each line is represented as a finite line segment with the four-tuple

$$l_s = [x_m \ y_m \ \theta \ l]^T$$

where (x_m, y_m) is its midpoint, θ the orientation and l its length. Given the model of perpendicular and parallel uncertainties, the covariance matrix for the model of a segment is:

$$\Lambda_{l_s} = \begin{bmatrix} \frac{1}{2}\Lambda_{xy} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2\sigma_{\perp}^2/l^2 & 0 \\ 0 & 0 & 0 & 2\sigma_{\parallel}^2 \end{bmatrix} \quad (3.10)$$

where Λ_{xy} is the 2×2 endpoint covariance matrix of Equation (3.8). It was shown by Deriche and Faugeras [30] that in this representation the covariance matrix for a given line is independent of its position in the image plane. Also, the midpoint uncertainties are uncorrelated with the orientation and length; this will be used to advantage in the matching.

Each aggregate structure of three lines is represented as a hypothesized 3D structure in two parts. Its image projection is a 12×1 vector,

$$M_{Iloc} = [x_{m1} \ y_{m1} \ \dots \ x_{m3} \ y_{m3} \ \theta_1 \ l_1 \ \dots \ \theta_3 \ l_3]^T = [M_m^T \ M_{\theta l}^T]^T \quad (3.11)$$

composed of the three line segments. All the midpoints have been concatenated. Its 3D location is represented as its currently estimated depth \hat{Z} . Note that this 13×1 representation (image location and depth) completely defines an aggregate shallow structure in 3D; it is called the *location state* of the structure in the following.

The dynamics of the structure are represented by the current four affine motion parameters (Equation 3.7), their covariance matrix, the total residual error (Equation 3.6) and a *projection error*. Recall that the residual error measures only the error in the transverse positions between lines in frame $t + 1$ and the affine transformed lines in frame t . In addition, to measure the total image location error for the affine projected aggregate structure, a projection error is defined which measures the sum of the *Mahalanobis distances* [30, 58] between the affine projections of *line segments* in frame t and the corresponding lines in frame $t + 1$. That is,

$$merr_{proj} = \sum_{i=1}^3 (l'_{s_i} - l_{s_{aff-proj-i}})^T (\Lambda_{l'_{s_i}} + \Lambda_{l_{s_{aff-proj-i}}})^{-1} (l'_{s_i} - l_{s_{aff-proj-i}}) \quad (3.12)$$

where l'_{s_i} is the vector for the i th line segment in frame $t + 1$, and $l_{s_{aff-proj-i}}$ is the vector for the affine projected corresponding line of frame t . The Mahalanobis distance between two state vectors is the covariance normalized Euclidean distance between them.

These location and motion state vectors completely describe the current location and the current affine motion parameters of a given structure along with their associated covariances.

3.6.2 Model Matching with Measurement and Prediction Errors

For the purposes of the development in this section, it is assumed that the predicted affine parameters and their covariances for a hypothesized aggregate structure at time $t + 1$ are available from its past history. The specific prediction model used is discussed in Section 3.7.

Sources of Error

The process of matching the predicted model structure with potential aggregate matches must account for three sources of error:

1. Measurement uncertainty in the image data on which the prediction is based.
2. Departures from modeled predictions of motion, (e.g. non-uniform motion).
3. Error in affine description due to departures from a fronto-parallel plane for the real shallow structure.

First, each of these sources of error is discussed independently from the point of view of how they affect the location and affine motion models of an aggregate structure. In the next section, it is then shown how all these sources of uncertainty are incorporated into a unified error model through the covariances of the predicted model.

It is possible to account for measurement uncertainties by propagating the covariances of a line in the previous frame and those of the predicted affine parameters into the covariances of the predicted line. The problem with this approach is that if each line is matched individually to its potential match set, in effect each line is allowed a perturbation within the limits of its variance independent of the other lines in the model. This is not desirable since beyond the individual line measurement uncertainties, model matching should incorporate the perturbation of the model as a whole when searching for the best match.

In order to model deviations from uniformity of motion in the prediction process for the motion of aggregate structures, we now analyze the typical imaging scenario for possible non-uniformities. Assume that the camera is mounted on a mobile platform³ and the sequence of frames is sampled uniformly in time under smooth motion. The

³Like the Denning vehicle used in all our experiments.

most significant source of error in this scenario is excessive rotation around either of the three axes. These errors typically occur due to two major causes — (i) the rotations induced due to non-uniformity of torque on the wheels or differential slipping and (ii) the small differential slants and tilts (small bumps, shallow depressions and ramps) on an otherwise planar ground plane. Out of the three rotations, those in depth (ω_x and ω_y) are the dominant source of error in prediction for small FOV cameras [1]. Consequently, errors in rotation in depth are modeled as uncertainties in certain of the affine motion parameters. In addition, it is to be emphasized that uncertainties due to errors in the other motion parameters also can be handled within the framework of dynamic prediction and matching with uncertainties.

It was shown in Equation 3.4 that the 3D rotations ω_x and ω_y lead to translations in the image plane under the shallowness constraint. The non-uniformity of motion is primarily accounted for by adding a diagonal covariance 2×2 matrix, $\Lambda_{t_{err}}$, to the already computed covariance, $\Lambda_{t_{pred}}$, for the predicted affine translation vector. This is equivalent to adding plant noise to the dynamic model in a Kalman-filter [37]. Similarly, uncertainties in the prediction of the other two parameters (s and ω_z) due to motion uncertainties can be modeled by increasing their measurement covariances. An advantage of handling non-uniformity in this way is that it provides a principled method for model matching while allowing for modeling uncertainties.

The third source of error, approximation of a structure by a fronto-parallel plane, can also be modeled by allowing for uncertainties in the predicted parameters. In this case, however, the constituent lines in the structure will be affected independently and not as a whole model. Each line can be the projection of a real 3D line that lies in front or behind the reconstructed plane with equal probability. The parameters of a predicted line that are most likely to be affected by this are the scale and the orientation.

A Model Match Measure

If we consider the complete specification of the model as the 12×1 image location vector of Equation 3.11 and match this model as a whole using the Mahalanobis distance as the match measure, it requires the inversion of a 12×12 covariance matrix for every match to be checked. This is not very practical. However, from the discussion in the previous section, the major uncertainty is expected to be in the prediction of the translation parameters. The translation parameters affect only the location of the midpoints for each of the lines and not their orientation or lengths. Thus, the 12×1 vector was separated in Equation 3.11 into a 6×1 sub-vector of midpoints and a 6×1 sub-vector of orientations and lengths.

Now we show that computing the propagated covariances of the 6×1 vector of midpoints achieves resilience to errors in prediction, due to the non-uniformity of motion, as expected. Assuming that at time instant t , $\mathbf{r}_{s_{pred}}$ and \mathbf{t}_{pred} are the affine parameters for the predicted motion between t and $t + 1$, the predicted vector of midpoints can be written in terms of $\mathbf{r}_{s_{pred}}$ and \mathbf{t}_{pred} , and the data lines in frame t (using Equation 3.7) as follows:

$$\mathbf{M}'_m = M_D \mathbf{r}_{s_{pred}} + I_D \mathbf{t}_{pred}, \quad (3.13)$$

$$\mathbf{M}'_m = [x'_{m1} \ y'_{m1} \ \dots \ x'_{m3} \ y'_{m3}]^T$$

$$D_i = \begin{bmatrix} x_{mi} & -y_{mi} \\ y_{mi} & x_{mi} \end{bmatrix} \quad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$M_D = [D_{m1}^T \ D_{m2}^T \ D_{m3}^T]^T \quad I_D = [I_2 \ I_2 \ I_2]^T$$

Using the above equations, it is easy to derive the covariance matrix of the 6×1 vector M'_m .

$$\Lambda_{M'_m} = R_{sbig} \Lambda_D R_{sbig}^T + M_D \Lambda_{\tau_{s_{pred}}} M_D^T + I_D \Lambda_{t_{pred}} I_D^T \quad (3.14)$$

where $R_s = \text{Rotation_Matrix}[\omega_{z_{pred}}]$ is the 2×2 rotation matrix for angle $\omega_{z_{pred}}$, $R_{sbig} = \text{diag}[R_s, R_s, R_s]$ (a matrix with the 2×2 rotation matrices for $\omega_{z_{pred}}$ along its diagonals and zeros elsewhere), $\Lambda_D = \text{diag}[\Lambda_{m1}, \Lambda_{m2}, \Lambda_{m3}]$ where Λ_{mi} is the 2×2 midpoint covariance matrix for the i th model line of the previous frame, and $\Lambda_{\tau_{s_{pred}}}$ and $\Lambda_{t_{pred}}$ are the 2×2 covariances of the predicted affine parameters. $\Lambda_{\tau_{s_{pred}}}$ and $\Lambda_{t_{pred}}$ have been considered independent for convenience. A similar form could easily be derived under the assumption that they are correlated.

In order to provide insight into how this combined covariance matrix of midpoints actually encodes the model uncertainties due to motion non-uniformity, consider the last term in Equation 3.14. It was discussed above that modeling errors are added to $\Lambda_{t_{pred}}$ to account for perturbations. The last term thus transforms $\Lambda_{t_{pred}}$ into the 6×6 matrix

$$\begin{bmatrix} \Lambda_{t_{pred}} & \Lambda_{t_{pred}} & \Lambda_{t_{pred}} \\ \Lambda_{t_{pred}} & \Lambda_{t_{pred}} & \Lambda_{t_{pred}} \\ \Lambda_{t_{pred}} & \Lambda_{t_{pred}} & \Lambda_{t_{pred}} \end{bmatrix}$$

Thus, the modeling errors induce covariances across the lines in the predicted model and achieve the coupling desired for the search for an appropriate match. This has the effect of allowing the model to rigidly translate within a given region of uncertainty and still find a good match if one exists.

The match measure is the Mahalanobis distance between the 6×1 midpoint vectors (M'_m and M_m of Equation 3.11), and the orientations and lengths of the constituent lines in the predicted model and the potential match structure. That is, $mmeas$ is given by,

$$\begin{aligned}
mmeas = & (M'_m - M_m)^T (\Lambda_{M'_m} + \Lambda_{M_m})^{-1} (M'_m - M_m) \\
& + \sum_{i=1}^3 [(\Lambda_{\theta'_i} + \Lambda_{\theta_i})^{-1} (\theta'_i - \theta_i)^2 + (\Lambda_{l'_i} + \Lambda_{l_i})^{-1} (l'_i - l_i)^2] \quad (3.15)
\end{aligned}$$

Here Λ_{M_m} is the covariance matrix of midpoints of the potential data matches with the three $2 \times 2 \frac{1}{2} \Lambda_{xy}$'s (Equation 3.8) along its diagonals. $\Lambda_{\theta'_i}$, Λ_{θ_i} , $\Lambda_{l'_i}$ and Λ_{l_i} are the variances in orientations and lengths of the constituent lines in the predicted and the potential match structures.

3.7 Shallowness as Affine Trackability

In this section, we use the formulations of affine describability and model-matching developed earlier to design an algorithm to decide whether or not a hypothesized structure is shallow based on its trackability as an affine structure.

As mentioned earlier, we are interested in applying the system to man-made environments where most surfaces are smooth and largely textureless, and lines provide a fairly complete description of the image in terms of surface boundaries and significant surface markings. In general, shallow structures in the image are composed of only a few lines. Thus we cannot rely on Hough-like clustering techniques over two frames, where every primitive structure votes for a set of affine parameters and sets of structures with similar parameters are clustered as shallow structures [1]. However, the evolution of a hypothesized structure over time is an alternative source of measurement which can be used to check the validity of a shallowness hypothesis, even when it involves a set of only a few lines. The essential idea is that if a hypothesized structure can be consistently tracked and its 3D depth over time is consistent with a shallow structure model, then the structure is identified as shallow, otherwise it is labeled non-shallow.

3.7.1 Cycle of Prediction and Matching

A hypothesized aggregate structure, as defined in Section 3.6, undergoes a cycle of prediction and matching over a sequence of frames, with both the location and dynamic state vectors updated for each frame, before it is declared shallow or non-shallow. The process consists of the following phases:

- Bootstrap Phase
- Tracking Phase consisting of *Prediction, Matching and Update*.

Bootstrapping occurs only once for every new structure instantiated in any frame. The three parts of the tracking phase are repeated cyclically. Instead of always representing the motion and depth between consecutive frames, a moving window of, say m , frames is considered. The first frame in this window is called the *anchor frame*. The anchor frame for a freshly instantiated structure is its frame of instantiation. For every newly acquired frame in the window the motion parameters are computed and depth represented with respect to the anchor frame. This improves reliability of motion and location computations over time because the magnitude of motion starting from the anchor frame increases successively with every newly acquired frame. The following description assumes that the translation possesses a z -component (translation in depth) along with the x and y components. It can be easily modified for the cases when the z -component is zero. Also, a non-zero z -component of translation is necessary if the scale parameter is to be used for depth computation. It is also expected that the magnitude of the translation, say T_z , is known; otherwise all depth computations are with respect to a scale of unity.⁴

⁴Recall that this scale factor is not recoverable with any monocular motion algorithm.

Bootstrap Phase

For a newly instantiated structure (nominally a triple of lines), the motion of the structure is unknown. The line tracking algorithm of Williams and Hanson [96], which matches lines to their displacement field-based predictions, is used to generate correspondences in frame 2. A sample of this matching is shown in Figure 3.4. In many instances, the flow-based predictions can lead to multiple matches. These are disambiguated by choosing the one with the best match measure of Equation 3.15. Using the correspondences thus derived, the initial affine motion parameters and their covariances are computed (Equations 3.7 and 3.9).

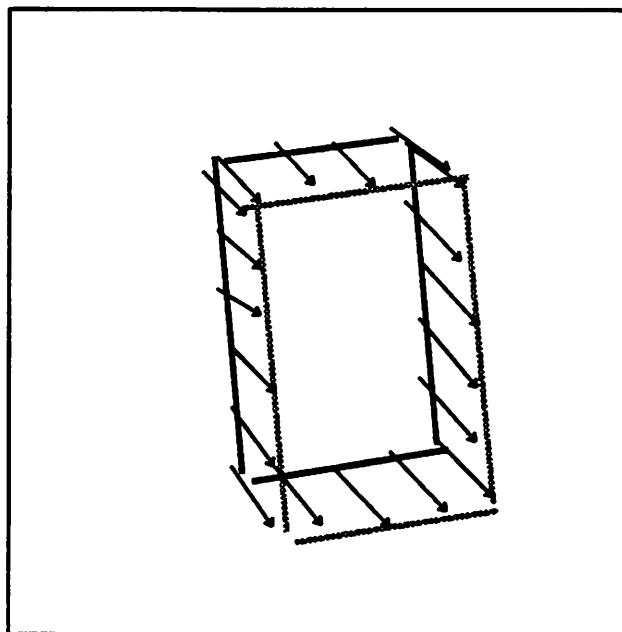


Figure 3.4: Bootstrap matching using flow. Lines in bold are frame 1 lines and those in lighter gray are frame 2 lines. The displacement vectors approximately along the length of the lines are shown as lines with arrows.

Tracking Phase

In the *prediction phase* of tracking, at time t , the motion parameters between the current anchor frame 1 and frame t in the current window are used to predict the motion between frames t and $t+1$. The covariances are propagated into frame $t+1$ as well. The predictions assume uniformity of motion but non-uniformity is dealt with by modeling the uncertainties in the predictions within the framework developed in Section 3.6.2. So the motion parameters between t and $t+1$ are,

$$\begin{aligned} s_{pred} &= 1 / \left(1 + \frac{1-s}{t-1} \right) \\ \omega_{z_{pred}} &= \omega_z / (t-1) \\ t_{pred} &= \frac{1}{t-1} t \end{aligned} \tag{3.16}$$

Under the assumption of small rotations, these provide fairly good predictions for uniform motion. From the computed covariances of the affine parameters r_s and t in Equation 3.9, the covariances of the predictions can be easily derived. These are called $\Lambda_{r_{s_{pred}}}$ and $\Lambda_{t_{pred}}$, as in Section 3.6.2. It was shown there how these are employed to handle deviations from uniformity.

The predicted affine motion parameters are used to project each line in the aggregate structure at frame t into its position in frame $t+1$ to obtain the predicted structure. Around each predicted line, a window query is performed to obtain potential matches for each line. Let L_1 , L_2 and L_3 be the three potential match sets for each line respectively in an aggregate triple of lines. Then all the triples from the product set $L_1 \times L_2 \times L_3$ are matched against the prediction.

In the *matching phase*, the match measure of Equation 3.15 is computed for each potential data triple against the prediction, and the best triple below a threshold is chosen as the match. This threshold depends on the model of measurement errors in lines, the allowable non-uniformity in motion, and the extent to which the real 3D

structure is not a fronto-parallel plane. If all the errors are assumed to be Gaussian, then the Mahalanobis distance of Equation 3.15 has a chi-squared distribution with the appropriate degrees of freedom [10]. A threshold on this distance can be chosen by using the chi-squared value corresponding to a desired level of confidence in accepting a match. However, it is not reasonable to assume that all the sources of error are Gaussian. For instance, errors in prediction arising from the departure of a structure from being a fronto-parallel plane cannot, in general, be modeled as Gaussian. This is especially true when the structure consists of a small number of tokens as is the case here. In such situations, the error in modeling the structure dynamics can be systematic. Ideally, an on-line determination of this process noise [63] is desirable so that a threshold for this source of error can be automatically chosen based on the allowable departure from shallowness. In order to accomplish this, more theoretical work along the lines of adaptive filtering needs to be done. In our implementation, the chi-squared values have been used in conjunction with an experimentally determined threshold. It is to be emphasized that in all the experiments, the tracking has been found to be robust for the choice of the same threshold throughout.

Once an acceptable match is found, the model's new motion parameters are computed between the anchor frame and the current frame using the newly found matches. This is called the *update phase*. The covariances of the current location vector and the computed affine parameters are also recomputed. Since depth is a part of the location vector, it is also updated. Additionally, the variance-weighted sample mean and sample dispersion of depth are updated by incorporating the new measurement.

An acceptable match may not be found in the current frame due to failures of line grouping, occlusions and motion discontinuities. The algorithm allows for graceful handling of many of these conditions by upgrading the current prediction to model status whenever a suitable match is not found. That is, the prediction serves as the

best current model in the absence of a good match to the data. A counter which keeps track of the number of frames missed is also incremented. In addition, the variances of the model's motion parameters and those of the line segments for its potential matches in the next frame are increased, and consequently, the search window for the next prediction/matching phase is expanded. If a match is re-acquired after a lapse in the previous frame, the motion variances and the window size are reduced, but not below the levels at the start of tracking.

There is an issue of computational complexity versus the temporal persistence of a model when a match is not found. After every frame in which a match is not found, the search windows become larger thus increasing the number of potential matches. This leads to an increase both in the computational expense for matching and the possibility of false matches. In general, there is no theoretically sound mechanism to address this problem because a combination of failures can always be designed to defeat any mechanism. However, any practical system, in which this algorithm is embedded, can place hard computational bounds on the time spent on search. If this maximum limit is reached then it might be reasonable to abandon the current model being tracked. For instance, consider the case where an object is occluded and either remains occluded for a large number of frames, or undergoes a significant change in motion (say reverses direction) while it is occluded. In general, the actual position of the object could be far away from the predicted location when it is reacquired by the system. In such a case, it seems reasonable to abandon the current model and to re-instantiate a new model for the object when it is again seen in the image.

The last three cycles of the tracking phase discussed above are repeated for every new hypothesized aggregate structure within its window of frames. If 1) it has been tracked for more than half the frames in the window, and 2) its depth dispersion is within an allowed limit, and 3) its projection error (Equation 3.12) for all matched

frames is less than a threshold, then it is declared as a shallow structure, else it is not and is dropped from further consideration.

3.7.2 The Algorithm

The algorithm presented above can be applied to image data in either an interactive mode or in an automatic mode. In the interactive mode, a set of manually selected lines is presented to the algorithm as a hypothesized shallow structure. The algorithm tracks the structure as described above and declares it shallow or otherwise.

In the automatic mode, triples of lines all over the image are instantiated as hypothesized aggregate structures and the algorithm automatically cycles through them and labels any given structure as shallow or non-shallow. We employ proximity and convexity as generic heuristics to create triples of line tokens as aggregate hypotheses. In most man-made environments, appearances of objects can be described as convex regions or a union of significant convex regions enclosed by boundaries. Each pair of line segments in a triple should be completely contained in the half-space defined by the remaining line extended infinitely. Some amount of tolerance is allowed in testing for the half-space containment in that a small part of the line can straddle the half-space defining line and still qualify. Triples passing this convexity test are represented as hypothesized models and the above algorithm is applied to each one. The result is a labeling of structures in the scene as shallow and non-shallow.

The complexity of extracting triples out of image lines is $O(n^3)$, where n is the number of lines. This can be considerably improved upon by using proximity as a heuristic. Around each endpoint of a line, all lines within a given distance are chosen, and the convexity test is applied to these sets of lines. The complexity of spatial queries based on proximity is $O(1)$ if the image lines are pre-processed and are hashed into a spatial grid defined over the image plane [20]. It is reasonable to assume, and we have found it to be so in our experiments, that the number of lines

in the proximal line sets is bounded by a small constant. Thus, the complexity of finding triples is almost always $O(n)$ with a fairly small constant (small compared to n). Consequently, the number of approximately convex triples found is also $O(n)$. We will present specific numbers to illustrate this in the next section.

The inner core of the algorithm for either mode of application is the same and is presented here.

Given a set of lines constituting a hypothesized shallow structure in frame 1, the following tracking algorithm is applied. The tracking is done for a few frames before the structure is labeled. Also, the first frame in the sequence is the *anchor frame*, that is, the affine parameters are computed between this frame and every new frame. This improves reliability of motion and depth computations over time because the magnitude of motion displacement starting from the anchor frame is expected to increase with every newly acquired frame.

Various steps of the algorithm are:

Step 1: Bootstrap

Compute the line matches for frame 2 using flow-based predictions [96].

Compute the affine motion parameters and their covariances.

(Equations 3.7 and 3.9).

Instantiate a model with its location and motion states.

If (less than 3 matching lines found) declare *Non-trackable* and exit.

For every new frame t , Repeat until frame m processed:

Step 2: Prediction

Compute the predicted parameters between time t and time $t + 1$.
(Equation 3.16).

Project the current model lines at t into predicted lines at $t + 1$.

Compute the covariances of the predicted model using the covariances of motion and data at t , and the model noise covariances accounting for non-uniformity (Equation 3.15).

Step 3: Find potential match sets, L_1 , L_2 and L_3 of data lines for each predicted model line: (a) within the model line's search window, and (b) selecting data lines within a *δorientation* (typically, 15°) of the model line.

Form the product set $L_1 \times L_2 \times L_3$ from the potential match sets for each line.

For each element of this set, compute the Mahalanobis distance between the element and the predicted model (Equation 3.15).

Choose the best match below a threshold.

If (none) increment *no-match-count* else decrement
no-match-count.

Step 4: Update

If match found then

Compute the new affine parameters between frame 1 and the matches in
frame $t + 1$ and the associated covariances.

Update the sample weighted-mean and dispersion for the depth parameter
in the model.

else

Promote the prediction to model status with increased modeling noise
(Equation 3.14).

end Repeat.

Step 6: If *no-match-count*, *depth-dispersion* and $merr_{proj}$ (Equation 3.12)
are all less than their thresholds (Section 3.7.1),

then declare model set as *shallow* else *non-shallow*.

The threshold for *depth-dispersion* is in general chosen to be some percentage
(typically, 10 or 20) of the mean depth. Threshold for $merr_{proj}$ is chosen based on

considerations which were discussed in Section 3.7.1. It is reasonable to allow the *no-match-count* to be a fraction (typically 1/3) of the number of frames in a window.

3.8 Experimental Results

The implementation of our algorithm was greatly facilitated by the use of a Lisp based database called the ISR [20] running on a TI Explorer II. The aggregate structures and their potential matches are instantiated through spatial queries and represented by ISR token sets.

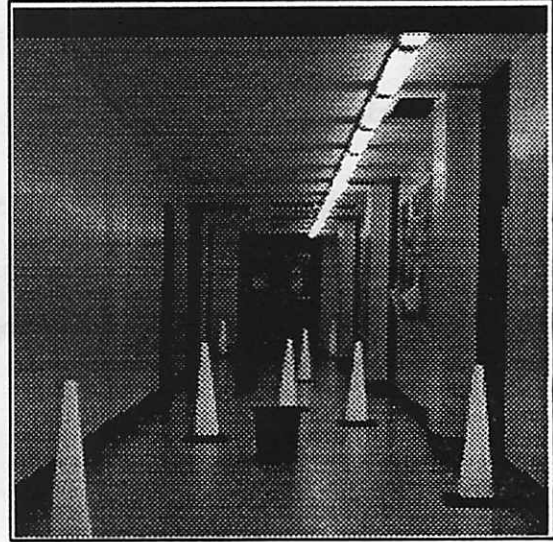
3.8.1 Tracking Results

We present the tracking results on two image sequences, *cones-seq* and *room-seq-1*, both of which were captured with a SONY B/W AVC-D1 camera, with effective FOV 24 by 23 degrees mounted on a Denning robot, and digitized to 256-by-242 pixels. The camera moved into the scene with a translation magnitude measured to be approximately 1.95 feet for the *cones-seq*, and 0.39 feet for the *room-seq-1* between successive frames. Four image frames for each of these sequences are shown in Figures 3.5 and 3.6, respectively. It is emphasized that the effective motion is neither purely translational nor uniform. In each frame lines are extracted using Boldt's [14] line grouping system. A window of six frames is used for most of the results here. However, for the *room-seq-1*, some interesting aspects of the algorithm are shown with ten-frame windows.

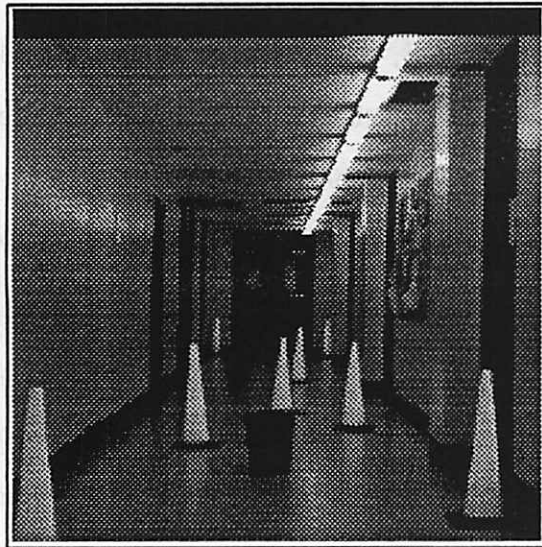
For both sequences, Figures 3.8-3.16 are to be read left-to-right and top-to-bottom. In each figure, panels a) and b) show the hypothesized aggregate of lines overlaid in black or white on the first and the last images, respectively. Panel c) shows the structure highlighted in bold and overlaid on lines in frame 1. Panel d) highlights the corresponding structure in frame 2; the correspondence was derived in



a) Image Frame 1



b) Image Frame 3

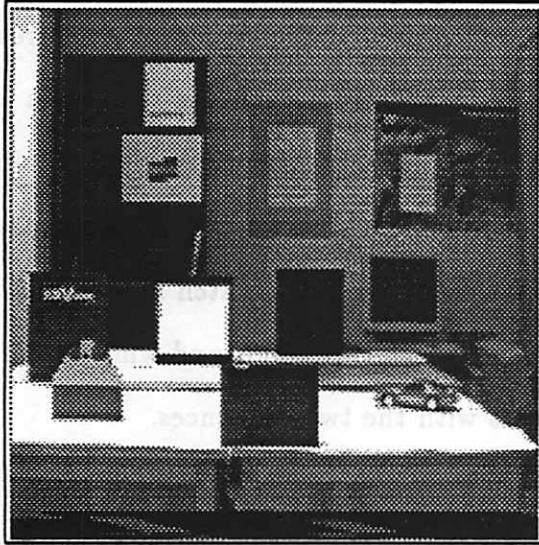


c) Image Frame 4

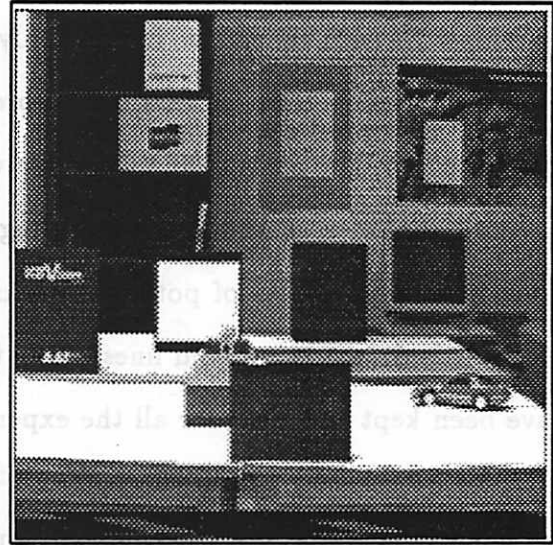


d) Image Frame 6

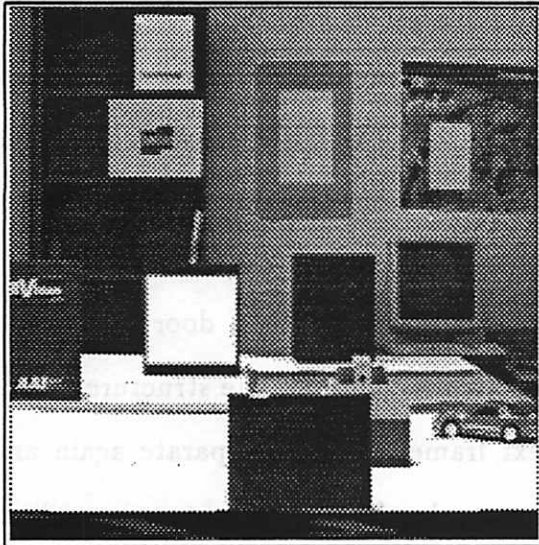
Figure 3.5: Four image frames of the *cones-seq*. Frames 1, 3, 4 and 6.



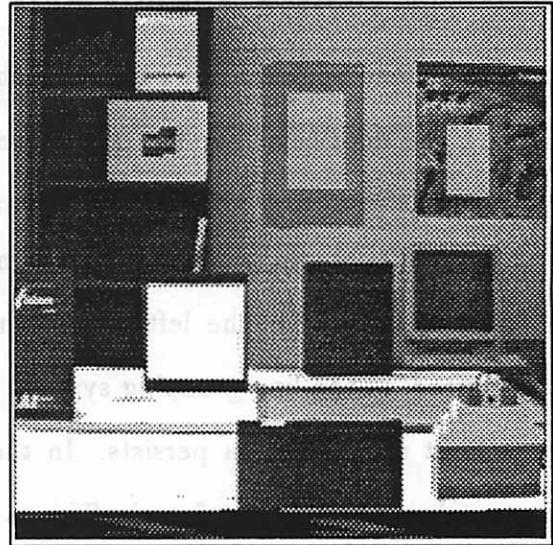
a) Image Frame 1



b) Image Frame 5



c) Image Frame 8



d) Image Frame 10

Figure 3.6: Four image frames of the *room-seq-1*. Frames 1, 5, 8 and 10.

the bootstrap phase using flow-based line tracking [96]. Subsequently, corresponding to each frame in the sequence, each panel, starting with panel e) onwards, depicts matching for each successive frame. Only the region around the structure of interest is expanded and shown in detail. The prediction windows for each line are shown as shaded areas. The central spine of these windows is the actual prediction. Thin lines show all the lines in and around the region of interest. Lines of medium thickness show the union of sets of potential matches for each line. If a match is found in a frame, it is drawn using bold lines. Note that the match thresholds and window sizes have been kept the same for all the experiments with the two sequences.

Figure 3.7 shows the image motion of the lines on the doorway at the far end of the hallway in the *cones-seq* over six frames. It is clear from the up and down motion of the image lines over time that the motion is definitely not uniform. Even on the smooth surface of the floor in the hallway, slight undulations lead to rotations in depth and around the optical axis. Figure 3.8 shows the tracking of the three-line structure defining the doorway. For this structure, the matching is fairly unambiguous, and this example sets up a reference for comparison with other tracking examples.

Figure 3.9 depicts tracking of three lines on a cone. An interesting event happens in frame 4 (panel f); the left line of the cone is merged with a door line in the background by the line grouping system. No match is found for the structure in this frame, but its prediction persists. In the next frame, the lines separate again and the match is successfully found. This is an example of how the system is resilient to overgrouping errors. In general, such failures which occur due to coincidence of viewpoint are not expected to persist over time. Thus, model persistence in the absence of reliable data can handle these situations.

Finally, for the *cones-seq*, Figure 3.10 shows the attempt at tracking a non-shallow structure. Two lines on a cone and one on a structure in the background have been chosen for this illustration. The figure illustrates the fact that an affine description

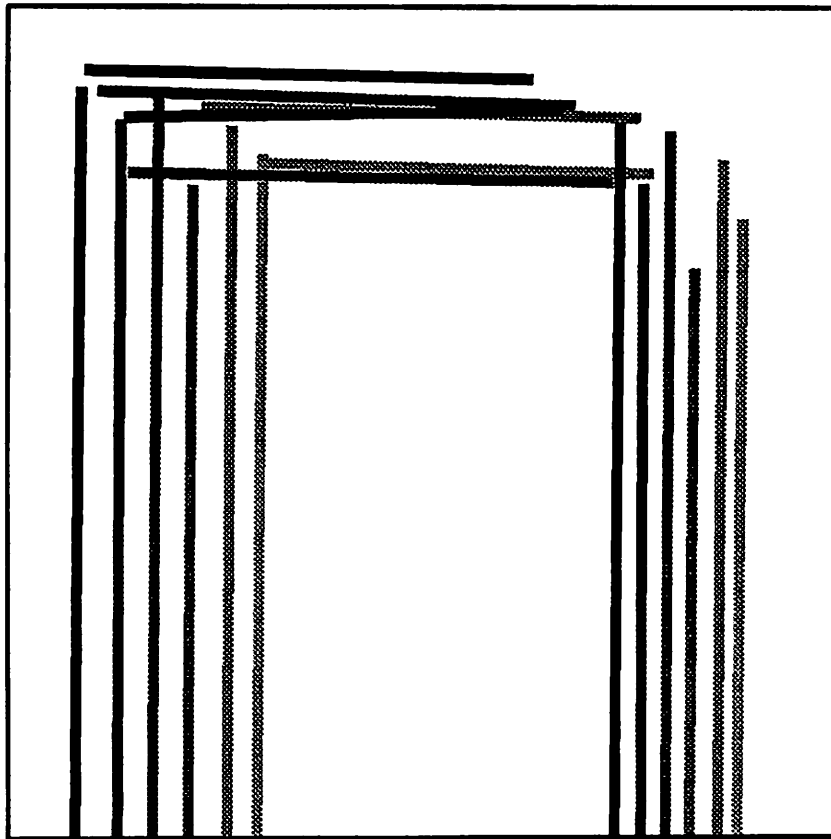
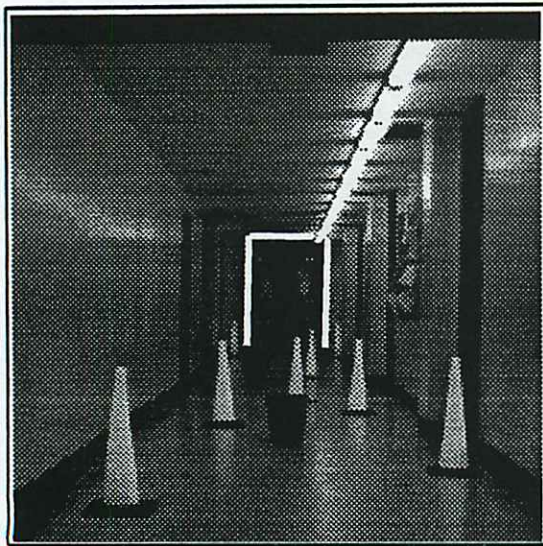
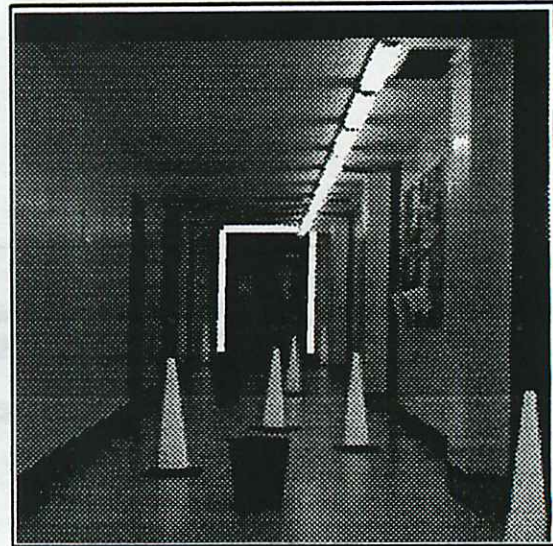


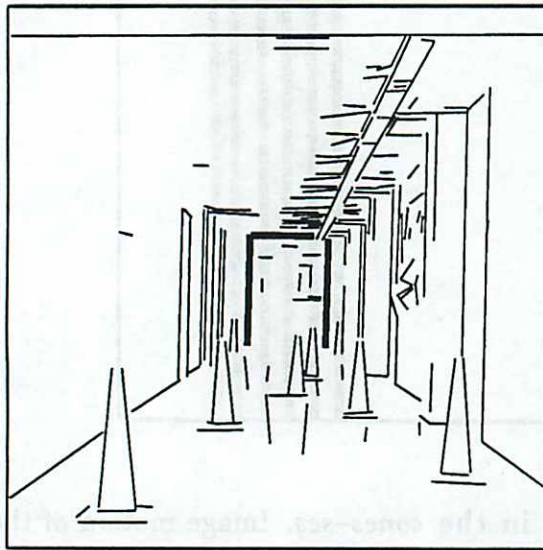
Figure 3.7: Motion of the doorway lines in the *cones-seq*. Image motion of the doorway lines from frames 1 to 6. Frame 1 lines are in the lightest shade and frame 6 in the darkest. The up and down motion in the image plane shows the non-uniformity of motion.



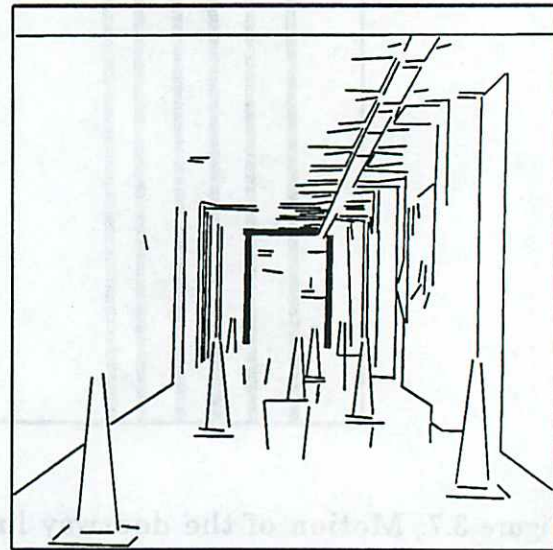
a) Image Frame 1



b) Image Frame 6

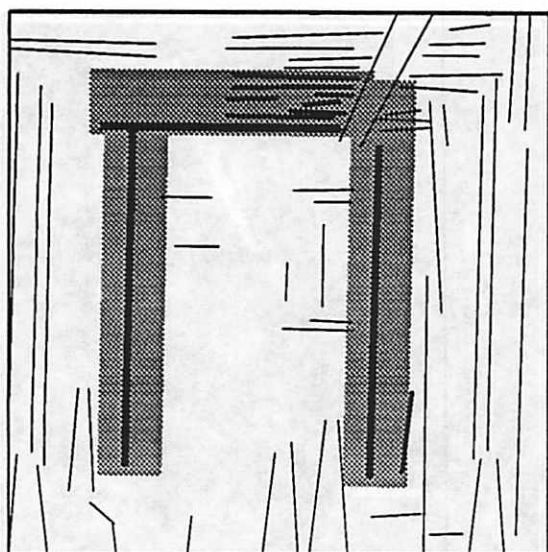


c) Frame 1

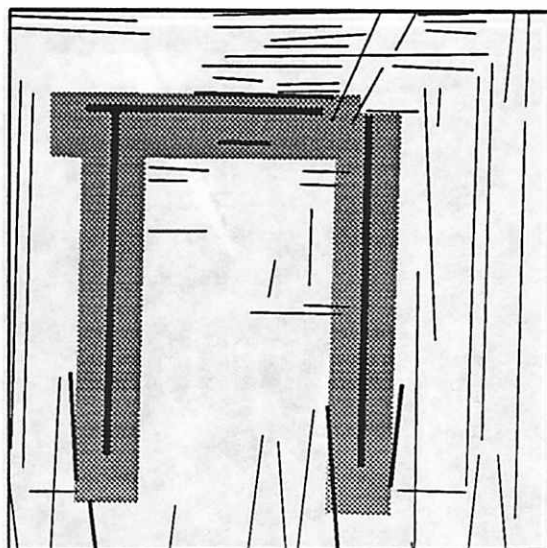


d) Frame 2

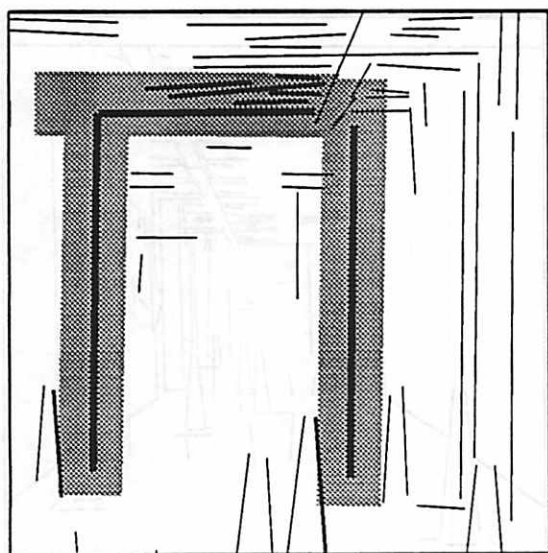
Figure 3.8: Tracking of the doorway in the *cones-seq*. Tracking over six frames is shown. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction in frame 1. (*contd. next page*)



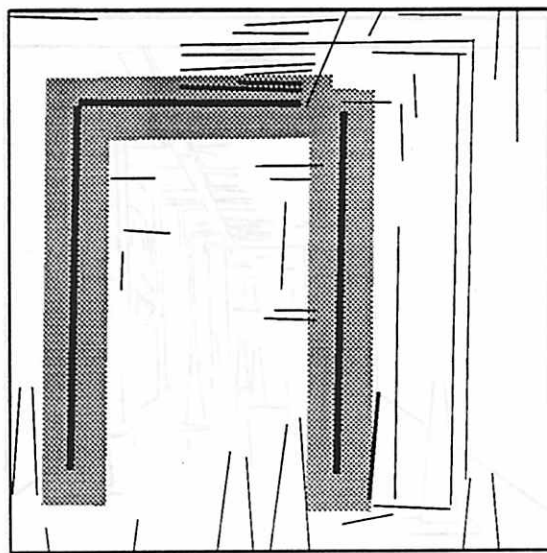
e) Frame 3



f) Frame 4

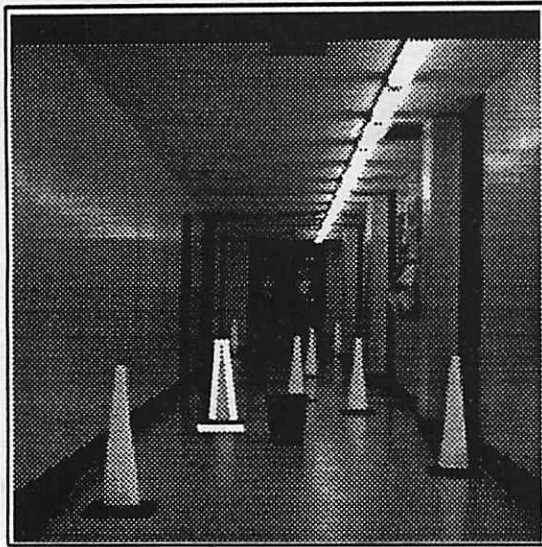


g) Frame 5

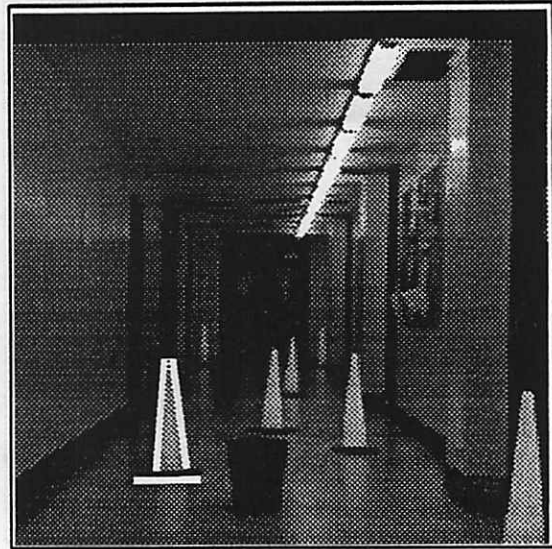


h) Frame 6

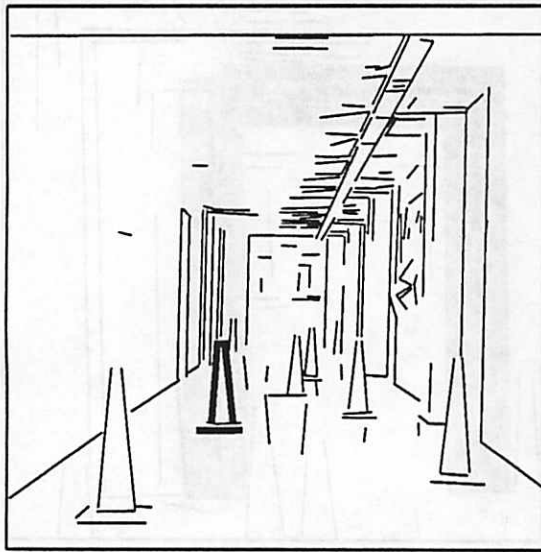
Figure 3.8: (contd.) e)–h): Matching and tracking in frames 3–6. The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found.



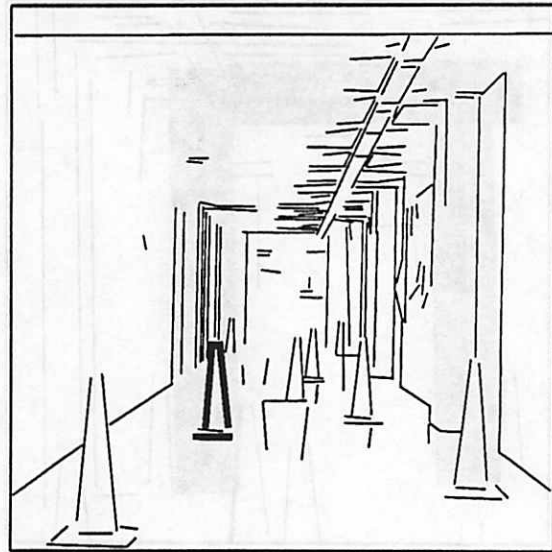
a) Image Frame 1



b) Image Frame 6

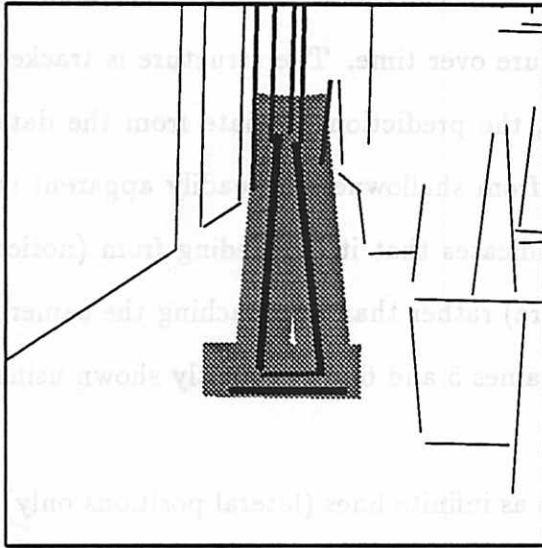


c) Frame 1

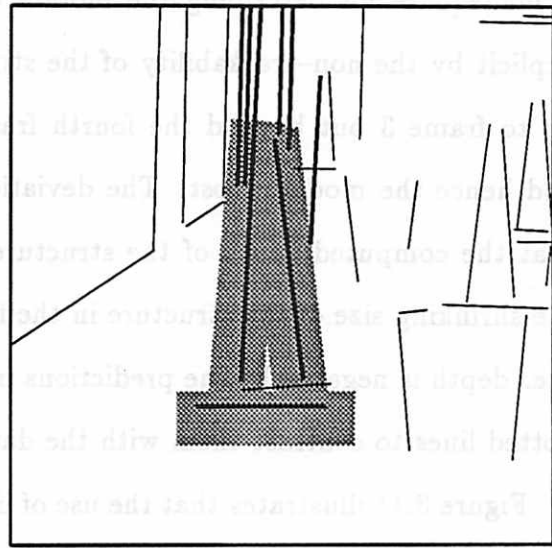


d) Frame 2

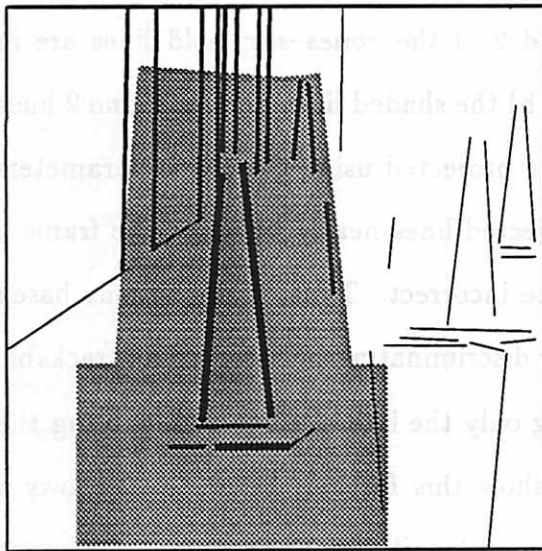
Figure 3.9: Tracking of a shallow triple in *cones-seq*. Tracking over six frames is shown. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction in frame 1. (contd. next page)



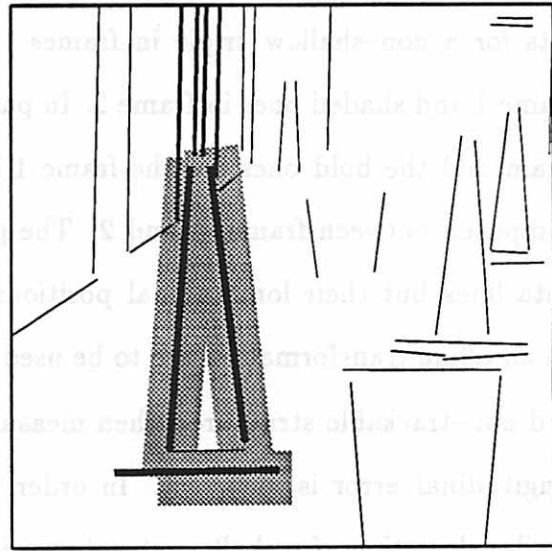
e) Frame 3



f) Frame 4



g) Frame 5



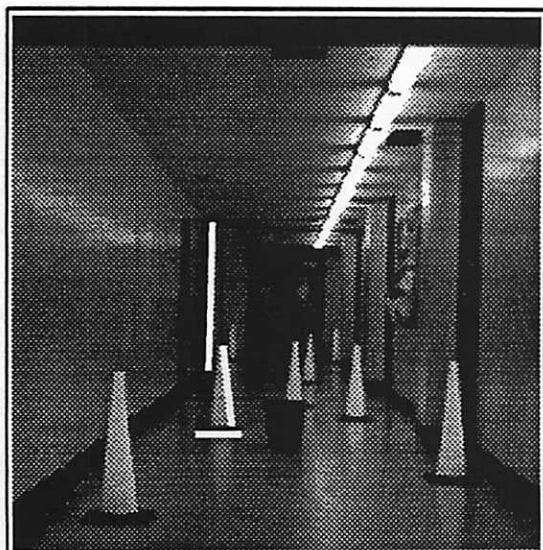
h) Frame 6

Figure 3.9: (contd.) e)–h): **Matching and tracking in frames 3–6.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. f): No match found due to overgrouping of the left cone line in frame 4; g) Recovery from error by model persistence in frame 5.

is inadequate for describing the motion of a non-shallow structure. This is made explicit by the non-trackability of the structure over time. The structure is tracked up to frame 3 but beyond the fourth frame, the predictions deviate from the data and hence the model is lost. The deviation from shallowness is readily apparent in that the computed depth of the structure indicates that it is receding from (notice the shrinking size of the structure in the figure) rather than approaching the camera (i.e. depth is negative). The predictions in frames 5 and 6 are explicitly shown using dotted lines to contrast them with the data.

Figure 3.11 illustrates that the use of lines as infinite lines (lateral positions only), when only small sets of lines are used for computing the affine transformation, is inadequate for affine tracking (Section 3.6). Panel a) shows two corresponding line sets for a non-shallow triple in frames 1 and 2 of the *cones-seq*; bold lines are in frame 1 and shaded ones in frame 2. In panel b) the shaded lines are the frame 2 lines again and the bold ones are the frame 1 lines projected using the affine parameters computed between frames 1 and 2. The projected lines nearly lie along the frame 2 data lines but their longitudinal positions are incorrect. Thus, if predictions based on an affine transformation are to be used for discriminating between affine trackable and non-trackable structures, then measuring only the lateral error and ignoring the longitudinal error is incorrect. In order to show this further, Figure 3.12 shows a similar depiction of a shallow structure with good longitudinal and lateral alignment of the projected and the data lines.

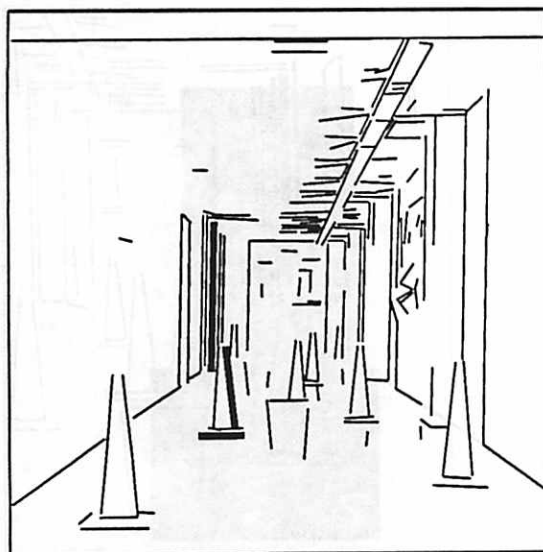
Figure 3.13 shows the results of tracking a four-line shallow structure for the *room-seq-1*. Figure 3.14 shows the image motion of a sample four-line structure to give an idea of the motion in the *room-seq-1*. The motion of the structure is shown from frames 3 to 8; the lightest shade is used for frame 3 and the darkest for frame 8. The motion discontinuity between frames 6 and 7 is apparent from the figure.



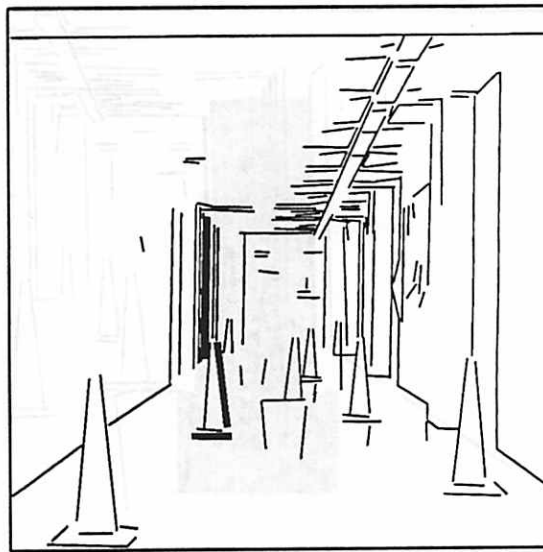
a) Image Frame 1



b) Image Frame 6

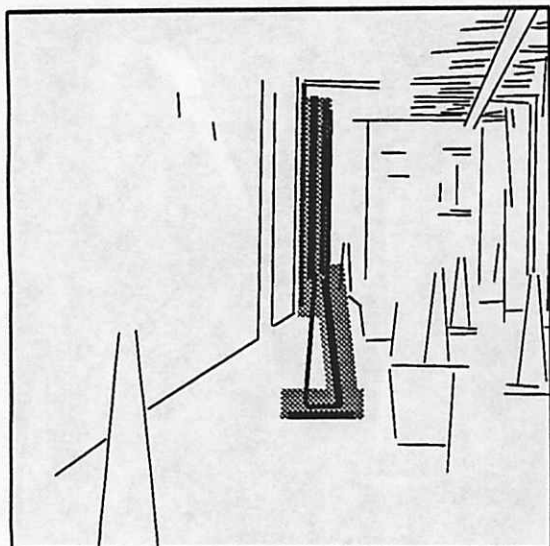


c) Frame 1

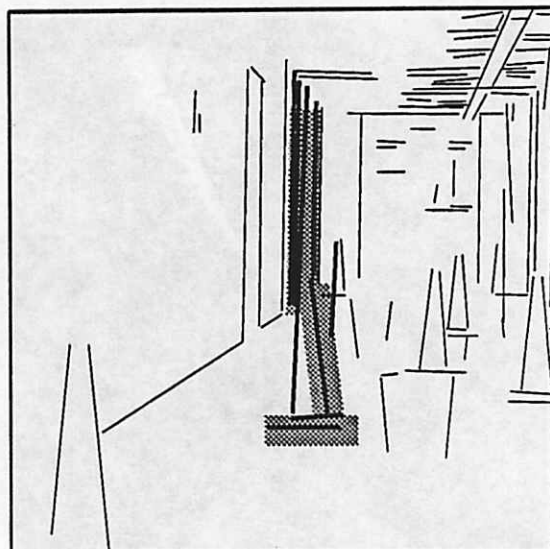


d) Frame 2

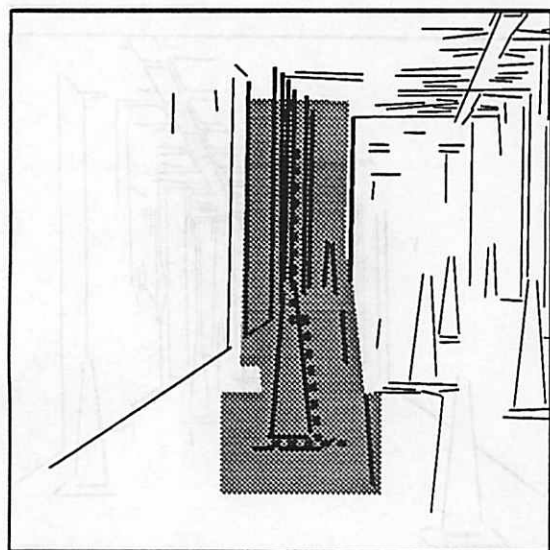
Figure 3.10: **Non-trackability of a non-shallow triple.** Two lines on a cone and one of the doorway lines in the background. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction in frame 1. *(contd. next page)*



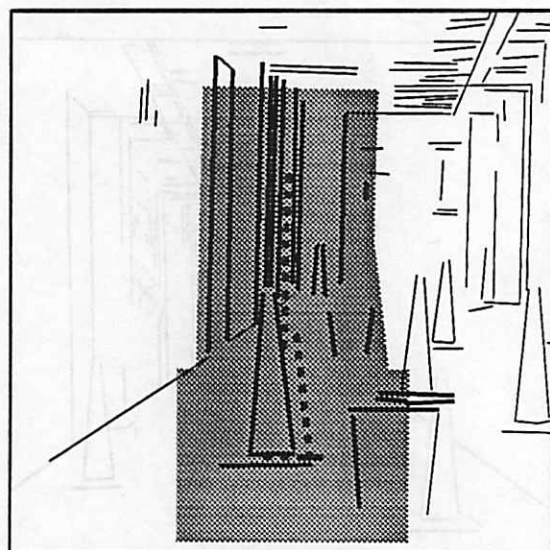
e) Frame 3



f) Frame 4



g) Frame 5



h) Frame 6

Figure 3.10: (*contd.*) **Non-trackability of a non-shallow triple.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. f)–h): No match found. The prediction is shown as dotted lines in frames 5 and 6.

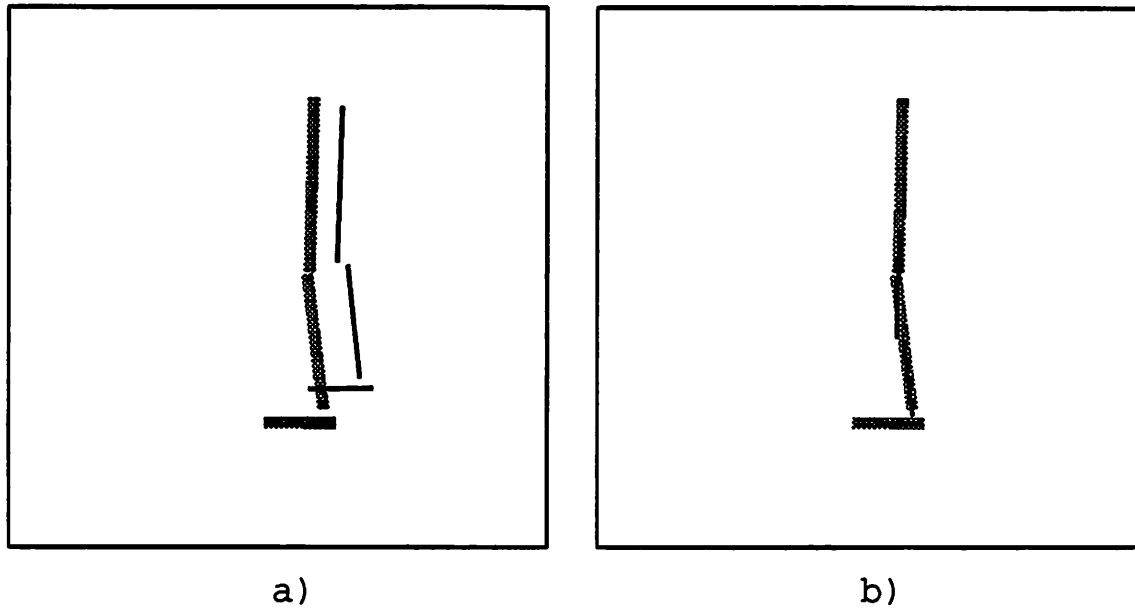


Figure 3.11: Longitudinal error in affine projection of a non-shallow structure. a) Frame 1 triple in bold and Frame 2 shaded. b) Frame 2 triple shaded and the affine projected Frame 1 triple bold.

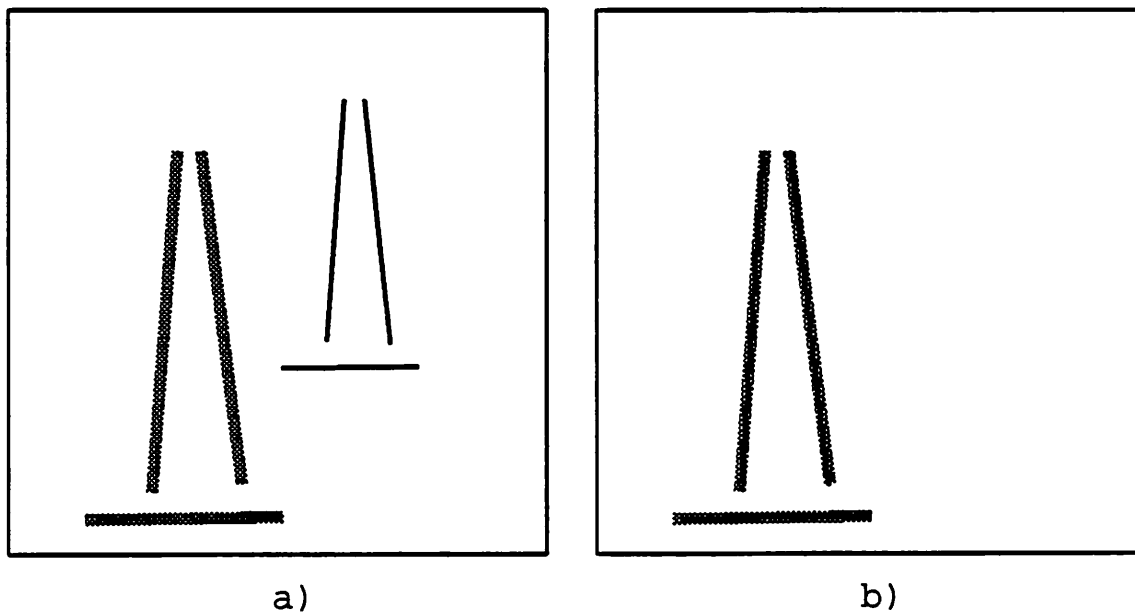


Figure 3.12: Longitudinal error in affine projection of a shallow structure. a) Frame 1 triple in bold and Frame 2 shaded. b) Frame 2 triple shaded and the affine projected Frame 1 triple bold.

In Figures 3.15 and 3.16, the window is extended to ten frames to show how the algorithm handles a motion discontinuity (Figure 3.15) and an independently moving object (Figure 3.16) which is occluded/deoccluded during its course of motion.

In Figure 3.15, a shallow triple is tracked. Note that in frame 5 (panel g), a break in one of the lines occurs with the result that the matching fails but the correct model is reacquired in the next frame. Also, note that this triple is surrounded by a similar triple throughout the sequence (they are on the same planar surface). In frame 3, the right hand side line of the predicted triple (the central spine of the vertical shaded region in the figure) lies almost along the position of the corresponding line in the incorrect triple. A matching algorithm based on individual line matches would have matched to the incorrect data line, but because of covariance based aggregate matching in the algorithm, the tracking system matches to the correct triple. Between frames 6 and 7 (panels h and i), there is a change in the motion; it is as if the robot started going up a gently sloping "hill". Consequently, the prediction and the data move in opposite directions. Although no match is found in frame 7, the prediction persists with expanded windows and variances and the model is reacquired in frame 8.

Finally, for the *room-seq-1*, we demonstrate the algorithm on an independently moving object with occlusions. Figure 3.16 shows an object constructed from Lego blocks being tracked as it goes behind another surface (in frame 5) and re-emerges (in frame 8) as the camera moves towards it (see also Figure 3.6). Note that the non-uniformity in motion mentioned above, between frames 6 and 7, further complicates the tracking because during these frames the object remains hidden and the error in prediction increases dramatically. However, the object is still reacquired when it re-emerges in frame 8 (panel j). This example serves as a demonstration of the algorithm's potential use in sequences containing both camera motion and independent object motions.

Note that the mechanisms of model persistence and model uncertainties have been demonstrated to successfully handle all the three types of tracking failures — line grouping errors, motion discontinuity and occlusions. The related issue of computational complexity of matching and the allowable limits on model persistence were discussed in Section 3.7.1.

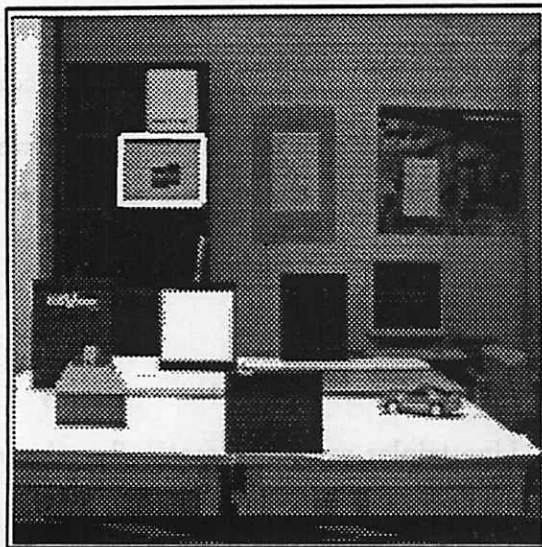
3.8.2 Segmentation and Reconstruction Results

The algorithm in Section 3.7.2 was applied to the *cones-seq* and the *room-seq-1* to identify the shallow structures in the scene. Line triples were automatically selected to hypothesize aggregate structures. Each of these was tested for affine trackability resulting in its labeling as a shallow or a non-shallow structure. Figures 3.17 and 3.18 show the structures identified as shallow by the algorithm in the two sequences. In the *cones-seq* and the *room-seq-1*, 121 and 79 triples were found out of a total number of 167 and 180 lines, respectively. This supports our hypothesis in Section 3.7.2 that the order of the number of triples found using pruning by proximity is typically closer to linear than cubic in the number of total lines.

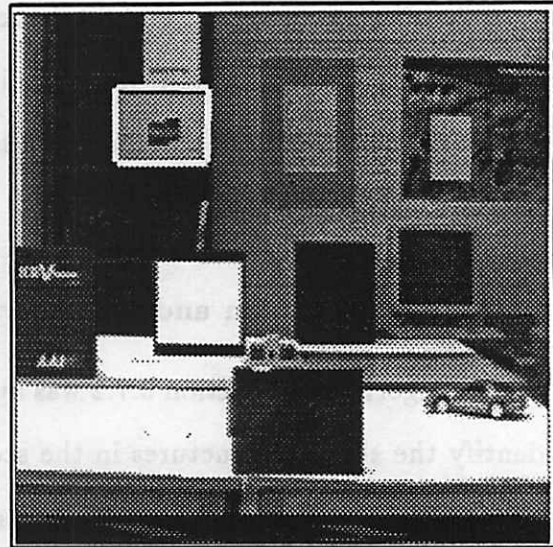
In the *cones-seq*, the two cones in the center of the image behind the trash can are merged together as a single shallow structure. This is because a) they are close to the FOE, and b) are far enough away so that their image motion is small.

The depth of some salient structures was measured with a tape measure. Tables 3.1 and 3.2 show a comparison of this ground truth with the computed depths for the *cones-seq* and the *room-seq-1*, respectively. The objects referred to in the tables are labelled in Figures 3.19 and 3.20. The average percentage depth errors for the *cones-seq* and *room-seq-1* are 3.8% and 3.4%, respectively.

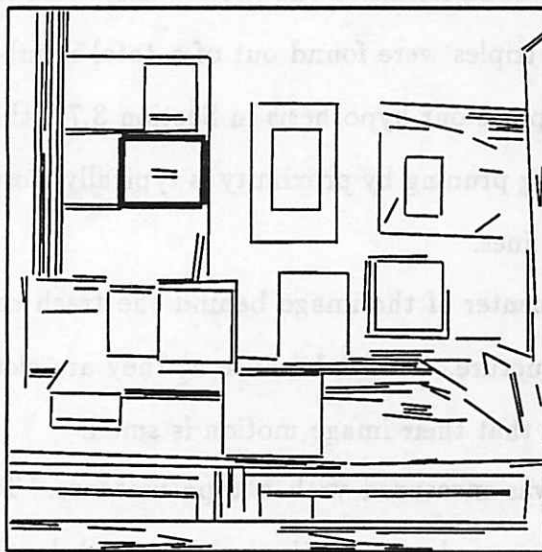
Now we present results of depth computation using the four-parameter affine description for planar objects in a scene which are at a variety of slant angles. The results are illustrated on an image sequence, called the *comp-seq*. Two image frames



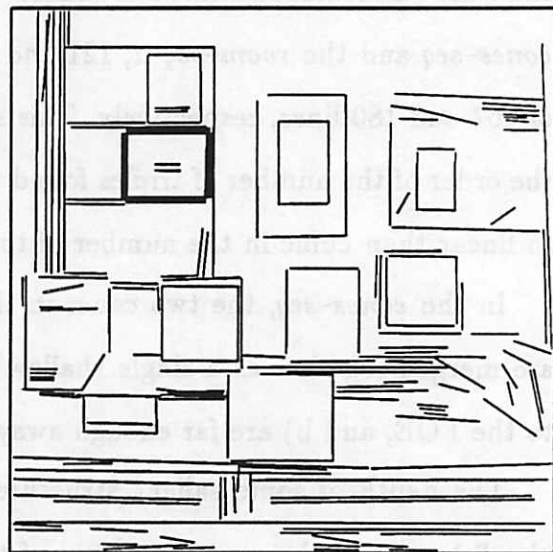
a) Image Frame 1



b) Image Frame 6



c) Frame 1



d) Frame 2

Figure 3.13: Tracking of a shallow 4-line structure in the *room-seq-1*. a), b): First and last image frames with the structure highlighted; c), d): Highlighted structure overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction in frame 1. (contd. next page)

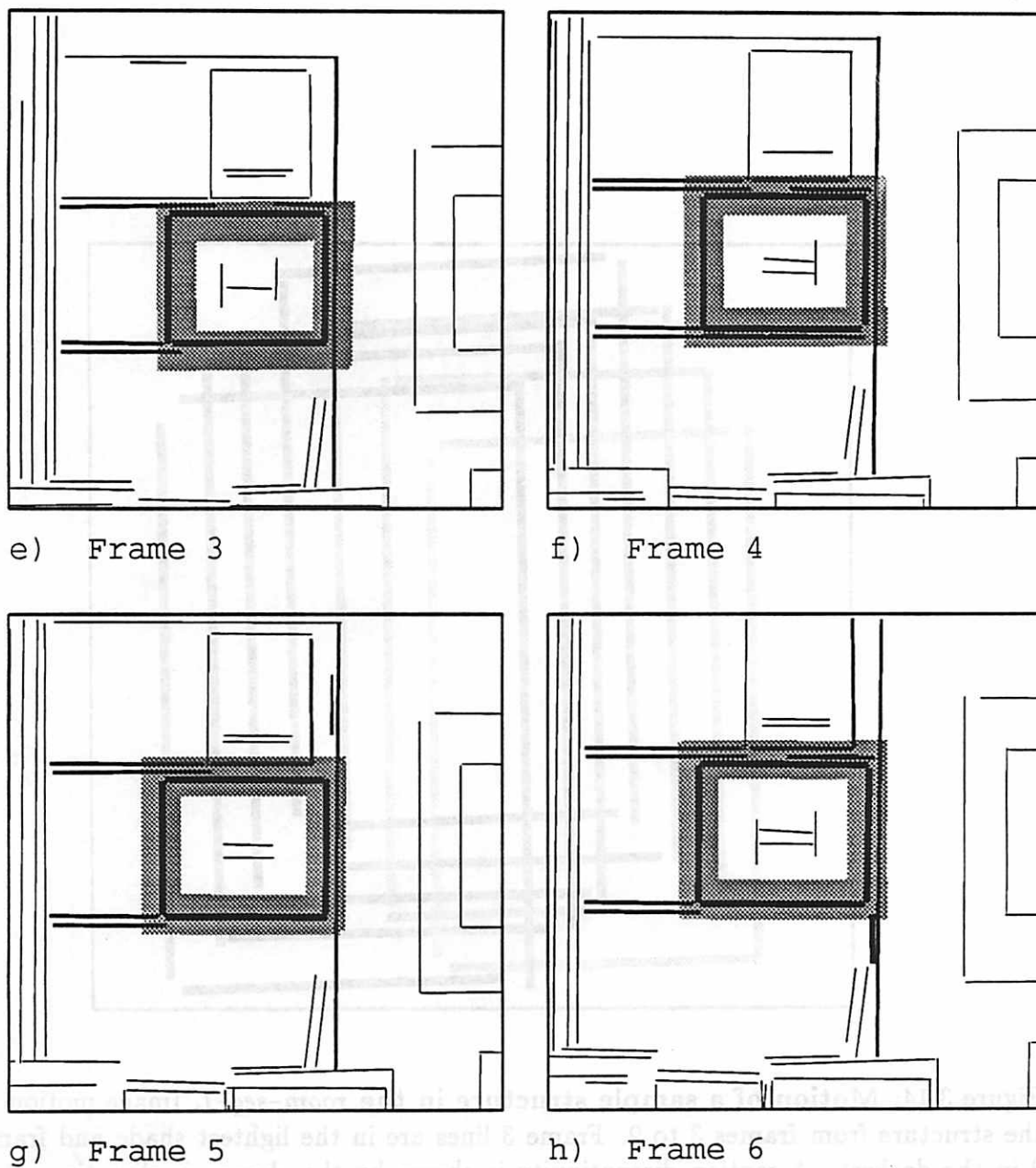


Figure 3.13: (*contd.*) **Tracking of a shallow 4-line structure over six frames in the *room-seq-1*.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found.

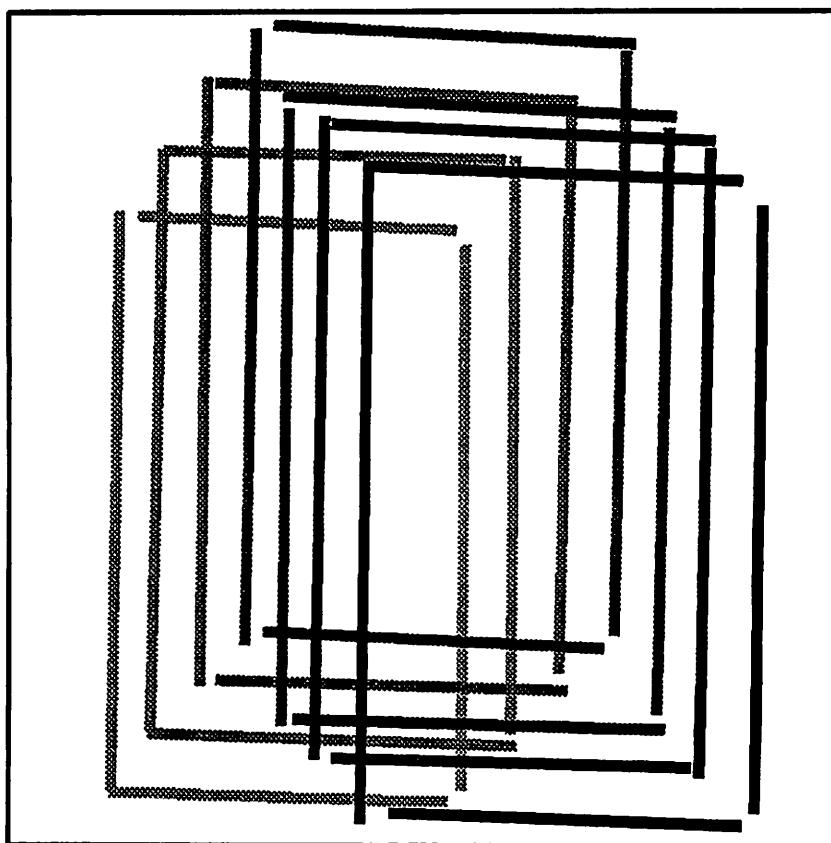
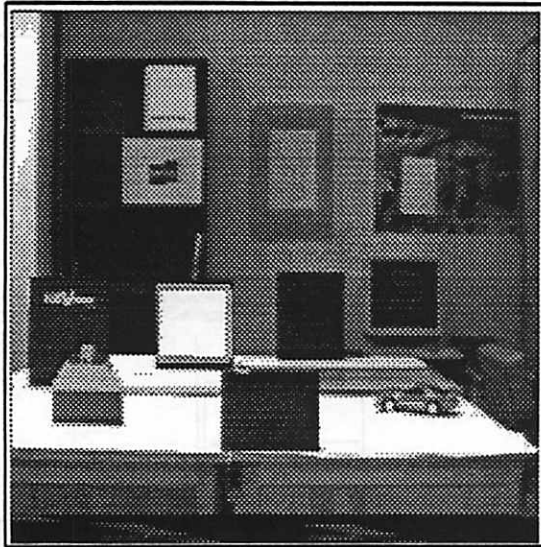
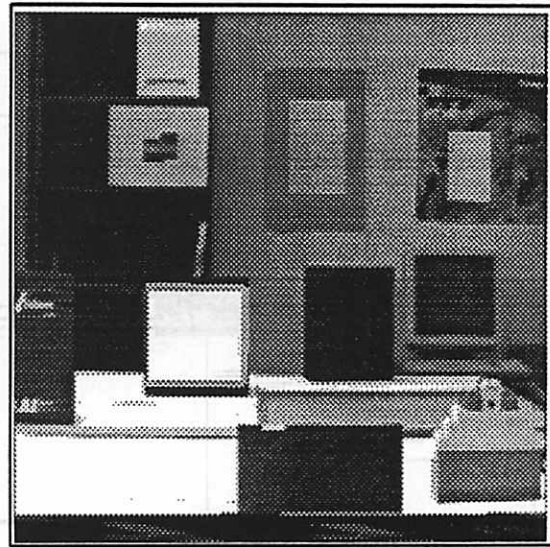


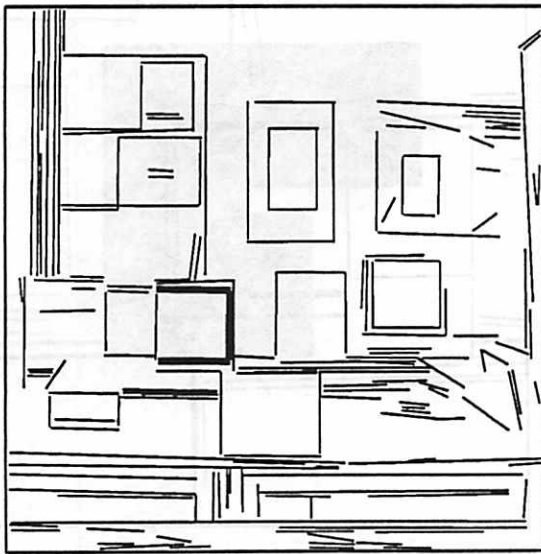
Figure 3.14: Motion of a sample structure in the *room-seq-1*. Image motion of the structure from frames 3 to 9. Frame 3 lines are in the lightest shade and frame 9 in the darkest. A motion discontinuity is shown by the change in direction after frame 6.



a) Image Frame 1



b) Image Frame 10

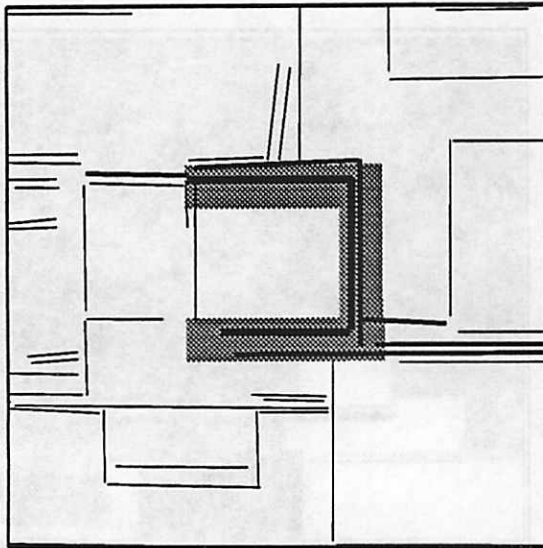


c) Frame 1

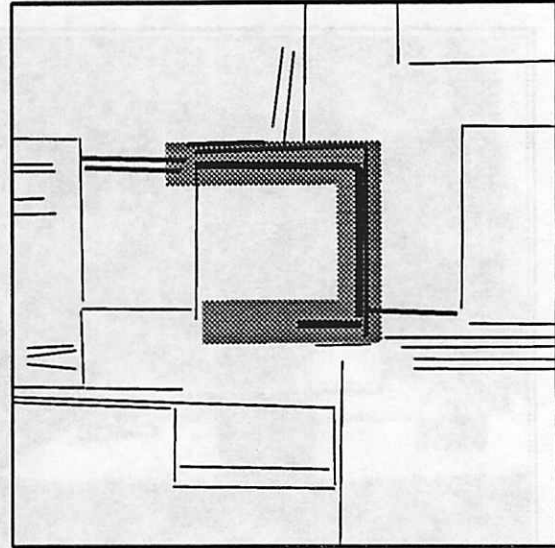


d) Frame 2

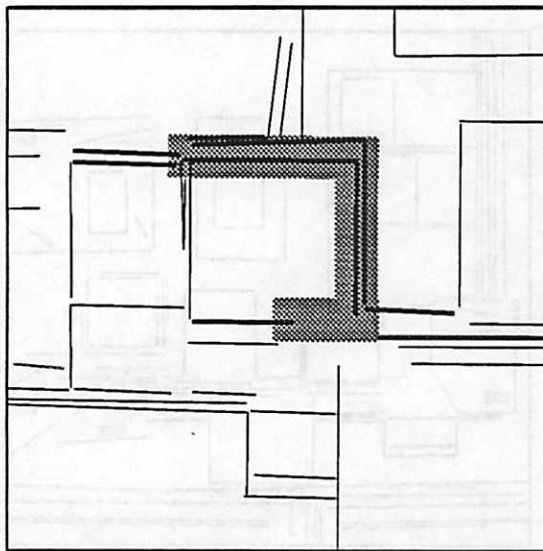
Figure 3.15: Tracking of a shallow triple in the *room-seq-1*. Shown for ten frames. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction in frame 1. (contd. next page)



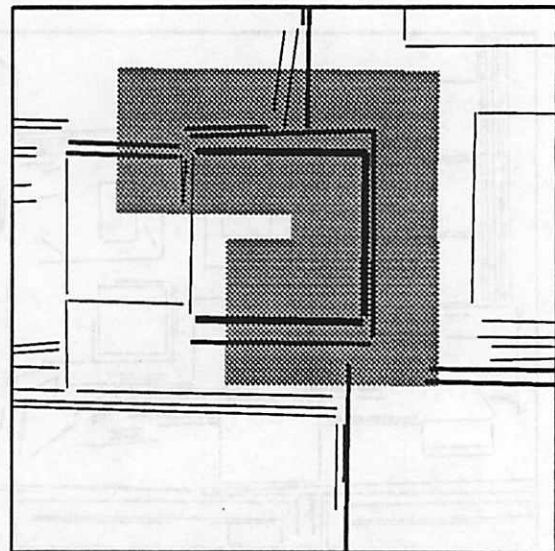
e) Frame 3



f) Frame 4

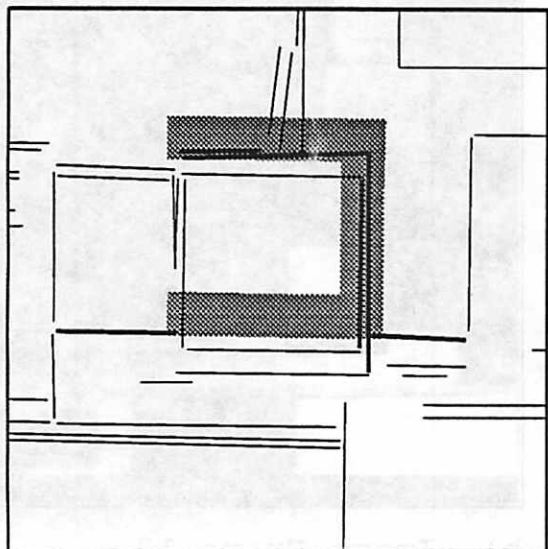


g) Frame 5

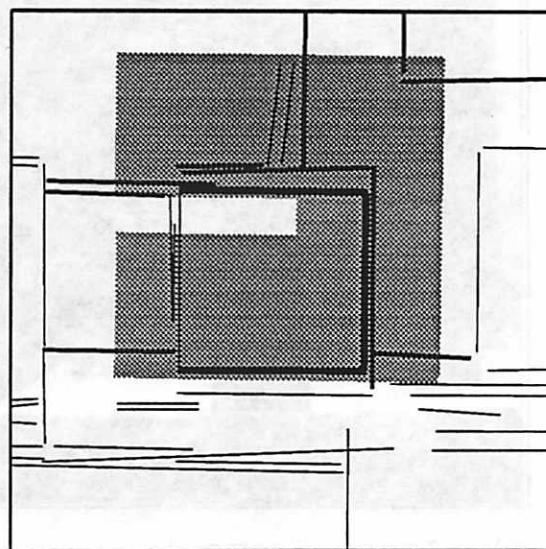


h) Frame 6

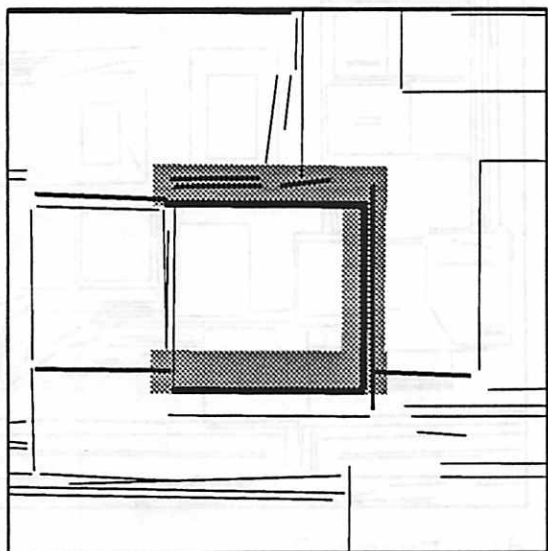
Figure 3.15: (*contd.*) Tracking of a shallow triple over ten frames in the *room-seq-1*. The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. g) No match found due to line breaking. h) Recovery from line break in frame 5. (*contd. next page*)



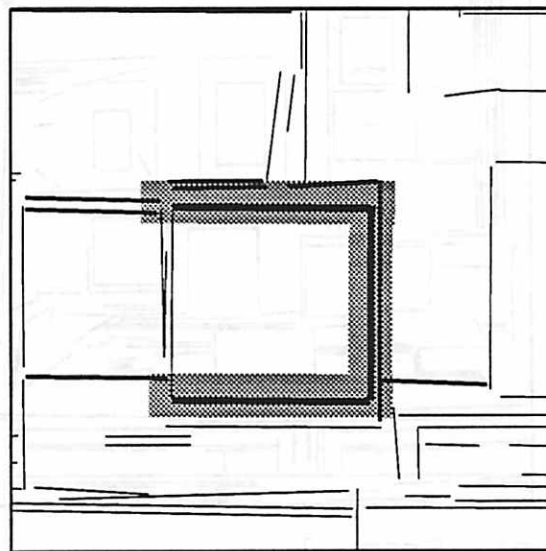
i) Frame 7



j) Frame 8

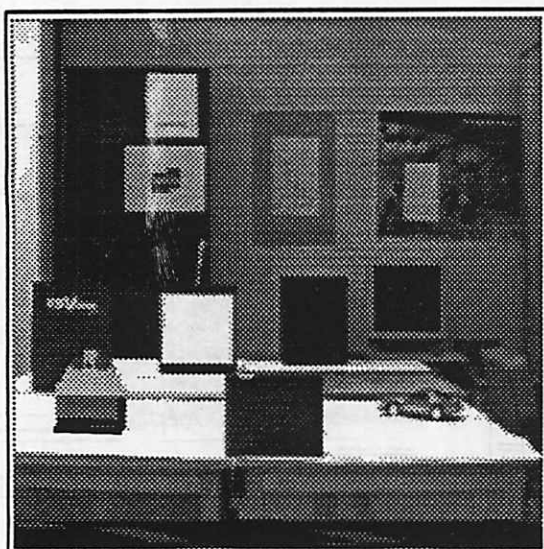


k) Frame 9

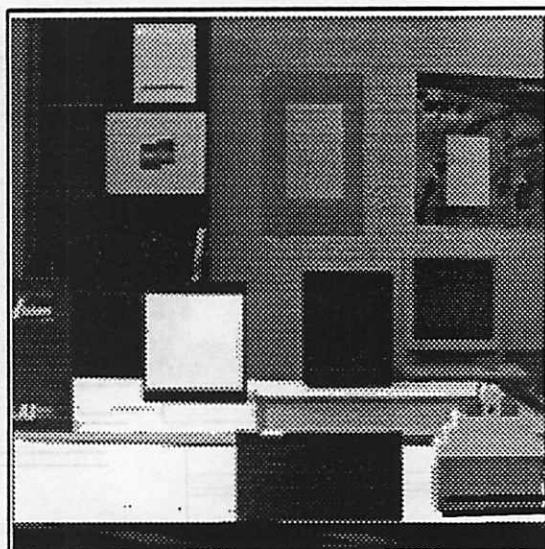


l) Frame 10

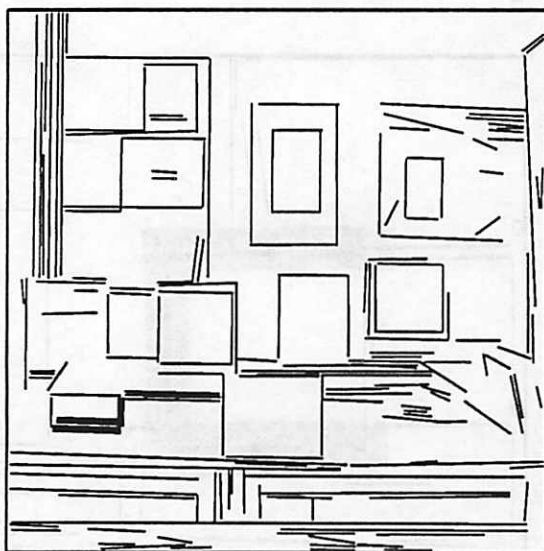
Figure 3.15: (contd.) Tracking of a shallow triple over ten frames in the *room-seq-1*. i) No match found due to motion discontinuity. j) Recovery from motion discontinuity between frames 6 and 7.



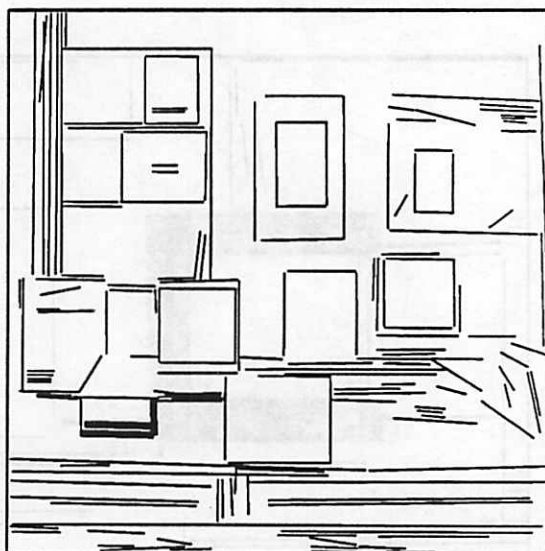
a) Image Frame 1



b) Image Frame 10

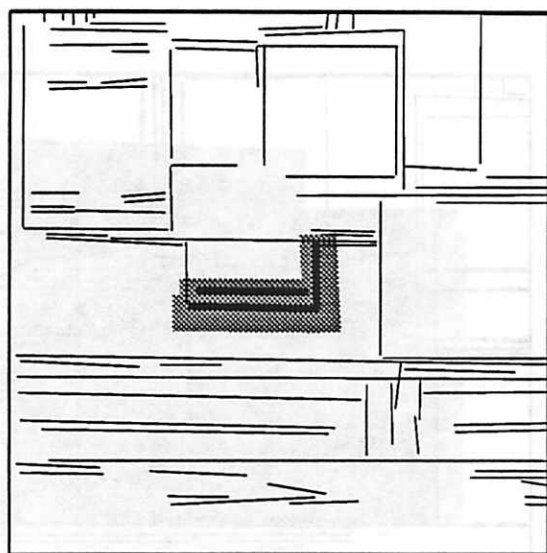


c) Frame 1

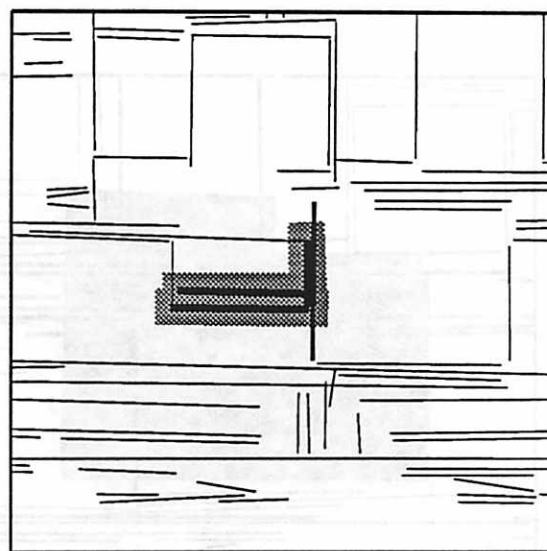


d) Frame 2

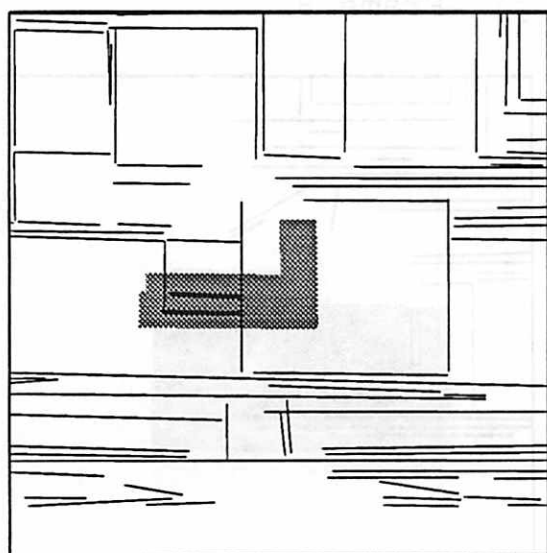
Figure 3.16: **Tracking of an independently moving object.** Tracking shown over ten frames in the *room-seq-1* with camera motion also present. a), b): First and last image frames; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction from frame 1. *(contd. next page)*



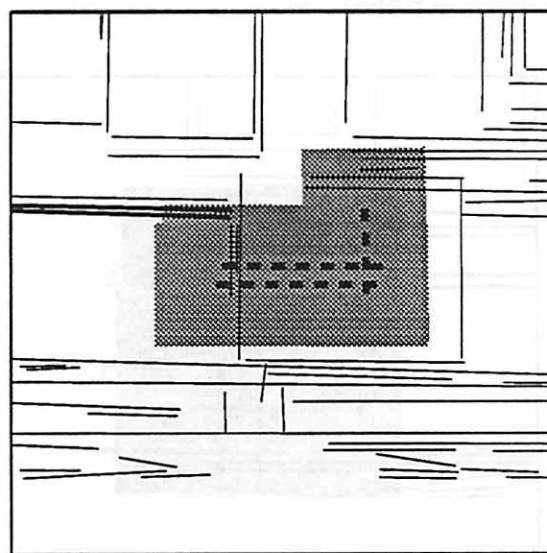
e) Frame 3



f) Frame 4

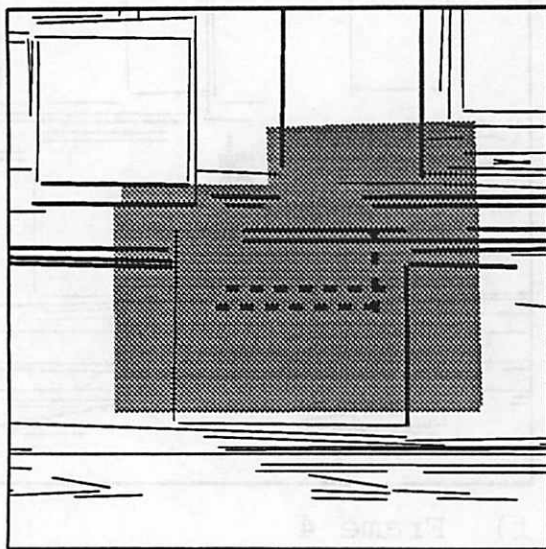


g) Frame 5

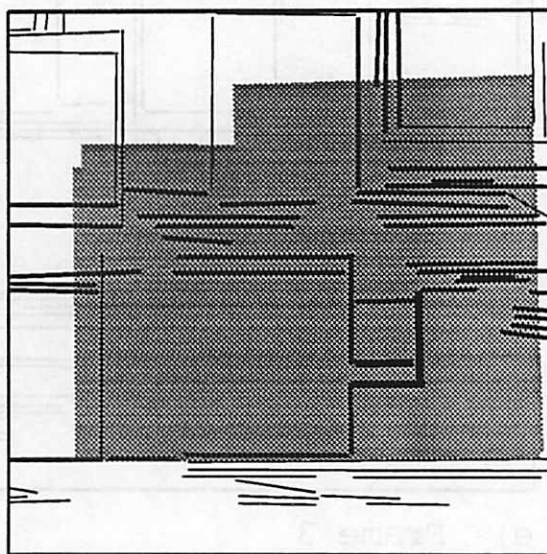


h) Frame 6

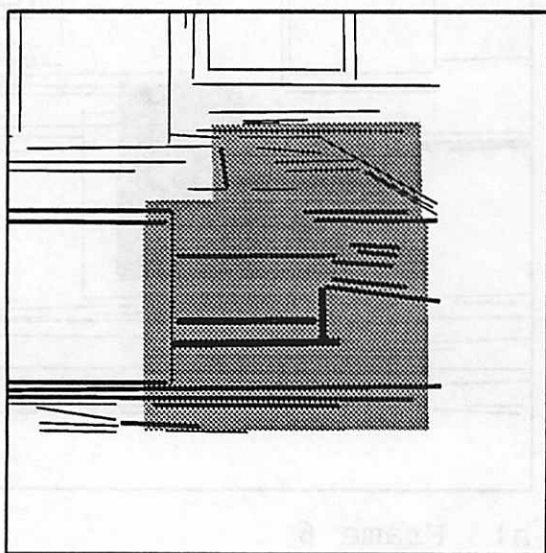
Figure 3.16: (*contd.*) **Tracking of an independently moving object.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. e), f) Matching in frames 3 and 4. g), h) No match found due to occlusion. (*contd. next page*)



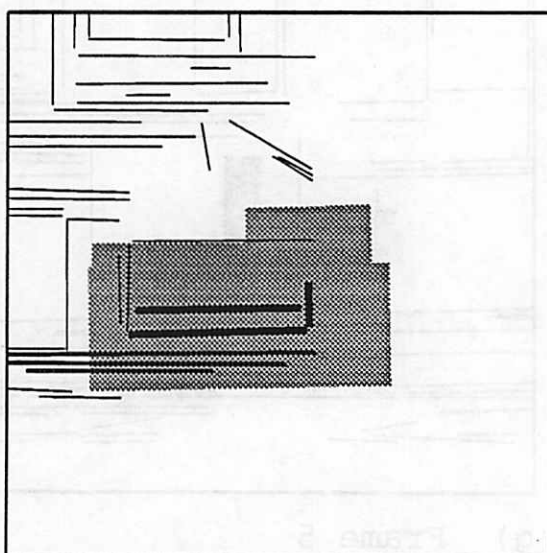
i) Frame 7



j) Frame 8



k) Frame 9



l) Frame 10

Figure 3.16: (contd.) Tracking of an independently moving object. i) No match found due to occlusion. j) Reacquisition of the object after occlusion in frames 5, 6 and 7.

of this sequence are shown in Figure 3.21. The approximate translation-in-depth between consecutive frames is 1.4 feet. The depths of some salient structures in the scene were measured from the camera in its position in frame 1. Recall that the affine transformation reconstructs a shallow structure as a fronto-parallel plane. So, for structures that have a large slant, the ground truth depths are the average depths. Figure 3.22 shows some labelled objects and Table 3.3 shows the measured and computed depths. The depths are computed over six frames of the sequence. The average absolute percentage error is 2.3%. These results suggest that when rotations are small, the fronto-parallel approximation for highly slanted shallow structures can also be computed robustly by the four-parameter affine approximation.

3.9 Summary

In this chapter, we have presented a framework for the integration of spatial constraints on generic object structure and temporal constraints on smooth motion to achieve a semantically useful description of a scene from a sequence of images. A motivation for characterizing many objects as shallow in man-made environments is presented. The motion of shallow structures in the image plane can be described by an affine transformation. Instead of clustering image features, observed over two frames, into an object hypothesis that is consistent with a shallow structure interpretation, we use the temporal evolution of a hypothesized structure to verify its consistency within the constraints of a shallow structure. Temporal evolution is characterized by the trackability of a structure under the affine constraint. Thus, a scene can be divided into shallow and non-shallow structures through the use of tracking as a verification process.

Tracking and dynamic estimation of the affine parameters of a shallow structure also lead to a reconstruction of the structure from changing scale (depth from

looming). The reconstruction of the shallow structure is as a fronto-parallel plane placed at a depth that is equal to the estimated depth. That is, the representation of shallow structures is in terms of cardboard cut-outs facing the camera for each shallow structure. An important advantage of this method is that structure reconstruction is achieved without the intermediate step of explicitly computing the 3D motion parameters (rotation and translation) between successive frames. The reconstructed structure is only an approximation, however, to the average depth of the corresponding true environmental structure. Nevertheless, the robustness of depth of the approximate structure representation might prove to be useful for tasks like obstacle avoidance, where the exact shape of an object may not be of consequence so long as collisions with it can be avoided.

We have also shown that tracking of a structure, which is formed as an aggregate of image features, is resilient to many of the common sources of errors in feature extraction and modeling of motion. Specifically, it is shown that for shallow structures, predictions of their image motion can be based on 3D constraints and not on heuristics about the image motion of features. This leads to a simple method of handling uncertainties in the modeling of 3D motion. Furthermore, matching predictions to newly acquired data of a model as a whole is more reliable than isolated feature matching.

The tracking, identification and reconstruction of shallow structures are demonstrated on real image sequences. Illustrations of how the system handles errors in feature extraction, and motion discontinuity are presented. Furthermore, it is also shown how the algorithm can track independently moving objects imaged with a moving camera. Tracking errors due to feature extraction errors, motion discontinuity and occlusions are handled in a single framework of covariance based prediction and matching.

Future directions of this research are presented in Chapter 5.



Figure 3.17: Shallow structures identified in the *cones-seq*.

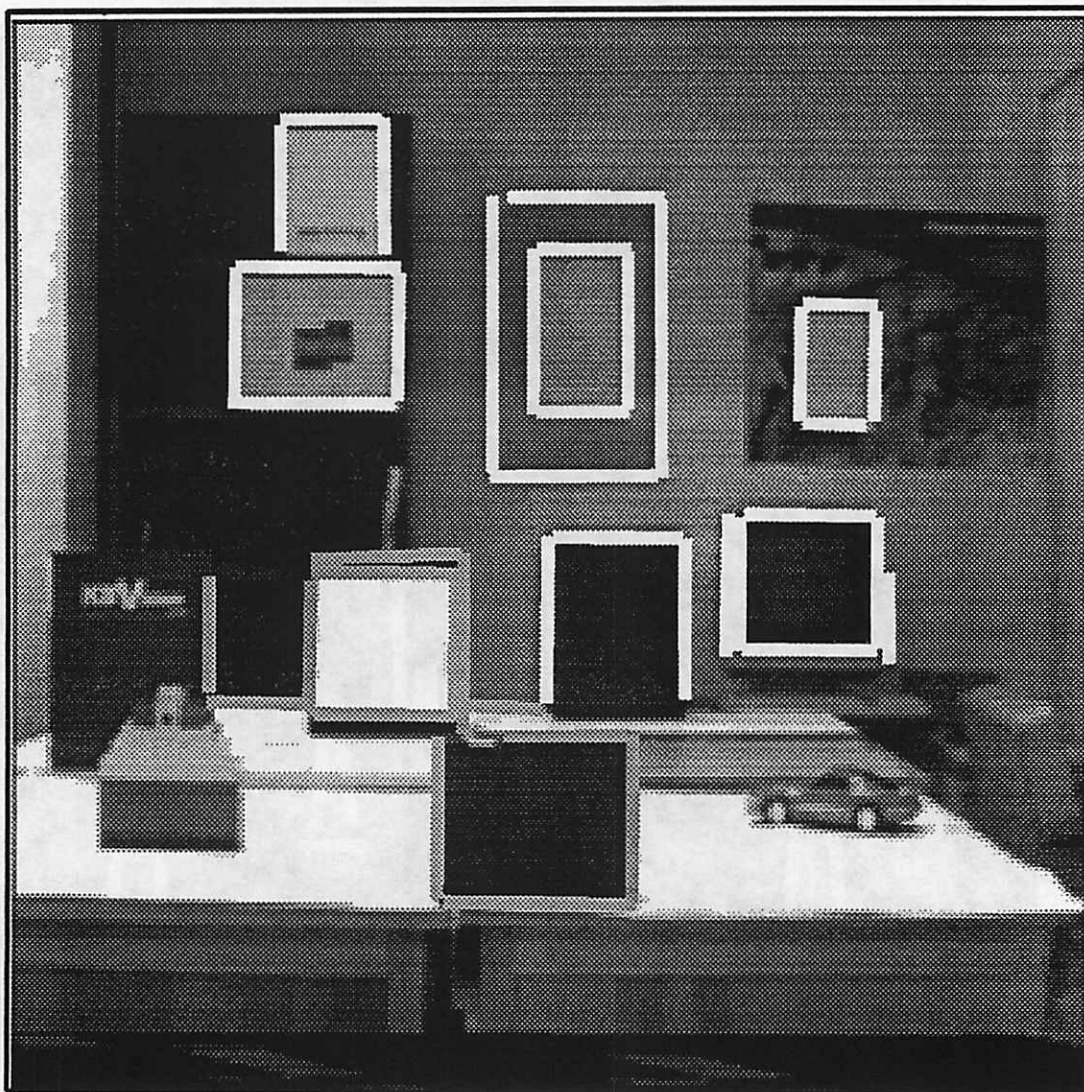


Figure 3.18: **Shallow structures identified in the *room-seq-1*.** Shown in thick white and light gray outlines.

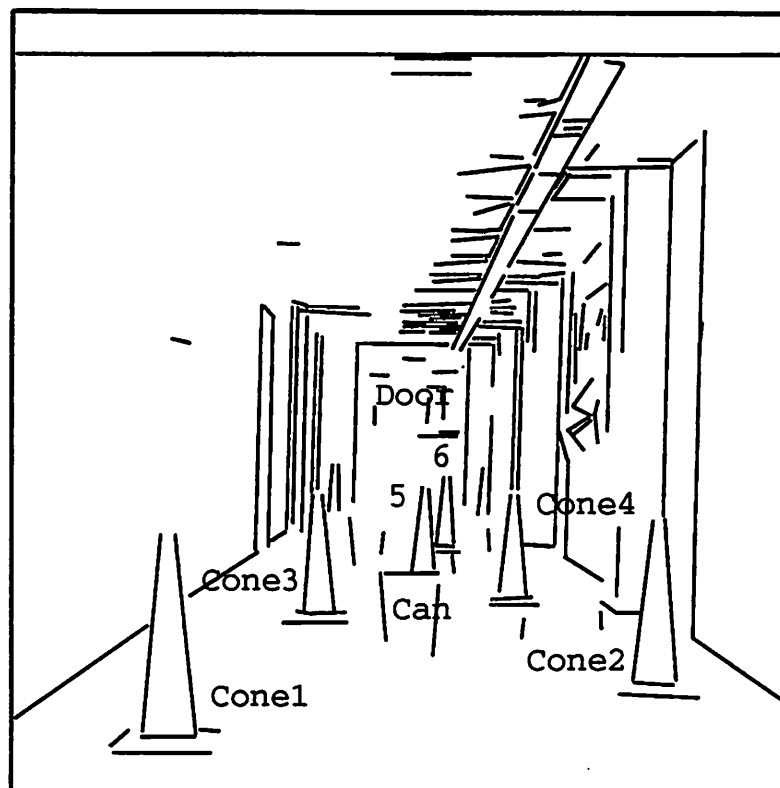


Figure 3.19: Labelled objects in the *cones-seq*. Shown in frame 1 lines.

Table 3.1: Depth results for the *cones-seq*. Computed vs. Measured Depths of some Objects in the *cones-seq* (in feet).

Object	Meas. Z	Comp. Z	Error (%)
Cone 1	20.0	20.24	1.2
Cone 2	25.0	25.91	3.6
Can	30.0	31.69	5.6
Cone 3	35.0	36.77	5.1
Cone 4	40.0	40.72	1.8
Cone 5	45.0	47.80	6.2
Cone 6	60.0	63.84	6.4
Door	87.1	87.70	0.7
Average Abs. Error			3.8%

Table 3.2: Depth results for the *room-seq-1*. Computed vs. Measured Depths of some Objects in the *room-seq-1* (in feet).

Object	Meas. Z	Comp. Z	Error (%)
1	8.3	8.02	-3.4
2	13.4	12.48	6.9
3	14.57	14.6	0.2
4	18.98	18.78	-1.1
5	11.57	11.78	1.8
6	19.04	18.01	-5.4
7	20.35	19.16	-5.8
8	20.35	19.84	-2.5
Average Abs. Error			3.4%

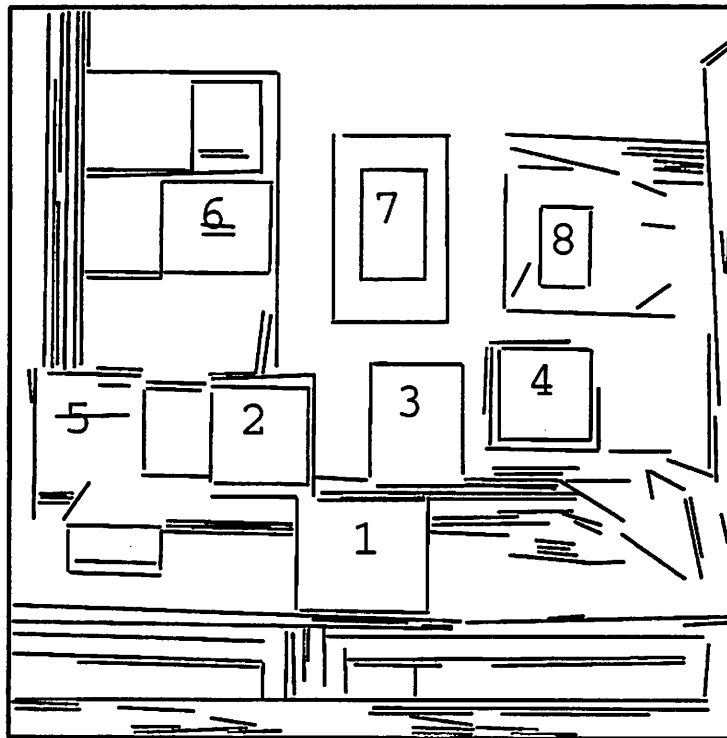
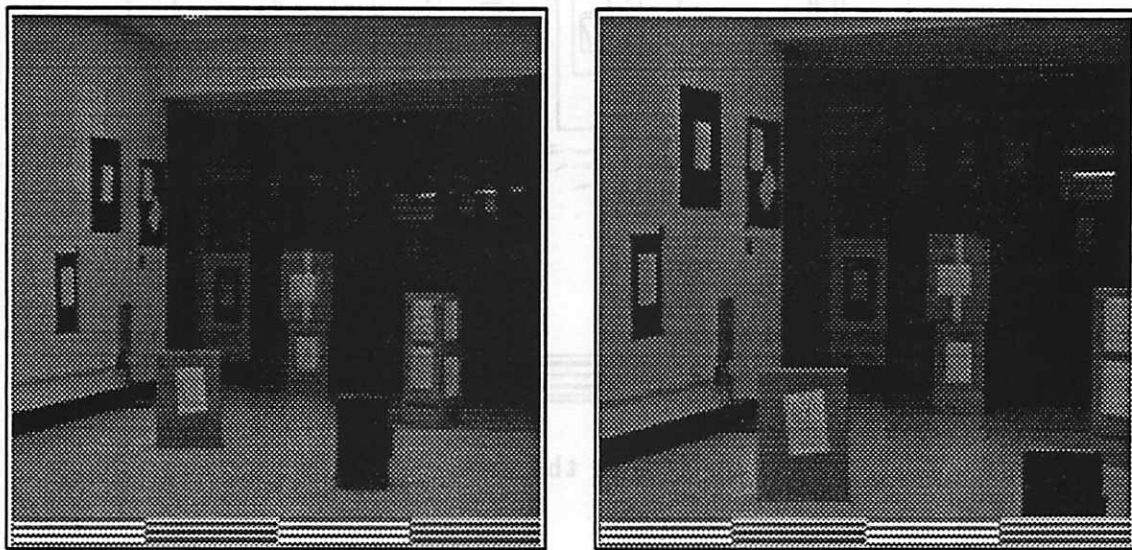


Figure 3.20: Labelled objects in the *room-seq-1*. Shown in frame 1 lines.



a) Image Frame 1

b) Image Frame 6

Figure 3.21: Two image frames of the *comp-seq*. Frames 1 and 6.

Object	Mean λ	Comp. λ	Error (%)
1	29.28	29.27	0.02
2	31.23	31.01	-0.71
3	33.22	34.22	3.19
4	35.08	33.84	-1.01
5	36.83	34.24	-4.11
6	38.18	38.24	0.16
7	41.22	44.88	8.42
8	42.22	42.08	-0.32
Average Abs. Error			2.27

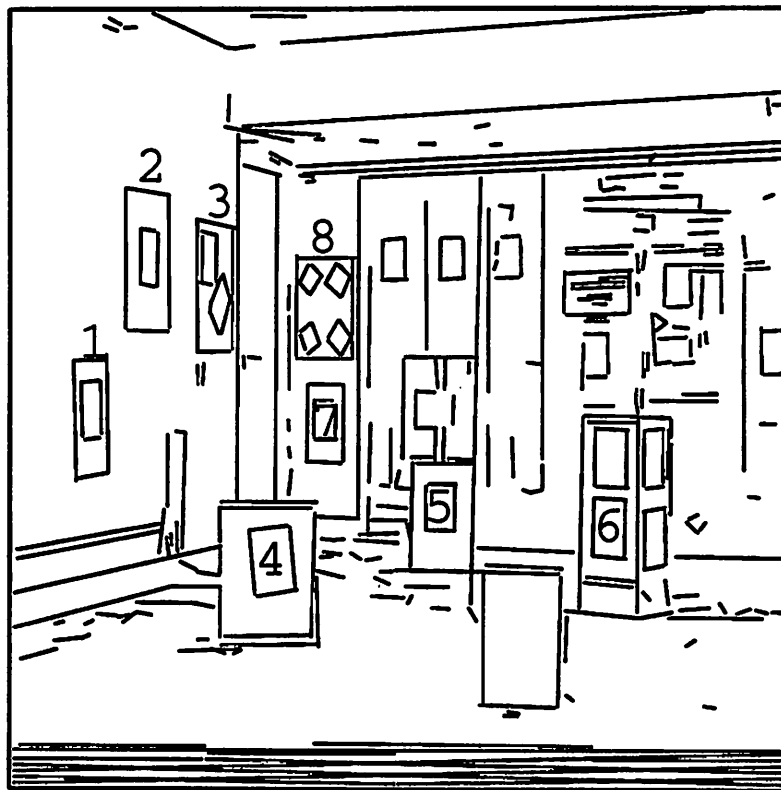


Figure 3.22: Labelled objects in the *comp-seq*. Shown in frame 1 lines.

Table 3.3: Depth results for the *comp-seq*. Computed vs. Measured Depths of some Objects (*in feet*).

Object	Meas. Z	Comp. Z	Error (%)
1	29.28	29.96	2.32
2	31.23	31.04	-0.61
3	33.23	34.37	3.13
4	25.68	25.94	1.01
5	35.83	34.24	-4.44
6	28.18	28.29	0.39
7	43.23	44.88	3.82
8	43.23	42.05	-2.73
Average Abs. Error			2.3%

CHAPTER 4

DESCRIPTION AND RECONSTRUCTION FROM IMAGE TRAJECTORIES OF ROTATIONAL MOTION

In many applications, it is desirable to automatically build internal models of the environment and/or objects by moving a camera in a constrained motion. One such motion is rotational motion that can be carried out even without the availability of a large work space. For instance, in industrial settings, a cartesian arm can pick up and rotate objects around some arbitrary axis. Internal models of these objects could be built by capturing a sequence of images of this motion from a fixed camera. Alternatively, a robotic arm, which holds a camera, could be rotated to build models of otherwise completely unmodelled environments.

In this chapter, we present a new technique for reliably computing 3D structure from a sequence of images of a scene undergoing a relative rigid-body rotation with respect to the camera. We do not assume knowledge of the parameters of motion but only that the motion is rotational around an arbitrary axis in space.

First, a closed-form solution has been developed for the 3D motion and structure parameters of a point given its image trajectory from rotational motion under *perspective projection*. For this technique to work on real images, it is necessary that image trajectories of points tracked over many frames be described reliably. We show that even when 80–100 degrees of an arc of a 3D trajectory is imaged, its description as a curve in the image plane is very unreliable for computing the 3D parameters. Consequently, a new grouping algorithm is developed which exploits common constraints across many trajectories to obtain robust combined fits to a group of these.

The trajectories thus obtained lead to a dramatic improvement in the reliability of the derived 3D parameters and hence to reliable 3D reconstruction. Application of both the grouping algorithm and the closed-form solution for 3D reconstruction to real image sequences is demonstrated.

4.1 Previous Work

It is well-known that 3D structure can be derived from images of a scene undergoing relative rotational motion with respect to the camera if the axis of rotation does not pass through the origin of the camera coordinate system. Algorithms for this problem of 3D interpretation from monocular motion can be broadly divided into two categories — two-frame and multi-frame. Two-frame algorithms first compute the relative orientation — the translation and rotation — between the camera positions at two time instants [1, 33, 43, 90]. Then the relative orientation is used to compute the 3D location for each imaged feature. In addition to advantages and disadvantages specific to instances of these algorithms, the two-frame methods suffer from two major problems. First, for some motions, there are inherent ambiguities in the computation of relative orientation from noisy image correspondences [3, 92]. Rotation and translation parallel to the image plane is one such case. Second, with just two frames of imaged features, the structure computation for each is based only on a single measured displacement vector. Thus, even a small amount of noise in the measurement can make the depth estimate quite inaccurate. Moreover, when both rotation and translation parallel to the image plane are present, the motion estimates are biased due to the inherent ambiguities; consequently, it is unlikely that use of multiple frames will improve the depth estimates. These aspects of two-frame reconstruction, and multi-frame reconstruction based on the two-frame motion computation, were discussed at length in Chapter 2.

Furthermore, two-frame methods inherently do not describe motion in its natural frame of reference. For instance, a pure rotation around an axis not through the camera origin can be described only as a rotation around a parallel axis through the origin and a translation. In principle, the more natural pure rotational description is derivable from many two-frame computations. However, given that these estimates can be biased, it is unlikely that the natural description thus derived will be robust.

In the following discussion, only those multi-frame reconstruction methods which use models of motion are considered, due to the relevance of these methods to the approach developed in this chapter. A broader review of two-frame and multi-frame reconstruction techniques is presented in Chapter 2.

Weng et al. [93] use a model of 3D motion that describes precession (rotation around an axis that itself rotates around a fixed axis, for instance, the motion of a spinning top). However, they fit their model to rotations and translations derived from many two-frame computations. In other words, for each pair of image frames, a 3D relative orientation is computed and then a number of these are reconciled using the model. Thus they potentially suffer from the instabilities of the underlying two-frame estimates. Their results are presented only with 2D motion data.

Shariat [81] employs a model of constant rotation and translation with uniform sampling of the image frames. His method uses only a specific number (minimal) of points and frames and is not easily extensible to arbitrary amounts of data. Broida [19] generalizes the motion models to a Taylor series expansion of the translational and rotational components. The sampling of frames can be non-uniform. He minimizes the error between the expected and measured positions of imaged features. The 3D motion and structure parameters for each feature are solved for in a single optimization stage. Convergence is slow, and in general, the multi-modal objective function can be sensitive to initial guesses. Results presented for real image sequences of uniform motion match well with the ground truth.

Webb and Aggarwal [91] solve for parameters of rotational motion using elliptical descriptions of image trajectories. However, their results are valid only for *orthographic projection*. Further, their ellipse fitting algorithm does not exploit any common constraints across *more* than two trajectories to derive robust fits. Thus, their 3D estimation fails quite badly with noisy data, as they themselves report. Also, orthographic projection cannot give 3D structure when the motion is parallel to the image plane. In this case, rotation around an arbitrary axis parallel to the optical axis can be decomposed into a rotation around the optical axis and a translation perpendicular to it. Neither of these motions provide any structure information under orthographic projection.

Jaenicke [45] applies Webb's method to radar doppler images. Under orthography, three of the five parameters of each elliptical trajectory are constrained to be the same for all trajectories. He derives the common parameters of each trajectory by averaging those for the independent trajectories. The averaging for each parameter is done in succession. That is, after computing an averaged parameter, the fitting process is repeated and the averaging for the next parameter is done on the newly derived set. This works only if the individual trajectories themselves are slight deviations of the correct ones, which is the case in his examples. In contrast, our trajectory grouping does not rely on the goodness of the individual fit parameters, as in general these will be far from the correct ones, which is the case in all our examples. Moreover, we exploit both the spatial and temporal extents of the input point correspondences. If one point is tracked for a fewer number of frames and hence a smaller extent of the 3D arc is available, its individual trajectory may be very erroneous. But by fitting trajectories to groups of point correspondences simultaneously (combined fitting), the trajectory of each point may be strongly constrained by another spatially proximal one even though they lie on distinct trajectories. Consequently, the spatio-temporal

extents of short and noisy point tracks mutually constrain their trajectories leading to robust descriptions.

In the context of model-based object recognition and pose estimation, both the problems of fitting ellipses to the image data, and that of reconstructing a 3D circular feature on a plane given its corresponding elliptical projection in the image, have been addressed [35, 60]. Marimont [60] presents a closed-form solution to the reconstruction problem. However, the 3D solution presented here is much simpler due to the particular parameterization chosen for the problem. Forsyth et al. [35] use projectively invariant measures of conic sections for fitting ellipses and for object recognition. However, they assume that most of the elliptical curve is present in the image data. It is not clear how their method will perform when objects are occluded or when only small fractions of the 3D trajectory are imaged.

4.2 Overview of the Approach

The 3D interpretation problem is divided into two distinct stages. It is shown that under perspective projection, the image trajectory of a circular rotational trajectory in 3D space is a general conic section. The first stage, the trajectory description stage (called **TRAJ-DESC**), takes as its input discrete point correspondences tracked over time for many points. For each point, the set of temporal correspondences is called a *Point Track*. A set of point tracks is grouped into a set of *Point Trajectories* based on the goodness of a combined fit error measure. The output of this stage is a conic curve describing each of the grouped point tracks. The second stage, the 3D estimation stage (called **3D-EST**), uses the closed-form solution developed in this work and applies it to the image point trajectories to output the 3D motion and structure parameters. The different stages of input, processing and output are schematically depicted in Figure 4.1.

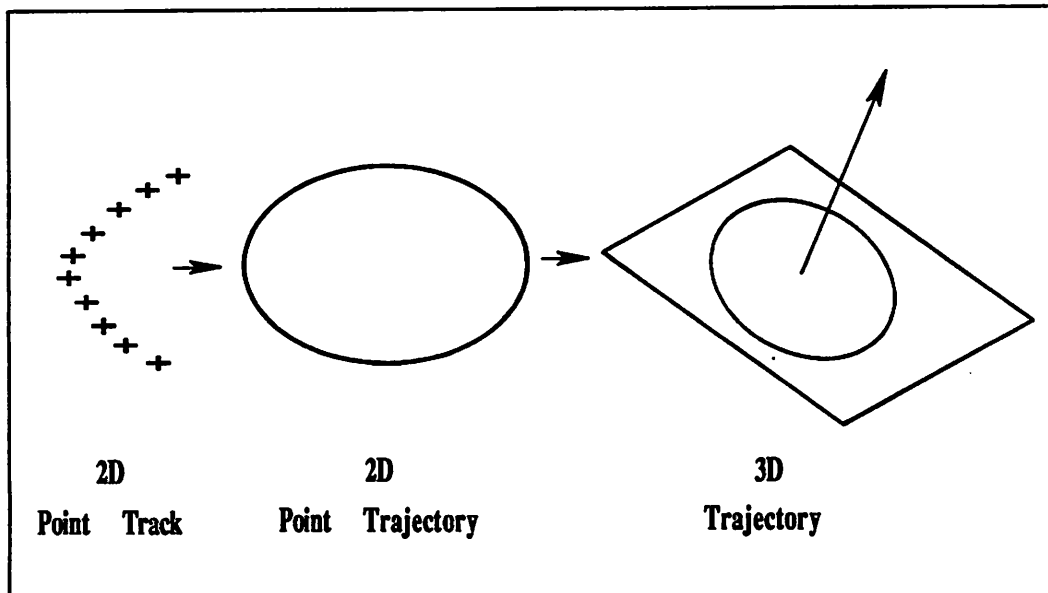


Figure 4.1: The stages of processing of rotational trajectories.

There are two distinct advantages in dividing up the problem into these two steps. First, a number of image frames for a single point are utilized in its trajectory description. This alone should lead to improvement in 3D depth estimates as opposed to methods which use only a small number of frames. This improvement is achieved through implicit averaging of random noise in the image features by fitting a continuous best-fit curve to the discrete correspondences. Second, in contrast with other multi-frame methods, the optimization problem is handled at the trajectory grouping stage. Experimental results show that the resulting error surface at this stage has a larger basin of convergence to the correct solution. That is, the common grouping constraint is strong enough that the initial guesses generated automatically by the incremental grouping algorithm seem to avoid the wrong local minima and largely converge to the correct solution.

This approach differs from most 3D structure-from-motion algorithms in its emphasis on an explicit stage of describing image motion over many frames by means of trajectories, prior to the computation of 3D motion and structure. It is our position that the local information provided by optical flow or displacement fields must be translated into extended-time descriptions of the motion of large-scale spatial structures in order to achieve reliable 3D reconstruction. Furthermore, the extended-time descriptions can also serve as a representation for perceptual organization in dynamic images to aid in the detection of occlusions and multiple object motions [88, 91].

The approach taken here has compelling parallels with Stevens' [84] idea of capturing global geometric organizations in Glass patterns through grouping. The point tracks used as input to the algorithm (Figures 4.19 and 4.21) are like temporally generated spatial Glass patterns. The grouping process exploits both their spatial and temporal aspects. Their common fate, in terms of the same 3D motion, is made explicit through the grouped trajectory descriptions.

The next section is devoted to the development of a closed-form solution to the 3D reconstruction problem (algorithm 3D-EST). Section 4.4 presents the trajectory description algorithm (TRAJ-DESC) and section 4.5 describes experimental results on some real image sequences and their comparison with other algorithms.

4.3 The 3D Estimation Problem

This section presents a solution to the 3D reconstruction of rotational motion. Reliable image trajectories (e.g. Figures 4.22 and 4.23) expected as input to this solution technique are obtained through the grouping algorithm described in the next section. Its discussion is deferred because some results developed in this section will aid in the understanding of the grouping constraints. It is assumed that the intrinsic parameters of the camera are known and for simplicity, that the image is square.

Given the conic trajectory that describes the motion of a point in the image, the problem here is to determine the orientation and location of the rotation axis and the location and radius of the corresponding 3D trajectory.

4.3.1 Formulation

An outline of the approach to the 3D estimation problem follows:

- It is shown that the perspective projection of an arbitrary circular trajectory in 3D is a general conic section.
- Given this conic projection, a closed-form solution for the 3D parameters is derived using the eigenvectors of the symmetric matrix representing the conic. An apparent eight-fold ambiguity is shown in the solution.
- Six of the eight possible solutions are rejected by invoking the scene-in-front-of-image constraint or because they duplicate the other solutions.
- The remaining two-fold ambiguity is resolved by constraining different 3D trajectories to share the same axis of rotation.

The set of parameters defining the problem geometry as depicted in Figure 4.2 is:

\hat{b} : Unit vector along the rotation axis.

\vec{c} : Location vector of the rotation axis, given by the point where the axis intersects a plane normal to it that passes through the origin.

\vec{r} : (x, y, z) , Location vector of a 3D point.

d : Location of the center of the circular 3D trajectory, given by its signed distance from the point \vec{c} and measured positive in the $+z$ - direction.

k : Radius of the circular 3D trajectory, $k > 0$.

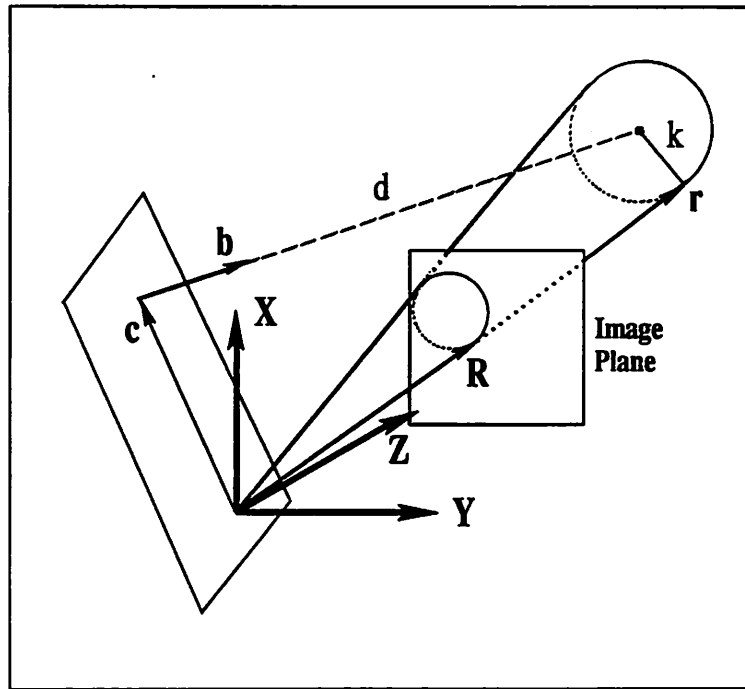


Figure 4.2: Geometry of a point's rotation.

f : Focal length of the camera.

\vec{R} : (X, Y, f) , image vector in homogeneous pixel coordinates.

This parameterization separates the motion and structure parameters.

A vector is represented as \vec{v} , a unit vector as \hat{v} and the corresponding column vector as v . Quantities enclosed in square brackets, e.g. $[M]$, represent matrices.

From Figure 4.2, it is evident that the motion geometry is constrained by

$$((\vec{r}_i - \vec{c}) - d_i \hat{b}) \bullet ((\vec{r}_i - \vec{c}) - d_i \hat{b}) = k_i^2 \quad (4.1)$$

$$\vec{r}_i \bullet \hat{b} = d_i \quad (4.2)$$

$$\vec{c} \bullet \hat{b} = 0 \quad \hat{b} \bullet \hat{b} = 1 \quad (4.3)$$

The index i refers to a 3D point in the scene. For notational convenience, it is dropped in the following treatment.

The perspective equation in homogeneous coordinates is:

$$\vec{R} = f \frac{\vec{r}}{\vec{r} \bullet \hat{z}} \quad (4.4)$$

From Equation 4.2, using similar triangles:

$$\vec{r} = \frac{d}{\vec{R} \bullet \hat{b}} \vec{R} \quad (4.5)$$

(The degenerate case where $d = \vec{r} \bullet \hat{b} = 0$ causes no difficulty and is discussed later.)

Substituting Equation (4.5) into Equation (4.1) and rearranging terms, we get

$$\frac{d^2}{(\vec{R} \bullet \hat{b})^2} \vec{R} \bullet \vec{R} - 2d \frac{\vec{R} \bullet \vec{c}}{\vec{R} \bullet \hat{b}} + \vec{c} \bullet \vec{c} - d^2 - k^2 = 0 \quad (4.6)$$

After multiplication by $(\vec{R} \bullet \hat{b})^2$, this can be expressed as the following quadratic form:

$$\mathbf{R}^T [d^2 [I] - d [\mathbf{c} \mathbf{b}^T + \mathbf{b} \mathbf{c}^T] + (\mathbf{c}^T \mathbf{c} - d^2 - k^2) [\mathbf{b} \mathbf{b}^T]] \mathbf{R} = 0 \quad (4.7)$$

This equation represents a general conic in X and Y , the image plane coordinates. Thus, given the circular 3D trajectory of a point, the *expected* image projection is determined by the following matrix:

$$[M_{exp}] = [d^2[I] - d[cb^T + bc^T] + (c^T c - d^2 - k^2)[bb^T]] \quad (4.8)$$

The image trajectory is an ellipse if the full 3D trajectory lies on the positive z side of the xy -plane's half-space (Figure 4.2). It is a hyperbolic arc when the 3D circle intersects the xy -plane in exactly two points. The four possible directions leading to these intersections determine the four asymptotes in the image plane. Finally, when the 3D circle is tangent to the xy -plane, the imaged arc is a section of a parabola. Again, the two possible directions of approach towards the tangent point generate the two unbounded paths in the image. In the latter two cases, the image trajectory is not closed. In the case of an ellipse, the trajectory may be either a closed curve — a complete ellipse — or an open partial ellipse.

In order to obtain a solution, a minimum number of points and frames are required. The rotation axis can be specified using a minimum of four parameters [73]. Two additional parameters (d and k) specify each 3D point relative to the axis. Thus, for n 3D points, there are $2n+4$ unknowns, of which only $2n+3$ can be determined because of the ambiguity in scale discussed in the next subsection. Each image point in each frame gives one constraint equation (Equation 4.7). Therefore, one 3D point imaged in five frames, two 3D points imaged in four frames, or more than two 3D points all imaged in more than two frames, provide adequate constraints for a solution¹. In practice, in order to obtain a robust solution in the presence of noise, more data is necessary.

¹Note that this is different from Shariat's [81] calculations because we do not assume constant rotational speed or uniform sampling.

Having shown that the image trajectory of a 3D point in rotational motion is a conic section, we now show how its 3D parameters can be derived from the parameters of its image trajectory.

4.3.2 Solution

Let $[M_{com}]$ be the symmetric 3-by-3 matrix representing a conic trajectory in the image plane. $[M_{com}]$ is computed using the trajectory grouping algorithm described in Section 4.4.2. The corresponding expected trajectory in terms of the 3D parameters d , k , \vec{c} and \hat{b} is represented by the matrix $[M_{exp}]$ of Equation (4.8). Any scalar multiple of $[M_{exp}]$ represents the same image curve, so that the 3D parameters can only be recovered up to a scale factor. (Note, however, that the data matrix $[M_{com}]$ has an intrinsic scale factor related to the units of X and Y per unit focal length. This factor is recoverable but does not affect the solution for the 3D parameters, hence is ignored in the following treatment). Thus, only the ratios d_n , k_n , c_n and \hat{b} are recoverable, where

$$d_n = \frac{d}{|\vec{c}|} \quad k_n = \frac{k}{|\vec{c}|} \quad \vec{c}_n = \frac{\vec{c}}{|\vec{c}|} \quad \mathbf{c}_n^T \mathbf{c}_n = 1 \quad (4.9)$$

This is the well-known ambiguity of scale in 3D reconstruction from monocular motion although it appears here in a different form. It is assumed here that the rotation axis does not pass through the origin, i.e. that \vec{c} is not the zero vector. The case when \vec{c} is zero can be characterized from the image data, and will be treated separately. $[M_{exp}]$ is written in terms of the scaled parameters as:

$$[M_{exp}] = [d_n^2[I] - d_n[\mathbf{c}_n \mathbf{b}^T + \mathbf{b} \mathbf{c}_n^T] + (1 - d_n^2 - k_n^2)[\mathbf{b} \mathbf{b}^T]] \quad (4.10)$$

Now the 3D variables in $[M_{exp}]$ are derived from the computed data matrix $[M_{com}]$.

Consider the generic case in which the axis does not pass through the origin. Dropping the subscript for the normalized parameters, $[M_{exp}]$ is rewritten as:

$$[M_{exp}] = [d^2[I] - d[\mathbf{c} \mathbf{b}^T + \mathbf{b} \mathbf{c}^T] + (1 - d^2 - k^2)[\mathbf{b} \mathbf{b}^T]] \quad (4.11)$$

First the eigenvalues of $[M_{exp}]$ are derived. It is a standard result in linear algebra that the eigenvalues of a matrix are invariant to rotations of the coordinate system. Since b and c are orthogonal, these can be transformed to $[0\ 0\ 1]$ and $[0\ 1\ 0]$, respectively, in a rotated coordinate system. $[M_{exp}]$ of Equation (4.11) has the following simple form in this rotated system,

$$[M'_{exp}] = \begin{bmatrix} d^2 & 0 & 0 \\ 0 & d^2 & -d \\ 0 & -d & 1 - k^2 \end{bmatrix}$$

but its eigenvalues remain the same as the original matrix.

The three eigenvalues of this matrix are:

$$\lambda_1 = d^2 \quad (4.12)$$

$$\lambda_2 = \frac{1}{2} \left((1 + d^2 - k^2) + \sqrt{(1 + d^2 - k^2)^2 + 4d^2 k^2} \right) \quad (4.13)$$

$$\lambda_3 = \frac{1}{2} \left((1 + d^2 - k^2) - \sqrt{(1 + d^2 - k^2)^2 + 4d^2 k^2} \right) \quad (4.14)$$

It is evident that λ_1 and λ_2 have the same sign. The sign of λ_3 is different, except in the degenerate case when d is zero for an image trajectory that is a degenerate conic, i.e. a straight line segment. By suitably normalizing the computed matrix $[M_{com}]$, one of its eigenvalues can be made negative and the other two positive. Hence the negative eigenvalue can be uniquely identified with λ_3 . Also, the larger of the two positive eigenvalues can be uniquely identified with λ_2 , which one can show is always larger than λ_1 except possibly in the degenerate case.

The three eigenvalues of $[M_{com}]$ can therefore be assigned unambiguously to λ_1 , λ_2 and λ_3 , corresponding to Equations (4.12), (4.13) and (4.14), respectively. The 3D parameters, d and k , can be solved for in terms of these eigenvalues. Let $\gamma_1 \equiv \lambda_2/\lambda_1$ and $\gamma_2 \equiv \lambda_3/\lambda_1$. Then,

$$d^2 = \frac{1}{\gamma_1 + \gamma_2 - \gamma_1\gamma_2 - 1} \quad k^2 = -\gamma_1\gamma_2 d^2 \quad (4.15)$$

Thus d and k are determined up to a sign ambiguity in d . Sign ambiguities of the solution are discussed in the next section.

Now, b and c are determined from the eigenvectors of $[M_{com}]$. It is evident from Equation (4.11) that one of the eigenvectors of $[M_{exp}]$ is a vector n_1 normal to the plane formed by b and c . Since n_1 satisfies,

$$[M_{exp}]n_1 = d^2 n_1 = \lambda_1 n_1 \quad (4.16)$$

it is associated with the eigenvalue λ_1 . The other two eigenvectors n_2 and n_3 , with the associated eigenvalues λ_2 and λ_3 , respectively, must span the plane formed by b and c , since all the eigenvectors are mutually orthogonal. Therefore,

$$\hat{c} = \cos \theta \hat{n}_2 + \sin \theta \hat{n}_3 \quad \hat{b} = \sin \theta \hat{n}_2 - \cos \theta \hat{n}_3 \quad (4.17)$$

for some θ . Further,

$$[M_{exp}]c = d^2 c - d b = \lambda_2 \cos \theta n_2 + \lambda_3 \sin \theta n_3 \quad (4.18)$$

$$[M_{exp}]b = -d c + (1 - k^2) b = \lambda_2 \sin \theta n_2 - \lambda_3 \cos \theta n_3 \quad (4.19)$$

Substituting Equation (4.17) into (4.18) and (4.19), we obtain,

$$\tan \theta = \frac{d^2 - \lambda_2}{d} = \frac{d}{\lambda_3 - d^2} = \frac{d}{(1 - k^2) - \lambda_2} = \frac{\lambda_3 - (1 - k^2)}{d} \quad (4.20)$$

which are all equivalent expressions. Thus, $\tan \theta$ can be computed in closed form in terms of the image parameters up to the sign ambiguity in d . It follows that b and c can also be obtained in closed form up to sign ambiguities, by solving for the eigenvectors n_2 and n_3 of the image conic matrix, which are identified unambiguously by their respective eigenvalues.

Hence, apart from the sign ambiguities, all the 3D parameters of a trajectory can be uniquely computed in terms of the eigenvalues and eigenvectors of $[M_{com}]$, the matrix computed from the image trajectory.

The case of the axis passing through the origin ($|\vec{c}| = 0$) can be similarly analyzed ([78] and Appendix A). It can be distinguished from the generic case because two of the eigenvalues will be identical. Pairing the expected and the computed eigenvalues uniquely as before, the axis direction can be computed as the eigenvector corresponding to the distinct eigenvalue. Note that only the ratio k/d is computable in this case.

4.3.3 Multiple Solutions

There are two solutions for d in Equation (4.15). For each of these, there are four solutions for b and c from the four sets of signed values of n_2 and n_3 in Equation (4.17). The four solutions corresponding to one sign of d are depicted pictorially in Figure 4.3 and those corresponding to the opposite sign are shown in Figure 4.4. For each sign of d , the four solutions are obtained by reflecting b and c across each of n_2 and n_3 . The *eight* solutions can be grouped into two sets of four solutions each. The four solutions within each set differ only in the signs of b , c and d . For each solution in one set, there is a corresponding solution in the other which is distinct from the first in the sense that it cannot be obtained from the former by a simple sign reversal. The two sets, each corresponding to the same k , can be written as:

$$S_1 = \{ \{b_1, c_1, d_1\}, \{-b_1, -c_1, d_1\}, \{b_1, -c_1, -d_1\}, \{-b_1, c_1, -d_1\} \}$$

$$S_2 = \{ \{b_2, c_2, d_1\}, \{-b_2, -c_2, d_1\}, \{b_2, -c_2, -d_1\}, \{-b_2, c_2, -d_1\} \}$$

It is clear from Equation (4.11) that the different signed values for the parameters within each set lead to the same computed matrix. To see the relation between the corresponding solutions in S_1 and S_2 , refer to Figure 4.5. For each solution in S_1 , the corresponding solution in S_2 is obtained by reflecting each of $\{b_i, c_i\}$ in S_1 across any one of the eigenvectors, n_2 or n_3 . Figure 4.5 depicts one of these reflections

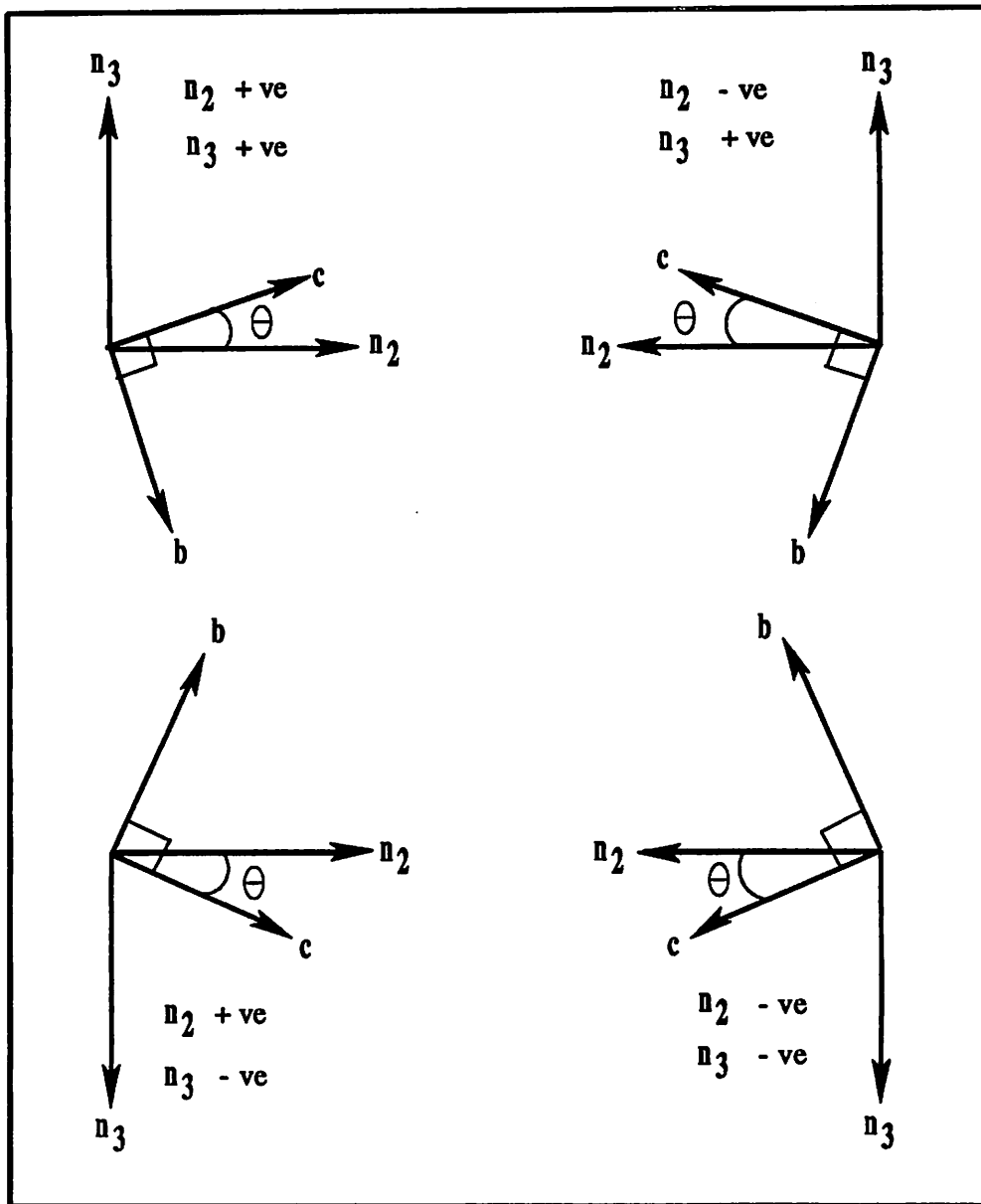


Figure 4.3: Four solutions corresponding to $+d$. $+ve$ and $-ve$ denote positive and negative, respectively.

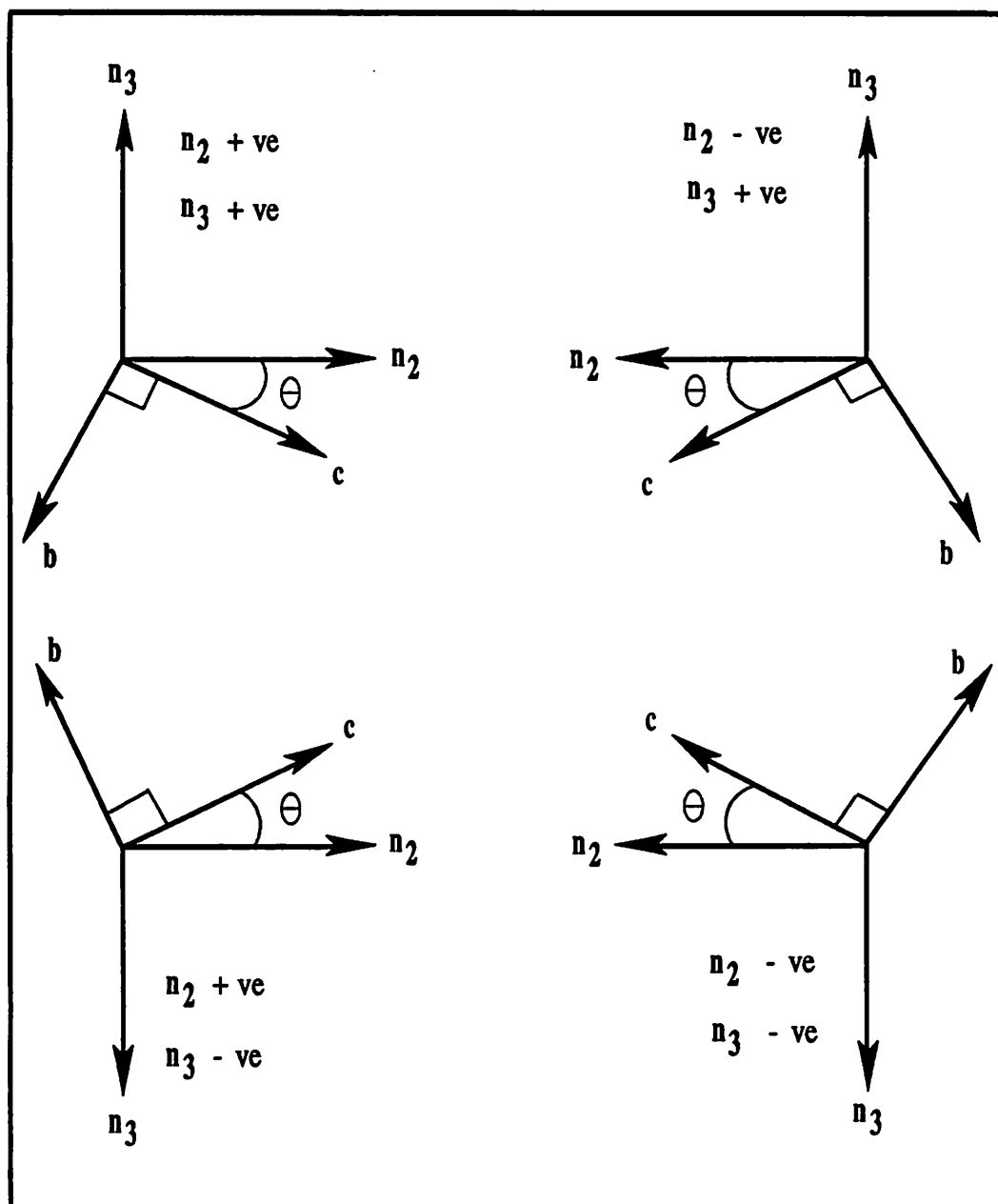


Figure 4.4: Four solutions corresponding to $-d$. + ve and - ve denote positive and negative, respectively.

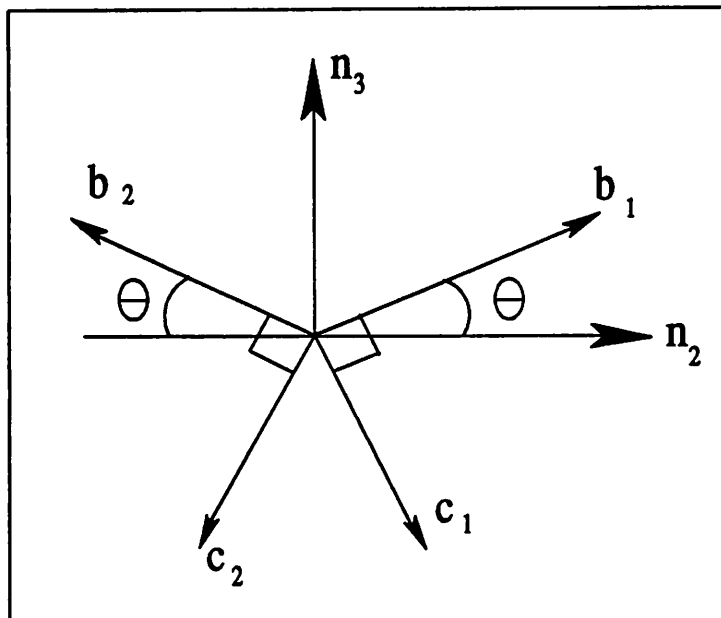


Figure 4.5: Two distinct solutions for one point.

across n_3^2 . The result of this reflection again yields a matrix with the same sets of eigenvalues and eigenvectors. Hence, given the measured image conic matrix, these eight indeed are the solutions. Within the constraints of Equation (4.3), these are the only possible solutions because they exhaust all options in the representation of the given matrix in terms of its eigencomponents, which in turn is a complete representation.

The apparent eight-fold ambiguity found above is not real. Since $\{b, c, d\}$ and $\{-b, c, -d\}$ represent the same point along the rotation axis, four of the above solutions simply duplicate the other four. This ambiguity is eliminated by always choosing the z -component of b to be positive. This leaves two solutions in each set. We next impose the constraint that a 3D point must lie in front of the image plane in order to

²Incidentally, Shariat's [81] claim that the dual axis direction is a reflection of the first solution across the line joining the focal point and the center of the 3D trajectory is incorrect because this line is clearly not an eigenvector of $[M_{exp}]$ of Equation (4.11).

be imaged [43]. From Equation (4.5), this implies that $\beta = \frac{d}{R\bar{b}}$ must be positive. If the parameter set $\{b, c, d\}$ satisfies this constraint, then the alternate set $\{-b, c, d\}$ cannot. Thus, two more solutions are eliminated, one from each set. The remaining two solutions, one from each set, cannot be disambiguated from the image trajectory of one point alone.

However, image trajectories of many 3D points rotating rigidly around a common axis can be used to resolve this ambiguity. The true solution for the axis will be common to *all* the points, while the other, incorrect solution will be unique for each 3D point. The true solution is therefore easily picked out. The mismatch among the incorrect solutions is illustrated in Figure 4.6 for two points. The figure shows two

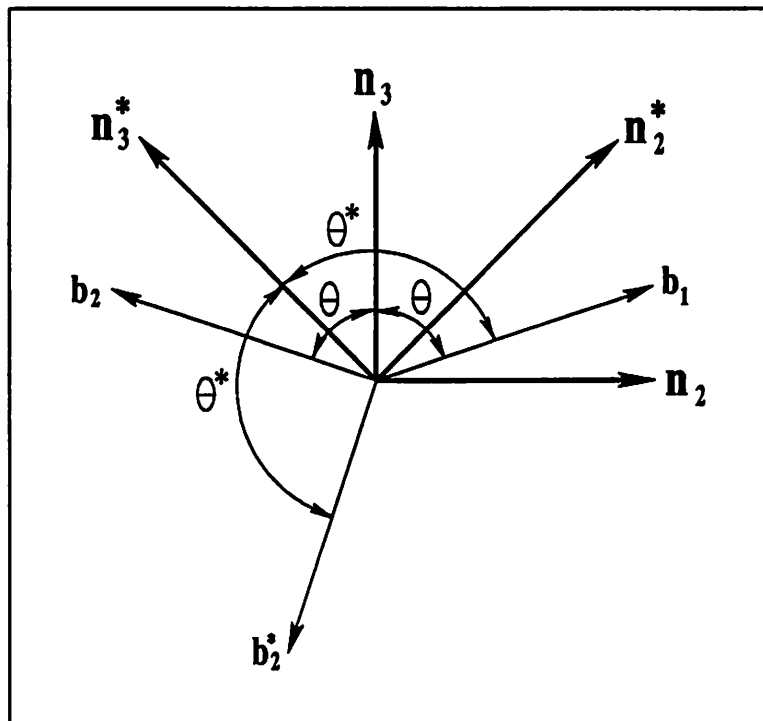


Figure 4.6: Unique solution as the common solution of two points.

sets of distinct eigenvectors, $\{n_2, n_3\}$ and $\{n_2^*, n_3^*\}$; one set each for the two points. One solution, say b_1 , is common to both the points. The second solution for each point, b_2 and b_2^* , respectively, is obtained by reflecting b_1 across the eigenvectors n_3 and n_3^* . These reflected solutions are in general different for each 3D point because the eigenvectors $\{n_2, n_3\}$ and $\{n_2^*, n_3^*\}$, of the matrix $[M_{com}]$, are different (except when the axis b passes through the origin, which is handled differently as shown earlier). Similarly for other points as well, the eigenvectors are distinct, hence the dual solutions obtained as reflections of the true solution are distinct.

The correct solution is found by clustering the sets of solutions obtained from several image trajectories. In general, out of the two solutions obtained for each trajectory, one set of solutions will form a dense cluster around the ideal solution. The solutions in the other set will either be spread out or there will be clusters of subsets of the solutions. Thus, the correct solution can easily be picked out by locating a compact cluster within which the solutions of all or a majority of trajectories is represented. The algorithm developed by Collins and Weiss [26] for clustering vectors on a unit sphere is used. The algorithm was originally developed for locating multiple vanishing points in images. Here the problem is simpler because only one cluster needs to be located.

After obtaining the estimates of b , c , d and k , the 3D vector for any point in any frame can be estimated by projecting its image vector onto the computed 3D trajectory. The 3D estimation algorithm can be summarized as follows:

For each trajectory found

Step 1: Find the eigencomponents of the matrix $[M_{com}]$ that represents the conic image trajectory.

Step 2: Compute the two solutions for b and c and the corresponding d and k .

The solution presented in Section 4.3.2 is used.

end for.

Step 3: Cluster the two sets of solutions over all trajectories.

Find the cluster that includes most trajectories within a small spread.

Step 4: Compute the best b and c by a least squares fit to the cluster.

For each point in a desired frame t

Step 5: Compute the corresponding 3D vector, \vec{r}_t using Equation 4.5.

Find the 3D vector nearest to \vec{r}_t that also lies on the reconstructed 3D trajectory.

end for.

4.4 Image Trajectory Description

4.4.1 Independent Conic Fitting

In principle, it should be possible to fit conic curves to each of the point tracks obtained by tracking each point over a number of frames. Ideally, if 360 degrees or a large fraction thereof of the 3D trajectory are available in the image, then even with noise, trajectories can be accurately described independent of each other. But in practical situations, due to self-occlusions or to limit the amount of data to realistic

levels, only a small part of the full trajectory, typically 50–100 degrees, is available. Given this scenario, conic curves can be very ‘creative’ when used to describe trajectories independently. Typically, a whole family of curves can equally well describe the same track.

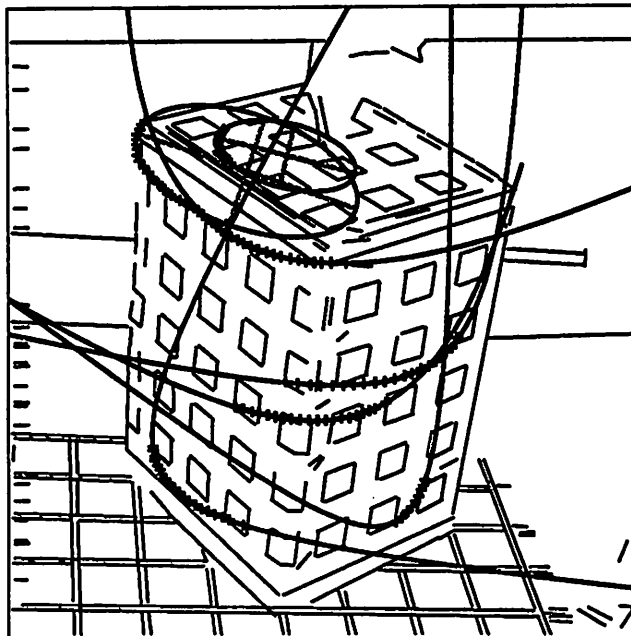
We observed the above behavior quite dramatically in our experiments. The input for the experiments is a set of point tracks derived from temporal point correspondences over many frames. Points are defined as intersection of lines forming plausible corners. However, any other method of reliably defining trackable point features would be acceptable. Two well-known methods were used to fit conic sections independently to the point tracks. First, Bookstein’s [16] closed-form least squares fit was used (see Appendix B for details). In this method, the square of the implicit defining equation of a conic is used as an algebraic distance measure for the distance of a point to a conic. This measure is minimized with an appropriate positive definite norm imposed on the parameters of the conic [16]. The optimal parameters can be found through a closed-form solution. Results of using this method on sample sets of point tracks for rotational data from two image sequences is shown in Figures 4.7a and 4.8a. The trajectories are shown overlaid on the first frame of lines extracted for each sequence. Leaving the details of how these sequences were captured to the results section, it is to be noted that all the trajectories should correspond to a single axis of rotation and should be elliptic given that all the 3D trajectories were fully in front of the image plane (Section 4.3.1). However, the unstable nature of the conic fits is evident from the figures. Not only is there no coherence amongst the fit parameters of different point tracks, but for one sequence (Figure 4.7a) most of the trajectories are hyperbolic instead of elliptic.

A similar behavior is reported by Porrill [69]. Porrill shows the instability of the ellipse fitting problem by plotting the confidence regions for the fits to small sections

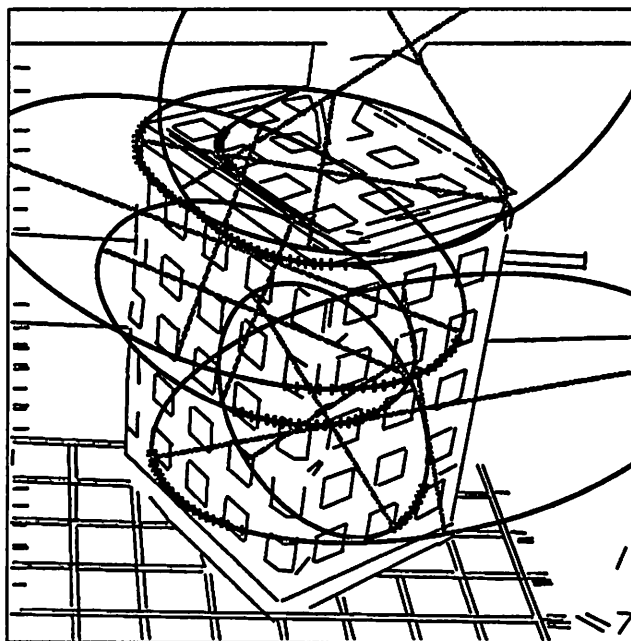
of the point data. Both he and Sampson [74] point out that the algebraic distance measure introduces a significant bias in the fit because it underestimates the actual distance of a point from a conic in high curvature regions. Consequently, the fitting algorithm tries to locate the input set of points in regions of high curvature on the fitted conic, thus keeping the distance measure low.

Furthermore, Porrill [69] and Sampson [74] suggest correcting the fitting bias by using a first order distance measure instead of the algebraic distance measure. In order to determine whether this would lead to a better fitting algorithm for the same data set, an algorithm based on the first order measure was developed. Note that now the fitting method is iterative and no longer closed-form [69, 74]. The iterative algorithm used a quasi-Newton unconstrained optimization algorithm as implemented in *Numerical Recipes in C* [95]. Results of this on the two data sets are shown in Figures 4.7b and 4.8b. It is evident from the figures that although there is an improvement in the fits, they still do not exhibit enough stability to make the common motion explicit. As a result, the 3D parameters derived from these fits are quite erroneous as will be shown in Section 4.5.2. While there might not be a bias in the results (as suggested by Porrill), there is still enough instability in the fitting process to render the results useless for any 2D grouping or 3D estimation based on the fits.

We emphasize that these failures are inherent in the computation of curve descriptions of noisy and quantized data obtained from imaging short segments of trajectories and not an intrinsic failure of the algorithms for independent fits. This became the point of departure for the investigation into methods for obtaining combined fits in such imaging scenarios.

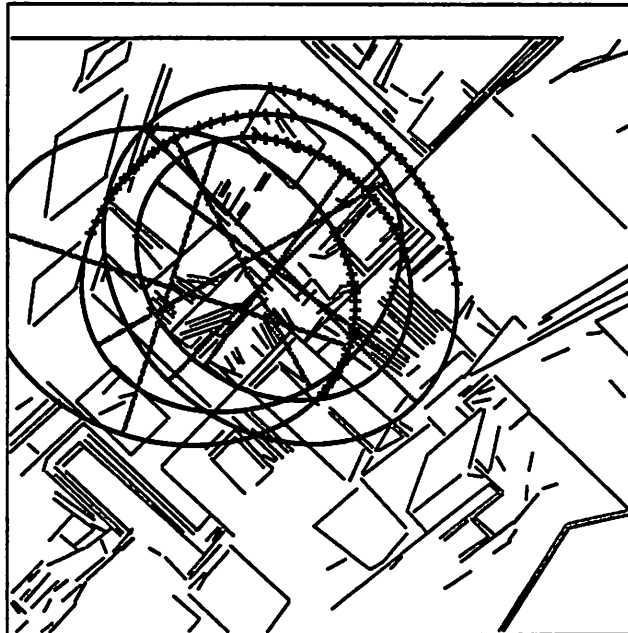


a) Conic fits using the algebraic distance.

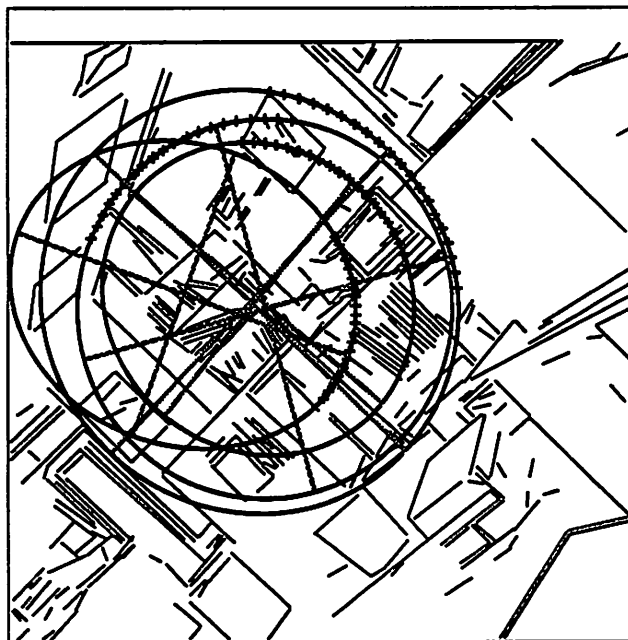


b) Conic fits using the first-order distance.

Figure 4.7: Sample independent conic fits for the *box-seq*. (a) Algebraic and (b) first-order distance solutions overlaid on frame 1 lines and the point tracks.



a) Conic fits using the algebraic distance.



b) Conic fits using the first-order distance.

Figure 4.8: Sample independent conic fits for the *room-seq-2*. (a) Algebraic and (b) first-order distance solutions overlaid on frame 1 lines and the point tracks.

4.4.2 Grouping Algorithm

The goal of the grouping algorithm is twofold — to obtain reliable trajectory fits to individual image point tracks and to make the similarities or dissimilarities across trajectories explicit. Robust fits to point tracks result in accurate estimation of 3D parameters. Explicit description of similarities across trajectories provides a basis for grouping various trajectories into a single object motion, detecting outliers, and possibly for detecting multiple object motions.

This algorithm is an incremental algorithm. At any stage, there is a set of point tracks already grouped together. The next track to be tried is picked using a grouping schedule discussed below. A least-squares fit over an error measure is performed on this new set. Using the best-fit parameters, an acceptability criterion is used to accept or reject the most recent track.

The algorithm is based on the following two observations:

1. Image trajectories resulting from 3D trajectories proximal in space can be well approximated by constraining three of their five individual parameters to be common across all of them. They should be of the same orientation and eccentricity and their centers should be collinear.
2. Point tracks which lie on non-overlapping segments of their corresponding trajectories constrain their combined fit better than ones which overlap (Figures 4.10–4.13).

The first constraint follows from a locally orthographic approximation for the projection of 3D trajectories. If two or more trajectories are proximal in space, then the corresponding image trajectories will have approximately collinear minor axes and the same eccentricities. This constraint is used to derive an error measure for the goodness-of-fit of trajectories over the participating point tracks. Note that this

constraint is not used as an a priori constraint to be satisfied by all the trajectories. Sets of trajectories which satisfy this constraint within reasonable fitting errors are discovered automatically by the algorithm.

The second observation is employed as a heuristic to design a grouping schedule that automatically selects the tracks to try next at any given stage of the incremental algorithm.

To derive a joint error measure for the tracks under consideration at any stage, conic sections are parameterized to make their common and distinct parameters explicit as shown in Figure 4.9. The collinearity of centers, and the common orientation

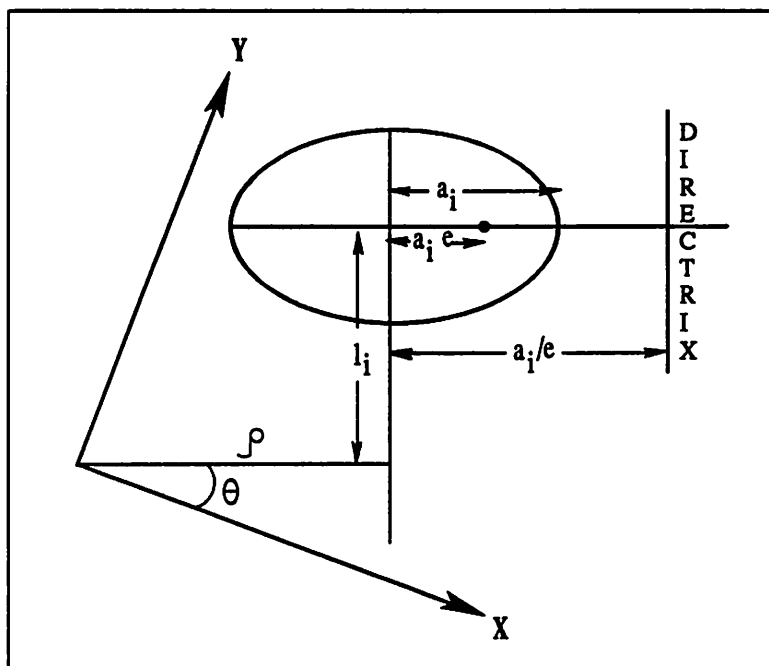


Figure 4.9: Conic section parameterization illustrated for an ellipse.

is represented by a line parameterized by ρ and θ . The third common parameter is e , the eccentricity. The two distinct parameters for trajectory i are l_i , the location

of the conic center along the common line and a_i , the length of the x-intercept. The equation of a conic section in terms of this parameterization is,

$$f(x, y; \rho, \theta, e, l_i, a_i) = (x \cos \theta + y \sin \theta - \rho)^2(1 - e^2) + (-x \sin \theta + y \cos \theta - l_i)^2 - a_i^2(1 - e^2) = 0 \quad (4.21)$$

where (x, y) are the image coordinates of points along the i th track. A first-order measure of the distance of a point from a conic curve is defined as,

$$F_{it} = |f(x_{it}, y_{it}) / |\nabla f(x_{it}, y_{it})|| \quad (4.22)$$

where the subscript it refers to the t th frame of the i th track. The following minimization leads to the optimum parameters for the trajectories of the current set:

$$\min_{\rho, \theta, e, l_i, a_i} \sum_t \sum_i F_{it}^2(\rho, \theta, e, l_i, a_i) \quad (4.23)$$

A Conjugate Gradient algorithm [95] with scaling among the variables ρ , θ and e is used for the optimization. Having found the best-fit parameters for the current set, an acceptability criterion is applied to decide whether to accept the last point track or not. The residual error for each track resulting from the application of its newly found combined-fit parameters is computed. If this error for each track in the set is within an experimentally determined scale factor of its residual error from the independent fit, then the most recent track is accepted in the current group and the process is repeated for the next track [2, 24]. Otherwise, the repetition is done with the next track after rejecting the current one.

This acceptability criterion is reasonable because even though the parameters of independent fits may be erroneous, the resulting error residual is a measure of how well the underlying track can be described as a conic curve. When combining the description of a set of point tracks, the trade-off is between the compactness of description and the residual error. This is similar to the minimum description length

formulation of image segmentation problems [53]. Each new point track included in a combined description reduces the number of parameters by three. For n tracks, individual descriptions require $5n$ parameters whereas a combined description requires only $3 + 2n$ parameters. But the addition of a new track can increase the residual error. However, if the residual error from the combined-fit description does not increase substantially or decreases, then it is better to accept a track as grouped.

Now we discuss the design of the grouping schedule. The essential goal here is to describe image trajectories which reflect the 3D geometry accurately.

Through the grouping schedule, the goal is for short noisy point tracks to progressively constrain their mutual fits, avoiding arbitrary local minima in the process. One heuristic we apply is that tracks, even when short, mutually constrain their trajectories better if they cover a larger span around their trajectories without overlap than if they do with any overlap. For instance, the point tracks in Figure 4.10 provide a stronger constraint for the correct combined description than those in Figure 4.11. Further, tracks which are proximal are more likely to satisfy the grouping criterion and hence should be tried first. Tracks which do not overlap and are proximal are given preference over those which are distant in the image. For example, the tracks shown in Figure 4.12 are tried before those in Figure 4.13. Note that this grouping schedule is designed only to define an ordering on the whole set of tracks input to the algorithm. The goal is to force early consideration of those tracks which provide the strongest mutual constraints, thus reducing the possibility of false minima and forcing the generation of good initial guesses for the successive optimization steps. However, no point tracks are left untried but each one is picked at a stage governed by this schedule. Thus, in general, rapid convergence to the correct combined-fits is achieved.

To implement this grouping schedule, each track is projected on the four sides of the XY -rectangle defined by the image plane as shown in Figures 4.10–4.13. Thus,

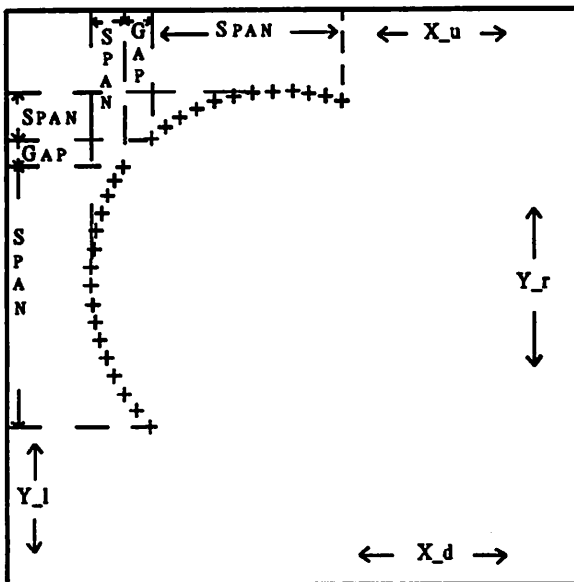


Figure 4.10: Case of small overlap along projections. Two point tracks with small overlap along projections.

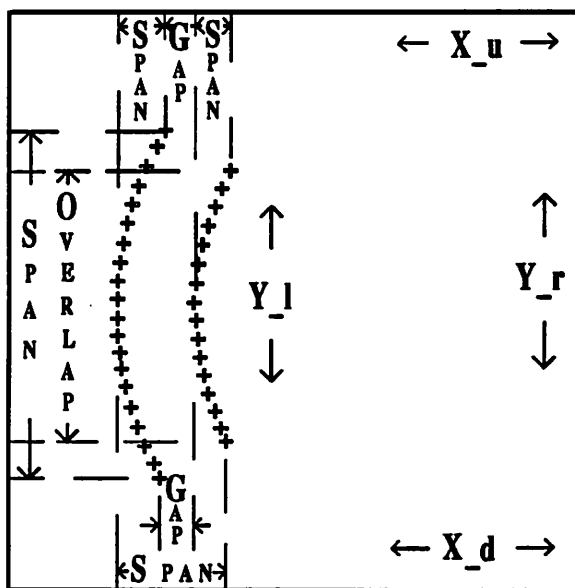


Figure 4.11: Case of large overlap along projections. Two point tracks with large overlap along projections.

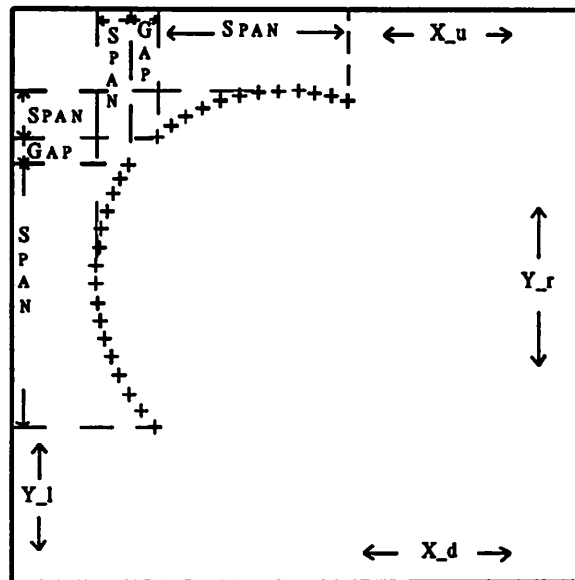


Figure 4.12: Case of no overlap and small gap. Two point tracks with no overlap and small gap along projections.

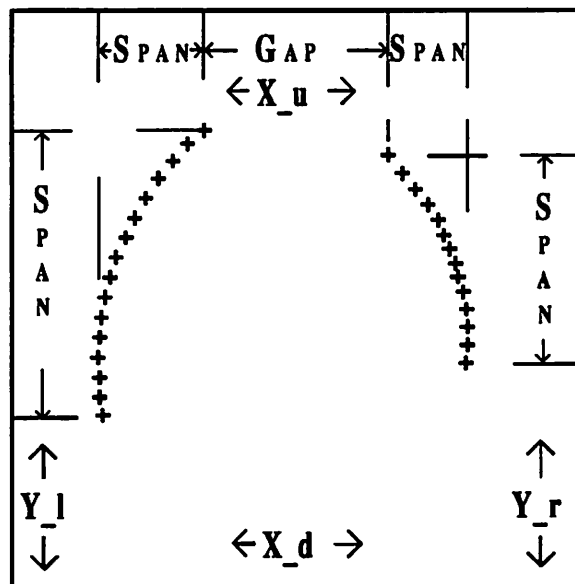


Figure 4.13: Case of no overlap but a large gap. Two point tracks with no overlap but a large gap along projections.

each track has four projections — X_u , X_d , Y_l and Y_r — some of which could be zero. For all the tracks in the current grouped set and for each untried track in the current cycle, the sum of the following measure along each side is computed:

$$gsm_p = span_p - overlap_p - gap_p, \quad p \in \{u, d, l, r\} \quad (4.24)$$

where $span_p$ is the sum of connected projections along the side p , $overlap_p$ is the sum of overlaps amongst the projections of all tracks along p and gap_p is the gap between projections as shown in the figures. A new point track that minimizes

$$\sum_{p \in \{u, d, l, r\}} gsm_p \quad (4.25)$$

with the current grouped set is chosen to be tried next.

An outline of the TRAJ-DESC algorithm follows:

Step 1: Fit conics to each point track independently and record the error residual.

Step 2: Set $ntry = 2$. (Start with an initial set of 2).

Step 3: Set the current set to include only the track with the longest arc length.

A polygonal approximation is used for the arc length.

Repeat until (No more tracks left ungrouped) or

(All tracks have been tried for this cycle)

Step 4: Add to the current set the track that minimizes the sum of the

gsm measure of Equation 4.25 along with the tracks already in the set.

Call this new track the $ntry$ th track.

Step 5: Solve the optimization problem of Equation 4.23 for the current set and obtain the best-fit parameters.

Step 6: If the fit error for each track is within a scale factor of its independent residual,

then accept the current set as grouped and increment *ntry* by one,

else reject the last track in the current set.

end repeat.

At start-up, the initial guess for the optimization step, Step 5, is generated from the parameters of the track with the longest arc-length. Subsequently, whenever a new track is assimilated, the initial guess for the next step is the current set of best-fit parameters for the grouped tracks. In our experiments, we have rarely encountered local minima. This implies better convergence properties for this formulation and the combined-fit error measure compared to one where 3D parameters are directly solved for from the image data. Furthermore, the grouping schedule progressively constrains the combined fits, generating initial guesses which lead to fast convergence.

4.5 Experimental Results

Results of the trajectory grouping algorithm, **TRAJ-DESC**, and the 3D estimation algorithm, **3D-EST**, on two image sequences are presented in this section. Both sequences were digitized with a GOULD frame grabber that outputs 512 by 484 pixel images. These were reduced further to 256 by 242 pixels for our experiments.

The first sequence, called the *room-seq-2*, is a set of images taken inside a robotics laboratory. Objects in the scene varied in depth from 10 to 30 feet. Twenty-five frames were captured with a SONY B/W XC-77 camera mounted on a PUMA arm which in turn was mounted on a platform at one end of the room. The effective field of view (FOV) of the lens-camera and grabber systems was found to be 42 by 40 degrees using the method of Lenz et al. [55]. The arm was rotated with the axis of rotation nearly parallel to the optical axis of the camera as constrained by the configuration of the gripper. The distance between the optical axis and the rotation axis was measured to be about 1.7 feet. The angle of rotation between consecutive frames was 4 degrees. Two frames of this sequence are shown in Figure 4.14.

For the second sequence, called the *box-seq*, a rectangular chequered box was clamped at the end of a cartesian robot arm. A stick which pierced the box along its longest dimension was used to grip the box. The box was rotated by the arm around this stick-axis, while the camera looked down obliquely at it. Using a SONY B/W AVC-D1 camera, with effective FOV 24 by 23 degrees, a sequence of 20 frames was captured. The approximate angle of rotation between consecutive frames was 3.6 degrees. The range of depths in this scene was about 550 to 700 mm. Two image frames are shown in Figure 4.16.

To generate point tracks, corner-like points defined by line intersections were tracked using the line-tracking system of Williams et al. [96]. This system tracks lines obtained from the line-extraction algorithm of Boldt et al. [14] by predicting their appearances in successive frames using the displacement field output of the algorithm by Anandan [8]. The displacement fields between frames 1 and 2 of *room-seq-2* and *box-seq* are shown in Figures 4.15 and 4.17, respectively. Figures 4.18 and 4.20 show a sample set of tracked lines overlaid on frame 1 lines for the respective sequences. Figures 4.19 and 4.21 depict the respective point tracks that form the

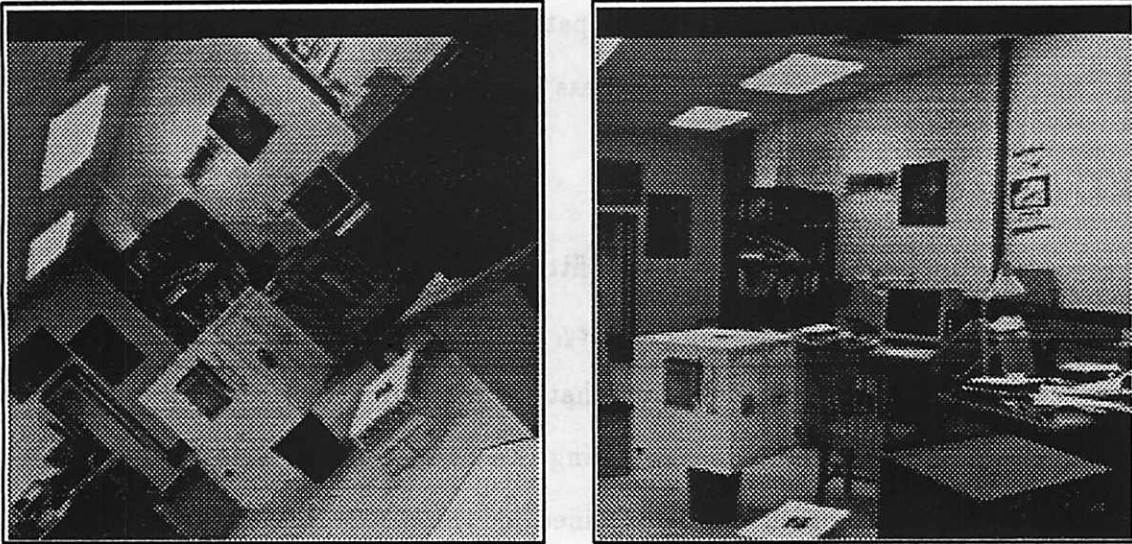
input to the algorithm TRAJ-DESC described earlier. Figures 4.19 and 4.21 underscore the similarity between the problem of making a common motion explicit amongst these motion-generated "Glass patterns" and the perceptual organization of geometric structure in Stevens' [84] Glass patterns.

4.5.1 Trajectory Grouping Results

Figures 4.7 and 4.8 show the results of fitting trajectories independently to sample sets of points for the *room-seq-2* and *box-seq*, respectively. As was discussed earlier, it is graphically evident from these figures that there apparently is nothing in common between the motion of the points generating these trajectories. This is amply borne out by the highly inaccurate results obtained for the 3D depth of these points (e.g. see Tables 4.2 and 4.4).

Algorithm TRAJ-DESC was run on 30 point tracks obtained from the *room-seq-2*. Figure 4.22 shows the output of this algorithm on two sample sets consisting of 8 point tracks. There is a visually dramatic improvement in the nature of the resulting trajectories. The common axis of rotation becomes explicit by the collinearity of the minor axes of the trajectories. This makes the resulting 3D parameters very accurate. Note that for this sequence, where the rotation is nearly parallel to the image plane, the grouping constraint described in Section 4.4.2 is globally valid. The image trajectories in this case are expected to be nearly circular.

Figure 4.23 shows similar results for the *box-seq*. Again, in contrast with the independent fits (Figure 4.7), the new trajectories make the common 3D motion of the underlying points explicit by the *approximate* collinearity of the minor axes of various groups. Note that for this case of motion, under perspective projection, the minor axes should not be collinear *globally*. The algorithm was tried on 50 point tracks obtaining the maximally grouped tracks. Five of the groups consisting of 18 tracks are shown in Figure 4.23.



a) Image Frame 1

b) Image Frame 13

Figure 4.14: Two frames of the *room-seq-2*. Frames 1 and 13.

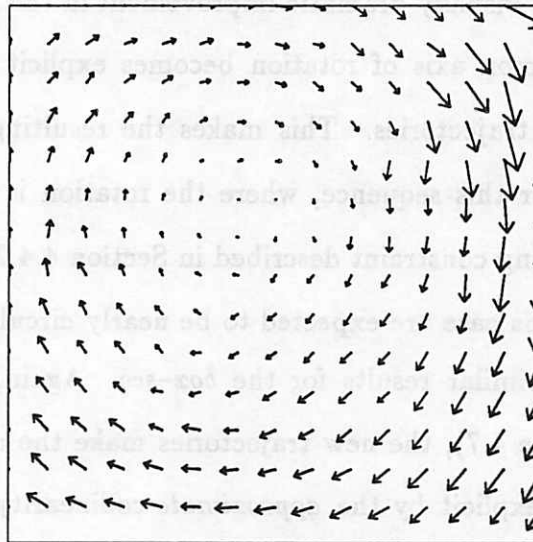
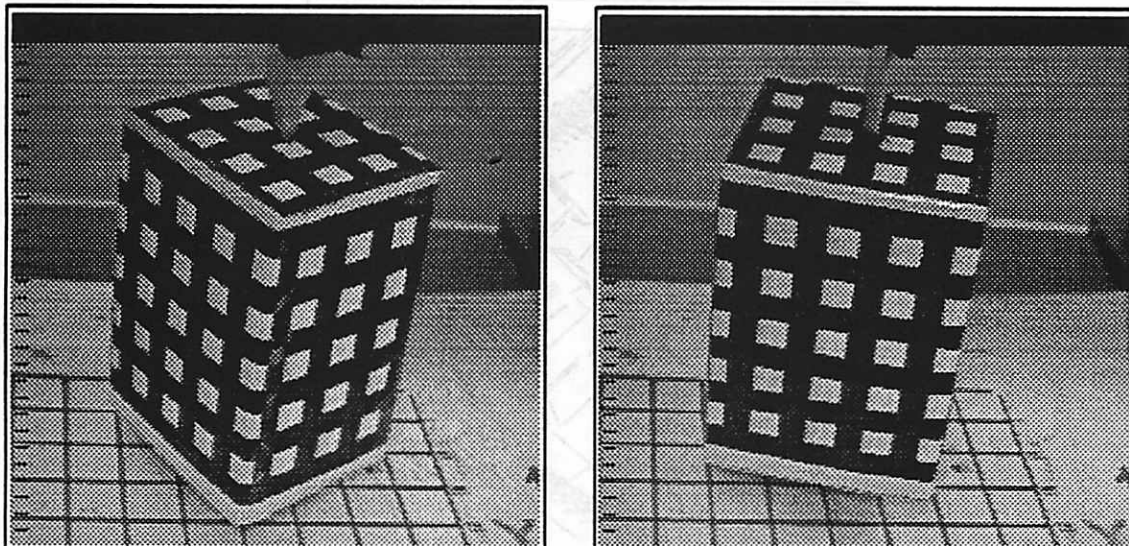
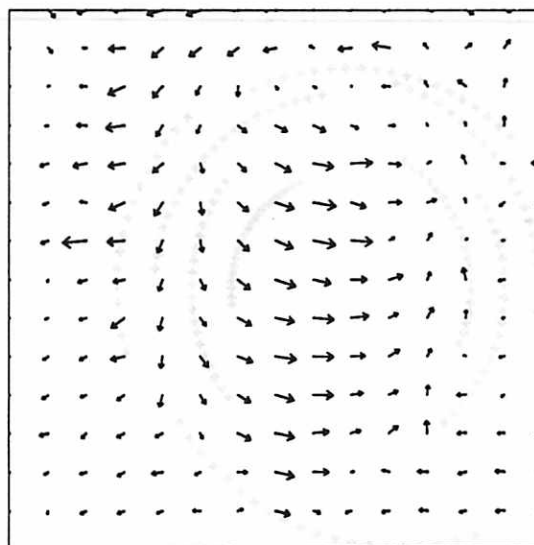


Figure 4.15: Displacement field for the *room-seq-2*. Sub-sampled field (16×16) between frames 1 & 2.



a) Image Frame 1

b) Image Frame 13

Figure 4.16: Two frames of the *box-seq*. Frames 1 and 13.Figure 4.17: Displacement field for the *box-seq*. Sub-sampled (16×16) field between frames 1 & 2.

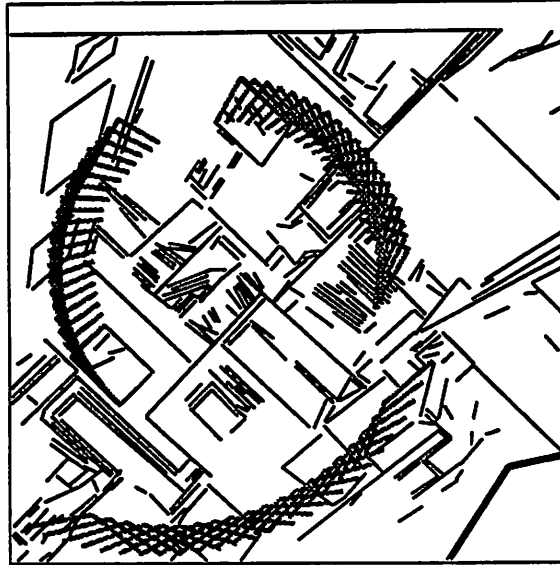


Figure 4.18: Tracked lines for the *room-seq-2*. A sample set overlaid on frame 1 lines.

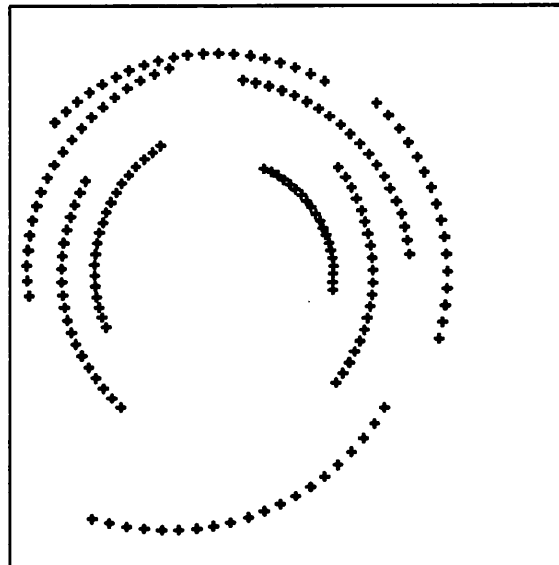


Figure 4.19: Sample point tracks for the *room-seq-2*. Corner points are tracked.

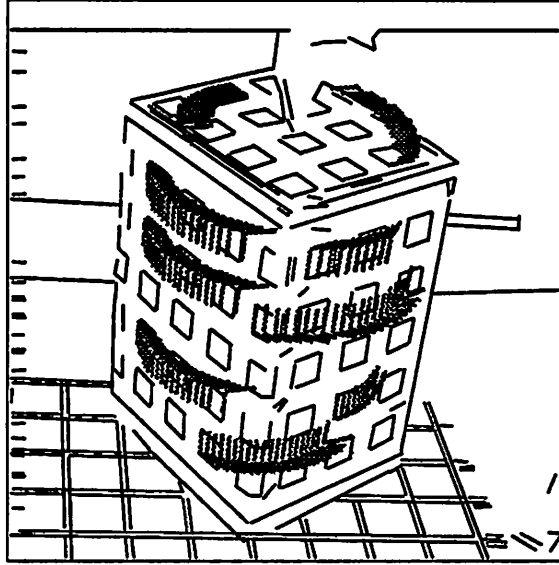


Figure 4.20: Tracked lines for the *box-seq*. A sample set overlaid on frame 1 lines.

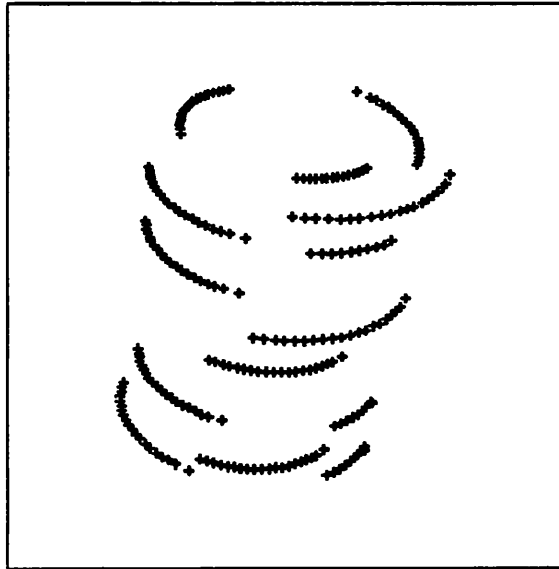
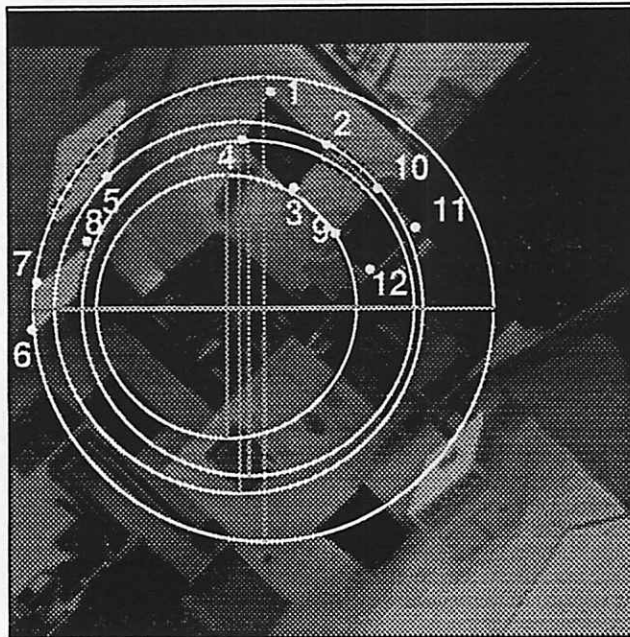
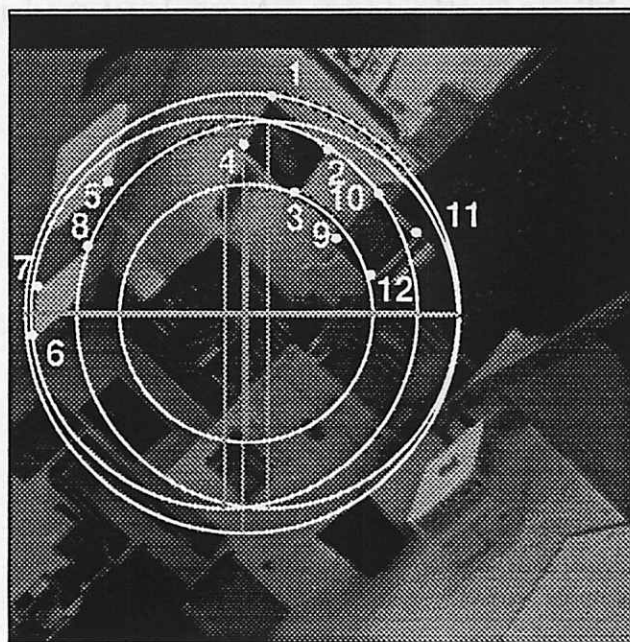


Figure 4.21: Sample point tracks for the *box-seq*. Corner points are tracked.

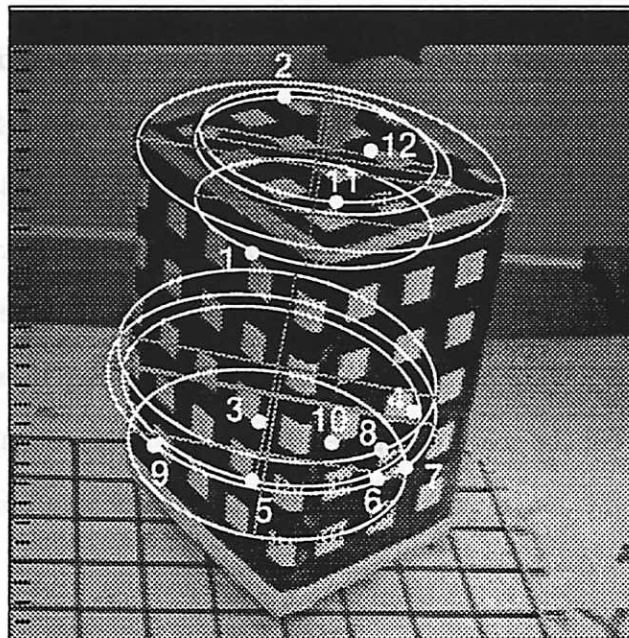


Combined conic fits -- Sample set 1.

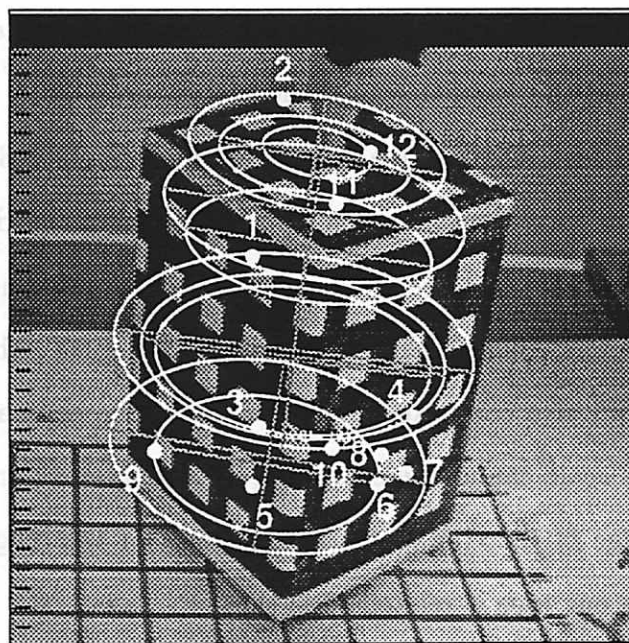


Combined conic fits -- Sample set 2.

Figure 4.22: Combined conic fits for the *room-seq-2*. Fits to eight sample 20-frame point tracks. Overlaid on image frame 1.



Combined conic fits -- Sample set 1.



Combined conic fits -- Sample set 2.

Figure 4.23: Combined conic fits for the *box-seq*. Fits to sample 20-frame point tracks. Overlaid on image frame 1.

4.5.2 3D Estimation Results

The 3D trajectory parameters were computed from the closed-form solution developed in Section 4.3.2 using the trajectories shown in Figures 4.22 and 4.23. The 3D results are in the form of distances (magnitudes of the reconstructed 3D vectors) and depths (Z component of the vectors) to points, and the rotation axis direction, \hat{b} , and location, \hat{c} . These are found by the algorithm given at the end of section 4.3.3. The reference distances were obtained by actually measuring these from the camera while the reference depths were computed using the pose estimation algorithm of Kumar and Hanson [50]. A model of the reference points was built in a fixed world coordinate system for the latter.

Room Sequence

For the *room-seq-2*, the reference distances and depths were computed for a set of twelve points, called the *sample set*, labelled in Figure 4.22³. The distances are accurate to about 0.1 feet. Given the accuracy in measuring the 3D coordinates, the depths computed from Kumar and Hanson's pose estimation algorithm in the camera frame are accurate to about 2 percent; these were used as reference depth estimates for comparison purposes. The scale for algorithm 3D-EST, the $|\bar{c}|$ of equation 4.9, was measured and also computed from the pose. The estimated scale was 1.67 feet. Recall that this scale cannot be computed by any monocular motion algorithm without the knowledge of the true distance to a point or the magnitude of effective translation (which in our case is related to $|\bar{c}|$).

In Table 4.1, the 3D reconstructions for the *sample set* of 12 points, selected from the grouped trajectories, are compared with the measured and pose computed

³Trajectories for only some of the points from the *sample sets* of both sequences are shown in the figures to avoid clutter. However, all the points have been labelled.

distances. In both the comparisons, the average percentage error is between 2 to 3 percent and all the errors are below 5 percent.

Table 4.1: 3D distance comparisons for the *room-seq-2*. Comparison of 3D distances computed by Rotational Trajectories with true distances and pose computed distances for 12 sample points of the *room-seq-2* (in feet).

Point Num.	True Distance	Rot. Distance	Error (%)	Pose Distance	Rot. Distance	Error (%)
1	18.25	18.54	1.59	18.46	18.54	0.43
2	17.94	17.47	-2.62	17.79	17.47	-1.80
3	19.59	19.75	0.82	19.40	19.75	1.80
4	19.90	19.75	-0.75	20.00	19.75	-1.25
5	22.65	22.84	0.84	23.36	22.84	-2.22
6	29.29	28.28	-3.44	29.12	28.28	-2.88
7	25.65	24.87	-3.04	25.31	24.87	-1.74
8	26.25	25.55	-2.67	26.80	25.55	-4.66
9	14.90	14.83	-0.47	15.06	14.83	-1.53
10	14.60	14.61	0.07	14.53	14.61	0.55
11	14.35	15.12	5.37	14.52	15.12	4.13
12	14.70	15.19	3.33	14.70	15.19	3.33
Avg. Abs. Err. 2.08%				Avg. Abs. Err. 2.19%		

In Table 4.2, the tremendous improvement achieved by TRAJ-DESC vis-a-vis the independent fits is demonstrated by comparing the 3D distances for the *sample set* computed from independent fits with those from grouped fits. The large improvement in accuracy obtained by the latter is evident from the comparison.

It was mentioned earlier that two-frame algorithms can be very unreliable in computing depth estimates especially when there is significant rotation and translation parallel to the image plane [3, 94]. This is indeed the case for the *room-seq-2*. Depths for the *sample set* were computed using Horn's [43] relative orientation algorithm. It was run on a set of 22 points spread out over the image; the *sample set* is a subset of

Table 4.2: Independent vs. grouped fit 3D distances for the *room-seq-2*. 3D Distances from Independent (Ind.) Fits vs. Grouped (Gpd.) Fits for the 12 sample points compared with true distances for the *room-seq-2* (in feet).

Point Num.	True Distance	Rot. Dist. Ind. Fit	Error (%)	Rot. Dist. Gpd. Fit	Error (%)
1	18.25	5.98	-67.23	18.54	1.59
2	17.94	9.73	-45.76	17.47	-2.62
3	19.59	5.60	-71.41	19.75	0.82
4	19.90	4.68	-76.48	19.75	-0.75
5	22.65	7.96	-64.86	22.84	0.84
6	29.29	5.01	-82.90	28.28	-3.44
7	25.65	6.71	-73.84	24.87	-3.04
8	26.25	13.40	-48.97	25.55	-2.67
9	14.90	3.44	-76.91	14.83	-0.47
10	14.60	5.93	-59.38	14.61	0.07
11	14.35	4.65	-67.60	15.12	5.37
12	14.70	5.86	-60.14	15.19	3.33
Avg. Abs. Err. 66.29%				Avg. Abs. Err. 2.08%	

this set. To minimize the effects of noise, frames 1 and 19 were chosen, as the average image motion between these was as large as 99 pixels with a standard deviation of 22 pixels. The effective translation magnitude between the two frames is about 1.7 feet. This translation, being almost parallel to the image plane, results in an image motion of about 30 pixels for a point approximately 15 feet in depth. In Table 4.3, the distances for the *sample set* obtained from the two-frame algorithm are compared against the trajectory results, using the measured distances as a reference. One qualitative reason for the bad performance of two-frame computations for this motion is that, with noise, there is an ambiguity between rotations in depth and translations parallel to the image plane [3]. It should be emphasized here that the estimates obtained from two-frame computations cannot be improved even if they are averaged

Table 4.3: Two-frame vs. trajectory 3D distances for the *room-seq-2*. Two-frame 3D Distances vs. Trajectory 3D Distances for the 12 points compared with True Distances for the *room-seq-2* (in feet).

Point Num.	True Distance	Two-frame Distance	Error (%)	Rot. Distance	Error (%)
1	18.25	10.54	-42.25	18.54	1.59
2	17.94	10.32	-42.47	17.47	-2.62
3	19.59	10.72	-45.28	19.75	0.82
4	19.90	10.90	-45.23	19.75	-0.75
5	22.65	11.59	-48.83	22.84	0.84
6	29.29	12.79	-56.33	28.28	-3.44
7	25.65	12.01	-53.18	24.87	-3.04
8	26.25	12.34	-52.99	25.55	-2.67
9	14.90	9.22	-38.12	14.83	-0.47
10	14.60	9.12	-37.53	14.61	0.07
11	14.35	9.09	-36.66	15.12	5.37
12	14.70	9.04	-38.50	15.19	3.33
Avg. Abs. Err. 44.78%			Avg. Abs. Err. 2.08%		

over many pairs of frames over time. For the *room-seq-2*, for example, averaging over a number of pairs leads to no improvement in results.

The least-squares axis direction vector computed by 3D-EST for the *room-seq-2*s (0.0113, -0.0049, 0.9999). The average spread of all the vectors around this best estimate is 0.228 degrees. The best location estimate is (0.9095, 0.4156, -0.0076) and the average spread is 9.078 degrees. The reason for the higher variance in the location estimates is that the trajectories are nearly circular and their centers are concentrated in a relatively small region. For nearly circular curves, the orientation of the minor axis is ambiguous. So for the trajectories of this sequence, the positions of the centers can be shifted slightly without changing their circularity or size. This ambiguity in the orientation of the collinear centers and their location leads to a larger variance in the axis location vector estimates for different points. In contrast, the size

or the eccentricity (i.e. circularity here) of trajectories is estimated unambiguously. Hence, the 3D distances and the axis direction are found very precisely.

Box Sequence

For the *box-seq*, the (x, y, z) coordinates of a set of points were measured on the three faces of the box to within $1mm$. Again, given the accuracy of these measurements, it is not unreasonable to use depths estimated from pose as the reference depths for the comparisons [51]. The magnitude of \bar{c} , the scale for 3D-EST, was estimated to be $569.66mm$.

In Table 4.4, the depth estimates from independent and grouped fits for a *sample set* of 12 points are compared. The points are labelled in Figure 4.23. Again, the significant improvement in depths from the latter is evident. The percentage errors in depth computation by 3D-EST are well within 2 percent with the average at approximately 0.87 percent.

Horn's algorithm was run over 40 points in two frames (1 and 8) between which the average image motion was 17 pixels with a standard deviation of 7 pixels. Depth results for the *sample set* are compared to those obtained from 3D-EST in Table 4.5. For this case, the two-frame depth results are good with an average percentage error of approximately 4 percent. This suggests that one might improve the two-frame results over many pairs. To test this, the depths for the points in the *sample set* were computed for six pairs of frames with frame 1 as the anchor (all the depths were computed in frame 1). On averaging these, it is seen that the depth errors are slightly lower than those obtained from the trajectory algorithm. This is also shown in Table 4.5.

The best axis direction was computed to be $(0.1521, -0.8340, 0.5303)$ with an average spread of 1.512 degrees among the individual estimates. The best axis location was estimated to be $(-0.0692, 0.5261, 0.8476)$ with a spread of 1.203 degrees. In

Table 4.4: Independent vs. grouped fit 3D depths for the *box-seq*. 3D Depths from Independent (Ind.) Fits vs. Grouped (Gpd.) Fits for 12 sample points compared with Pose Depths for the *box-seq* (in mm.).

Point Num.	Pose Depth	Rot. Depth Ind. Fit	Error (%)	Rot. Depth Gpd. Fit	Error (%)
1	591.38	257.71	-56.42	588.89	-0.42
2	666.34	572.13	-14.14	665.84	-0.08
3	621.78	664.52	6.87	617.77	-0.64
4	640.66	353.75	-44.78	635.02	-0.88
5	637.68	726.36	13.91	637.71	0.01
6	647.94	370.81	-42.77	650.89	0.46
7	656.56	603.94	-8.01	661.89	0.81
8	639.99	974.83	52.32	653.80	2.16
9	709.68	675.07	-4.88	700.74	-1.26
10	614.79	56.93	-90.74	603.58	-1.82
11	602.34	527.88	-12.36	606.16	0.63
12	628.94	11.18	-98.22	636.52	1.21
Avg. Abs. Err. 37.12%			Avg. Abs. Err. 0.87%		

Table 4.5: Two-frame vs. trajectory 3D depths for the *box-seq*. Two-frame Depths vs. Trajectory 3D Depths for the 12 points Compared with Pose Depths for the *box-seq* (in mm.).

Point Num.	Pose Depth	Two-frm. Depth	Error (%)	Rot. Depth	Error (%)	Two-frm. Avg. Depth	Error (%)
1	591.38	613.88	3.80	588.89	-0.42	591.68	0.05
2	666.34	694.42	4.21	665.84	-0.08	666.44	0.02
3	621.78	648.36	4.27	617.77	-0.64	624.91	0.50
4	640.66	667.46	4.18	635.02	-0.88	641.54	0.14
5	637.68	665.04	4.29	637.71	0.01	639.58	0.30
6	647.94	679.23	4.83	650.89	0.46	651.68	0.58
7	656.56	687.45	4.70	661.89	0.81	658.75	0.33
8	639.99	667.96	4.37	653.80	2.16	642.31	0.36
9	709.68	744.84	4.95	700.74	-1.26	714.42	0.67
10	614.79	644.11	4.77	603.58	-1.82	618.49	0.60
11	602.34	626.85	4.07	606.16	0.63	604.80	0.41
12	628.94	655.33	4.20	636.52	1.21	630.46	0.24
Avg. Err. 4.39%			Avg. Err. 0.87%		Avg. Err. 0.35%		

this sequence the trajectories are highly eccentric (average eccentricity computed to be approximately 0.8) so the directions of major and minor axes are well-defined. Hence, as expected, the variances in the estimates of both the direction and location are small.

4.6 Summary

In this chapter, techniques for spatial and temporal grouping are developed for the problem of reconstruction of 3D trajectories of point features in a scene from image trajectories generated through rotational motion. A new technique for reliably computing 3D structure from a sequence of images of a scene undergoing a relative rigid-body rotation with respect to the camera is presented. For structure reconstruction, intentional rotational motion between the scene and a camera can be carried out by either a camera mounted on a robotic arm or by rotating objects on a platform with the camera fixed. Our approach is an alternative to two traditional approaches for reconstruction: structure estimation over many frames using two-frame motion and structure estimates, and 3D structure and motion estimation based on specific models of 3D motion *directly* from discrete point correspondences.

The approach presented here decomposes the 3D reconstruction problem into that of first describing image motion as curved trajectories (conics) in the image plane from the discrete point correspondences, and subsequently reconstructing the 3D trajectory from the image trajectories. It is shown that in realistic imaging scenarios when only small segments of the 3D trajectory are imaged, descriptions of individual image trajectories are very unstable. There is a need to constrain the 2D trajectory descriptions of many imaged features into combined descriptions to achieve robust image trajectories and the reconstructed 3D trajectories. A grouping algorithm is developed which uses a grouping constraint to automatically select trajectories of

features and combine them into a description that is more compact than their individual descriptions while keeping the residual errors within limits of the expected noise levels.

The improvement in trajectory descriptions is substantiated by the accuracy of the reconstruction of the corresponding 3D trajectories. A new closed-form solution for the reconstruction of a 3D circular trajectory from the corresponding imaged conic trajectory, under perspective projection, is presented. This solution is applied to image conics obtained through the grouping algorithm. Results on real image sequences from both stages show an ability to achieve reliable 3D reconstruction. Furthermore, for rotations around an axis approximately parallel to the optical axis, significant improvement in 3D reconstruction is achieved in contrast with the inaccurate results from standard two-frame structure-from-motion algorithms. In addition, it is also argued that the incremental grouping algorithm for describing image trajectories is potentially useful for segmenting multiple object motions.

The improved performance of the algorithm developed in this chapter over previous approaches again underscores our claim that the combination of spatial and temporal constraints leads to more robust 3D reconstruction than either alone.

Future directions for this research are presented in the concluding chapter of this thesis (Chapter 5).

CHAPTER 5

SUMMARY AND FUTURE WORK

A major goal for the research described in this dissertation was to demonstrate the integration of spatial and temporal constraints over extended-time image sequences to compute robust scene structure from image motion. Two demonstrative problems were selected to highlight the integration of spatial and temporal constraints — one in which constraints on object structure are combined with those from smoothness of motion, and another in which a model of rotational motion is combined with constraints from spatial proximity and continuity. The major contributions of the work in the design of algorithms for the two problems can now be summarized.

5.1 Major Contributions

We have shown that combining spatial constraints of a shallow structure with the temporal constraints of smooth motion leads to a robust 3D reconstruction of shallow objects in the scene without requiring the intermediate step of the computation of 3D motion. This intermediate step of decomposition of image motion into a rotational and a translational component can lead to large errors in the subsequent computation of 3D structure due to certain inherent ambiguities in the computation of 3D motion.

In order to use the shallow structure constraint without a priori knowledge of the location of shallow structures in the image, an identification of such structures is required, along with a mechanism for maintaining their identity over time. The results

presented in Chapter 3 show that tracking within a Kalman filtering framework, and model-matching provide a sound framework for the identification of shallow structures and for maintenance of their identities over time. An important contribution of the work is the use of 3D motion and shallow structure constraints to track aggregates of image features which potentially are shallow structures. This is in contrast with traditional token tracking which utilizes heuristics about the image motion of tokens.

We have shown that incorporating motion modeling and measurement uncertainties into the mechanism for tracking aggregate structures provides resilience to errors due to token extraction, motion discontinuities and occlusions. Aggregate structures provide implicit figural constraints for relatively unambiguous matching. Incorporating uncertainties for the structure as a whole leads to reliable matching in comparison with those methods that match and track individual tokens without using the aggregate figural context. Aggregate structure matching using a statistical distance measure like the Mahalanobis distance would in general involve the inversion of large covariance matrices. However, we showed in Chapter 3 that decomposing the state vector that represents an aggregate structure into components that are almost independently affected by modeling and measurement uncertainties, requires the inversion of only small matrices.

The trackability of an aggregate structure is used as a criterion for its identification as a shallow structure. The integration of the spatial constraint of shallowness with the temporal constraint of smooth motion allows the use of tracking as a process of verification of the structural constraints. Predictions for non-shallow structures, under the affine constraints valid for shallow structures, lead to large errors in tracking. These errors can be used to label an aggregate structure as shallow or otherwise. Furthermore, tracking also leads to an incremental refinement of its depth estimate.

The efficacy of the central claims of this thesis are further demonstrated by considering the problem of reliable 3D reconstruction of a scene undergoing a rotational

motion relative to a camera. Points in the scene describe conic trajectories in the image plane under perspective projection. The goal is to describe the discrete point correspondences in the image plane as conic trajectories that can subsequently be used for the reconstruction of the corresponding 3D trajectories. An important contribution here is a grouping algorithm that combines the image trajectory descriptions of a set of point features to achieve robust image trajectories. The algorithm is motivated by the unreliable trajectories that are obtained when point trajectories are processed independently.

The advantage of describing trajectories by the grouping algorithm is further demonstrated by the promising 3D results obtained on real image sequences. A contribution here is a new closed-form solution for the problem of reconstruction of a 3D trajectory from a 2D conic image trajectory obtained under perspective projection. The solution is simpler than those available in the literature partly because of the particular parameterization chosen for the 3D geometry of the problem.

5.2 Directions for Further Research

The shallow structure representation achieved in this work can be useful for obstacle avoidance and path planning. Its utility can be further enhanced if visibility cues can be utilized to distinguish between shallow holes and surfaces. In the context of stereo reconstruction, some work towards representing free and filled space using Delaunay triangulation has been done by Le Bras-Mehlman et al. [17]. In addition, combining the shallow structure constraints with the figural organization of occluding boundaries [97] is another promising area of further research. We have used convexity heuristics to instantiate potential shallow structures. However, occluding boundaries and figural constraints can provide reliable information in a static image for object delineation. This static image constraint can be combined with dynamic testing for

shallowness to achieve an organization of occluding contours and figures into shallow and non-shallow structures.

Our work has used point and line tokens as the primitive image measurements for the subsequent 3D interpretation. In relatively unstructured outdoor scenarios, these tokens may not be consistent descriptors of structure in images over time. Presence of texture can be utilized in these domains along with occluding boundaries. Thus, extensions to shallow structure tracking and reconstruction for handling image intensity regions, along with points and lines, is another research direction that could make the algorithm more widely applicable.

Shallow structure reconstruction is a partial derivation of scene structure in terms of fronto-parallel planes. The ultimate goal of 3D reconstruction from motion is the description of the scene in terms of surface patches with the boundaries – depth and orientation – made explicit. To be able to achieve this, there is a need to conduct research in two directions. First, representations for 3D surface shape and motion need to be investigated so that a stable representation of shape can be separated from the changing coordinate system of the camera. For discrete point features under orthographic projection, this has been achieved in the recent work of Tomasi and Kanade [89]. Second, appropriate image representations for matching images of surface patches and their boundaries under coordinate transformations need to be derived. Use of discrete features like points and lines leads to only a sparse reconstruction. Interpolation of this sparse 3D data to obtain surfaces is a hard ill-posed problem. Moreover, in a sequence of images, the motion of significant discrete structural features and the deformation of intensity patches both embody information about 3D shape and motion.

In a recent work on stereo correspondence and surface reconstruction, Jones and Malik [46, 47] used outputs of linear derivative filters to derive surface orientation and location. Images can be represented using outputs of oriented filters at a num-

ber of spatial scales applied all over each image. These outputs implicitly encode the spatial frequency and orientation information available in an image. A fruitful area of research can be the derivation of surface structure over time through warping and matching these filter outputs. Alternatively, moments of intensity regions can be utilized [59]. For instance, computation of the affine transformation using moments and invariant axes has been shown in [54]. These image representations, when combined with suitable motion and shape representations, might provide some handle on the challenging problem of motion-based description of a scene in terms of smooth surfaces while preserving discontinuities by demanding weak continuity constraints [13]. Imposition of controlled continuity surface smoothness constraints on the 3D information inferred from motion data will avoid the heuristic continuity constraints currently used for the computation of optical flow.

Certain issues in shallow structure reconstruction need further exploration. First, how can the transition from a shallow to a non-shallow model for an approaching object be handled. The need here is to dynamically estimate the modeling error in affine description of a structure being tracked. Second, more extensive testing in a variety of imaging scenarios is required to further establish the robustness, and pinpoint the areas of brittleness for the algorithm.

Finally, at an abstract level, the ideas developed in the shallow structure work and the rotational trajectories work could be combined. The affine constraint provides a basis for describing scene structure and motion in the small. Looking for possible groupings amongst affine describable patterns by describing smooth trajectories in the parameter space could be useful for segmenting multiple motions and separating rigidity from non-rigidity.

A P P E N D I X A

SOLUTION FOR THE ROTATIONAL TRAJECTORY WHEN THE AXIS PASSES THROUGH THE ORIGIN

When the rotational axis passes through the origin, the matrix which represents an image conic can be written as:

$$[M_{exp}] = [d^2[I] - (d^2 + k^2)[bb^T]] \quad (A.1)$$

This matrix can be normalized by d , assuming it is non-zero, and can be written in terms of the ratio k/d (which is recoverable). The solution for the special case when d is zero follows trivially from the non-zero case. The normalized matrix is:

$$[M_{exp}] = [[I] - (1 + \frac{k^2}{d^2})[bb^T]] \quad (A.2)$$

The eigenvalue-eigenvector pairs for this matrix are:

$$\lambda_1 = -(\frac{k}{d})^2 \quad \lambda_2 = 1 \quad \lambda_3 = 1 \quad (A.3)$$

$$n_1 = b \quad n_2 = n_2 \quad n_3 = n_3$$

where n_2 and n_3 are any two independent vectors in a plane normal to b . Note that two eigenvalues are identical and the third one is of a different sign and magnitude. In the more general case where the rotation axis does not intersect the origin this redundancy of the eigenvalues does not occur.

In order to recover the 3D parameters of the trajectory from the image data, we compute the eigenvalues of the matrix $[M_{com}]$ derived from the conic fit algorithm.

If two of the eigenvalues are same and the third one is different in magnitude and of opposite sign, this implies that the trajectory is of a point rotating around an axis passing through the origin. Let the eigenvalues of $[M_{com}]$ be λ_1 , and $\lambda_2 = \lambda_3$. Since $[M_{com}] = [M_{exp}]$ (after adjustment of the scale factor α), these eigenvalues can be identified with the eigenvalues of $[M_{exp}]$ calculated above. Then,

$$\frac{k}{d} = \pm\sqrt{\lambda_1} \quad (\text{A.4})$$

and \mathbf{b} is the unit eigenvector for λ_1 . Without loss of generality, \mathbf{b} can be forced to lie in the hemisphere of positive z directions. Then a unique sign for $\frac{k}{d}$ can be determined by invoking the fact that the imaged 3D trajectory must lie in front of the camera. For the more general case, the resolution of this type of sign ambiguity is discussed in Chapter 4.

Hence, for this case there is a unique, closed-form solution for the circular trajectory in space given $[M_{com}]$, the conic section fit to the image point sequence.

APPENDIX B

CONIC CURVE DESCRIPTION

A plane conic curve can be written in an implicit algebraic form as follows:

$$f(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (\text{B.1})$$

or equivalently,

$$f(x, y) = \mathbf{z}^T Q \mathbf{z} + \mathbf{z}^T \mathbf{h} + F = 0 \quad (\text{B.2})$$

where,

$$Q = \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} D \\ E \end{bmatrix}$$

This is a homogeneous equation in six variables. It constrains its six defining parameters only up to a scale factor. An advantage of representing a conic in this form is that the particular type of conic emerges out of the parameters obtained through a best fit. A negative *discriminant*, $B^2 - 4AC$, implies that the conic is an ellipse, a positive one implies that it is a hyperbola and an identically zero discriminant is a parabolic form. Degenerate forms like the linear and circular are special instances of parabolic and elliptical forms, respectively. One requirement for a curve-fitting routine is that the parameters of the fit be covariant with respect to rotational and translational transformations of the coordinate system [6]. That is, if we rotate and translate the image coordinate system through a rotation R and a translation T which

transforms each point z into z' as $z = Rz' + T$, then Q , h and F should transform as follows:

$$Q' = R^T Q R$$

$$h' = 2R^T Q T + R h$$

$$F' = T^T Q T + T^T h + F$$

In other words, if the fitting algorithm computes Q , h and F as the conic parameters in the original coordinate system, then in the transformed coordinate system, the resultant parameters should be Q' , h' and F' . One way to achieve this is to use the Euclidean distance between a point and the corresponding curve as an error measure which, when minimized over the conic parameters, leads to the optimum parameters. But the distance of a point to a conic is expressible exactly as a quartic equation [65] which leads to a complex optimization function. So approximate distance measures are used.

B.1 Algebraic Distance Measure

One simple error measure for conic fitting is the algebraic distance measure. It is assumed that $|f(x_i, y_i)|$ is an approximation to the distance measure. Then, $\sum_i f^2(x_i, y_i)$ is minimized over the whole set of sponsoring points in the six-dimensional parameter space. Of course, a constraint on the six defining parameters is imposed to fix the scale. The algebraic measure $f^2(x_i, y_i)$ is a scalar, and is invariant to the coordinate transformations. Consequently, the variation of the constraining norm determines if the fitting algorithm for this algebraic distance measure is covariant with the coordinate transformations.

The magnitude of the six-dimensional parameter vector can be constrained to be unity. Then the optimization problem becomes

$$\min_{\mathbf{p}} \mathbf{p}^T S \mathbf{p} \quad \text{subject to} \quad \mathbf{p}^T \mathbf{p} = 1 \quad (\text{B.3})$$

where,

$$\begin{aligned} \mathbf{p} &= [A \ B \ C \ D \ E \ F]^T \text{ is the six-dimensional parameter vector,} \\ \mathbf{m}_i &= [x_i^2 \ x_i y_i \ y_i^2 \ x_i \ y_i \ 1]^T \text{ is the image measurement vector, and} \\ S &= \sum_i \mathbf{m}_i \mathbf{m}_i^T \text{ is the scatter matrix, each term of which is the dyadic} \\ &\quad \text{product of a measurement vector.} \end{aligned}$$

The solution to this minimization problem [41] is the unit-norm eigenvector corresponding to the smallest eigenvalue of S . Unfortunately, this simple scheme does not satisfy the covariance requirement. The unit-norm condition on the parameter vector is not covariant with the coordinate transformations.

Bookstein [16] suggests the use of inherently covariant measures of a conic as constraining norms. It is well known that the forms $A + C$ and $B^2 - 4AC$ are covariant under the Euclidean group. The following lowest order positive definite invariant can be formed from these quantities: $(A + C)^2 + (B^2 - 4AC)/2 = A^2 + B^2/2 + C^2$. Bookstein suggests setting the value of this norm to 2, leading to the following optimization problem:

$$\min_{\mathbf{p}} \mathbf{p}^T S \mathbf{p} \quad \text{subject to} \quad \mathbf{p}^T N \mathbf{p} = 1 \quad (\text{B.4})$$

where S is the scatter matrix as before and $N = \text{diag}(1, 1/2, 1, 0, 0, 0)$.

Bookstein solves this in closed-form by partitioning the matrices S and N and the vector \mathbf{p} . Let,

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \quad N = \begin{bmatrix} N1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$N1 = \text{diag}(1, 1/2, 1) \quad \mathbf{p} = [\mathbf{p}_1^T \quad \mathbf{p}_2^T]^T$$

where the matrices S and N are partitioned as 3-by-3 sub-matrices and the vector \mathbf{p} is partitioned into 3-by-1 vectors. Thus, we have the following problem:

$$\min_{\mathbf{p}} \mathbf{p}^T S \mathbf{p} = \mathbf{p}_1^T S_{11} \mathbf{p}_1 + 2\mathbf{p}_1^T S_{12} \mathbf{p}_2 + \mathbf{p}_2^T S_{22} \mathbf{p}_2$$

$$\text{subject to} \quad \mathbf{p}_1^T N_1 \mathbf{p}_1 = 2$$

For any fixed \mathbf{p}_1 , $\mathbf{p}^T S \mathbf{p}$ is minimal when,

$$\mathbf{p}_2 = -S_{22}^{-1} S_{12} \mathbf{p}_1 \quad (\text{B.5})$$

This implies that

$$\mathbf{p}^T S \mathbf{p} = \mathbf{p}_1^T (S_{11} - S_{12} S_{22}^{-1} S_{21}) \mathbf{p}_1 = \mathbf{p}_1^T S_{112} \mathbf{p}_1 \quad (\text{B.6})$$

This is to be minimized with $\mathbf{p}_1^T N_1 \mathbf{p}_1 = 2$. The solution to this is the eigenvector of matrix $N_1^{-1} S_{112}$, corresponding to that eigenvalue which minimizes the fit error.

B.2 First Order Distance Measure

One problem, in general, with the algebraic distance measure examined above is that it underestimates the distance to a conic in high curvature regions of the conic [74]. The level curves of the algebraic distance measure, $f(x, y)$, are closely packed in low-curvature regions and sparsely packed in high-curvature ones as illustrated in Figure B.1. So, when there is considerable noise in the data, Bookstein's algorithm, based on $f(x, y)$, may result in a bad fit as it tries to place the input set of points in high curvature regions of the conic.

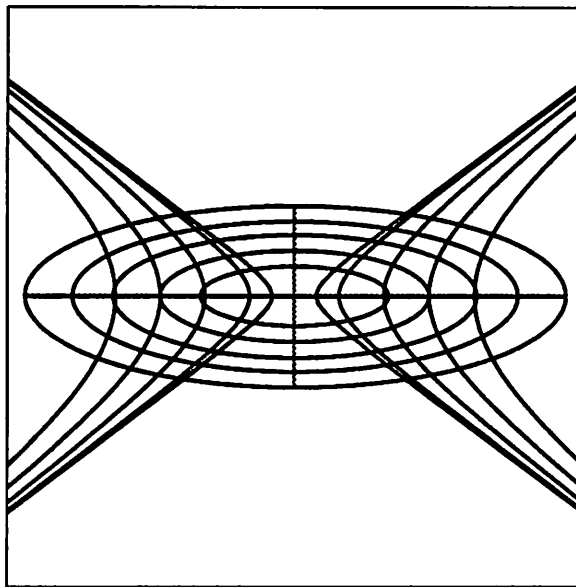


Figure B.1: Hyperbolic and elliptic level curves.

An improvement over the algebraic distance measure is a first order distance measure. Assuming that the points in the input data lie close to a level curve of $f(x, y)$ with level 0, it can be assumed that $\nabla f(x_i, y_i)$ is a good approximation to the gradient at (x_i, y_i) even when (x_i, y_i) does not exactly lie on $f(x, y) = 0$. Given this observation, a first order approximation to the distance of a point from the conic is $|f(x_i, y_i)|/|\nabla f(x_i, y_i)|$. We can use the sum of squares of these distances over all the points i as an error measure to be minimized by the fitting algorithm. Note that no explicit constraint on the fit parameters need be imposed in this method as the scale factor gets factored out by taking the ratio of the two terms. Now the unconstrained minimization problem is:

$$\min_{(A,B,C,D,E,F)} \sum_i (f(x_i, y_i)/|\nabla f(x_i, y_i)|)^2 \quad (\text{B.7})$$

where $f(x, y)$ is as in Equation (B.1) and

$$\begin{aligned}\nabla f(x, y) &= \frac{\partial f(x, y)}{\partial x} \mathbf{i} + \frac{\partial f(x, y)}{\partial y} \mathbf{j} \\ &= (2Ax + By + D)\mathbf{i} + (2Cy + Bx + E)\mathbf{j}\end{aligned}\tag{B.8}$$

Starting with the solution computed by Bookstein's algorithm as an initial guess, this minimization can be carried out by using any unconstrained non-linear iterative optimization technique. We have used the BFGS-DFP quasi-Newton routine from *Numerical Recipes in C* [95].

BIBLIOGRAPHY

- [1] ADIV, G. Determining 3D motion and structure from optical flows generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7, 4 (1985), 384-401.
- [2] ADIV, G. *Interpreting Optical Flow*. PhD thesis, University of Massachusetts at Amherst, MA, 1985. COINS TR 85-35.
- [3] ADIV, G. Inherent ambiguities in recovering 3D information from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 5 (1989), 477-489.
- [4] ADIV, G., AND RISEMAN, E. Recovery of 3D motion and structure from image correspondences using a directional confidence measure. Tech. Rep. COINS TR 88-105, University of Massachusetts at Amherst, MA, 1988.
- [5] AGGARWAL, J. K., AND NANDHAKUMAR, N. On the computation of motion from sequences of images - A review. Tech. Rep. TR-88-2-47, University of Texas at Austin, 1988.
- [6] AGIN, G. J. Fitting ellipses and general second-order curves. Tech. rep., The Robotics Institute, Carnegie-Mellon University, 1981.
- [7] ANANDAN, P. *Measuring Visual Motion from Image Sequences*. PhD thesis, University of Massachusetts at Amherst, MA, 1987. COINS TR 87-21.
- [8] ANANDAN, P. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision* 2, 3 (1989), 283-310.
- [9] ASADA, M., AND TSUJI, S. Representation of three-dimensional motion in dynamic scenes. *Computer Vision Graphics and Image Processing* 21 (1983), 118-144.
- [10] BAR-SHALOM, Y., AND FORTMANN, T. E. *Tracking and Data Association*. Academic Press, 1988.
- [11] BARRON, J. A survey of approaches for determining optic flow, environmental layout and egomotion. Tech. Rep. RBCV-TR-84-5, University of Toronto, 1984.

- [12] BLACK, M. J., AND ANANDAN, P. Robust dynamic motion estimation over time. In *Proc. Computer Vision and Pattern Recognition Conference* (1991), pp. 296-302.
- [13] BLAKE, A., AND ZISSERMAN, A. *Visual Reconstruction*. The MIT Press, Cambridge, MA, 1987.
- [14] BOLDT, M., WEISS, R., AND RISEMAN, E. Token-based extraction of straight lines. *IEEE Transactions on Systems Man and Cybernetics* 19, 6 (1989), 1581-1594.
- [15] BOLLES, R. C., BAKER, H. H., AND MARIMONT, D. H. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision* 1, 1 (1987), 7-55.
- [16] BOOKSTEIN, F. L. Fitting conic sections to scattered data. *Computer Graphics and Image Processing* 9 (1979), 56-71.
- [17] BRAS-MEHLMAN, E. L., SCHMITT, M., FAUGERAS, O. D., AND BOISSONNAT, J. D. How the Delaunay triangulation can be used for representing stereo data. In *Proc. 2nd Intl. Conf. on Computer Vision* (1988), pp. 54-63.
- [18] BROIDA, T. J. *Estimating the Kinematics and Structure of a Moving Object from a Sequence of Images*. PhD thesis, University of Southern California, Los Angeles, CA, 1987.
- [19] BROIDA, T. J., AND CHELLAPPA, R. Experiments and uniqueness results on object structure and kinematics from a sequence of monocular images. In *Proc. IEEE Wkshp. on Visual Motion* (1989), pp. 21-30.
- [20] BROLIO, J., DRAPER, B., BEVERIDGE, J. R., AND HANSON, A. ISR: A database for symbolic processing in computer vision. *IEEE Computer* 22, 12 (1989), 22-30.
- [21] BRUSS, A. R., AND HORN, B. K. P. Passive navigation. *Computer Vision Graphics and Image Processing* 21, 1 (1983), 3-20.
- [22] BURT, P. J., BERGEN, J. R., HINGORANI, R., KOLCZYNSKI, R., LEE, W. A., LEUNG, A., LUBIN, J., AND SHYVASTER, H. Object tracking with a moving camera. In *Proc. IEEE Wkshp. on Visual Motion* (1989), pp. 2-12.
- [23] BURT, P. J., HINGORANI, R., AND KOLCZYNSKI, R. Mechanisms for isolating component patterns in sequential analysis of multiple motion. In *Proc. IEEE Wkshp. on Visual Motion* (1991), pp. 187-193.
- [24] CHEN, D. S. A data-driven intermediate level feature extraction algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 7 (1989), 749-758.

- [25] CLOCKSIN, W. F. Perception of surface slant and edge labels from optical flow : A computational approach. *Perception 9* (1980), 253-269.
- [26] COLLINS, R. T., AND WEISS, R. An efficient and accurate method for computing vanishing points. In *Proc. Topical Meeting of the Optical Society of America on Image Understanding* (1989), pp. 92-94.
- [27] CROWLEY, J. L., STELMASZYK, P., AND DISCOURS, C. Measuring image flow by tracking edge-lines. In *Proc. 2nd Intl. Conf. on Computer Vision* (1988), pp. 658-664.
- [28] CUI, N., WENG, J., AND COHEN, P. Extended structure and motion analysis from monocular image sequences. In *Proc. 3rd Intl. Conf. on Computer Vision* (1990), pp. 222-229.
- [29] DANILIDIS, K., AND NAGEL, H. H. Analytical results on error sensitivity of motion estimation from two views. In *Proc. 1st European Conference on Computer Vision* (1990), pp. 199-208.
- [30] DERICHE, R., AND FAUGERAS, O. Tracking line segments. In *Proc. 1st European Conference on Computer Vision* (1990), pp. 259-268.
- [31] DEWILDE, P., AND DEPRETTERE, E. F. Singular value decomposition, an introduction. In *SVD and Signal Processing : Algorithms, Applications and Architectures*, D. Ed. F, Ed. Elsevier Science Publishing Company, Inc., NY, 1988, pp. 3-42.
- [32] FAUGERAS, O. D., AND LUSTMAN, F. Let us suppose the world is piece-wise planar. In *Proc. The Third International Symposium on Robotics Research* (1987).
- [33] FAUGERAS, O. D., LUSTMAN, F., AND TOSCANI, G. Motion and structure from motion from point and line matches. In *IEEE First International Conference on Computer Vision* (1987), pp. 25-34.
- [34] FENNEMA, C. L., AND THOMPSON, W. B. Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing 9* (1979), 301-315.
- [35] FORSYTH, D., MUNDY, J. L., ZISSERMAN, A., AND BROWN, C. M. Projectively invariant representations using implicit algebraic curves. In *Proc. 1st European Conference on Computer Vision* (1990), pp. 427-436.
- [36] FRANZEN, W. O. *Structure from Chronogeneous Motion*. PhD thesis, University of Southern California, Los Angeles, CA, 1991.
- [37] GELB, A. *Applied Optimal Estimation*. The MIT Press, Cambridge, MA, 1986.

- [38] GILL, P. E., MURRAY, W., AND WRIGHT, M. H. *Practical Optimization*. Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [39] HEEGER, D. J., AND JEPSON, A. D. Subspace methods for recovering rigid motion I: Algorithm and implementation. Tech. Rep. RBCV-TR-90-35, University of Toronto, 1990.
- [40] HILDRETH, E. C. *The Measurement of Visual Motion*. The MIT Press, Cambridge, MA, 1984.
- [41] HORN, B. K. P. Relative orientation. In *Proc. DARPA Image Understanding Workshop* (1988), pp. 826-837.
- [42] HORN, B. K. P. Recovering baseline and orientation from essential matrix. Internal Report, 1990.
- [43] HORN, B. K. P. Relative orientation. *International Journal of Computer Vision* 4, 1 (1990), 59-78.
- [44] HUTTENLOCHER, D. P., AND ULLMAN, S. Recognizing solid objects by alignment. In *Proc. Computer Vision and Pattern Recognition Conference* (1989), pp. 1114-1124.
- [45] JAENICKE, R. A. Structure from limited motion of complex objects. In *Proc. IEEE Wkshp. on Visual Motion* (1989), pp. 256-263.
- [46] JONES, D. G., AND MALIK, J. A computational framework for determining stereo correspondence from a set of linear spatial filters. Tech. Rep. UCB/CSD 91/655, University of California, Berkeley, CA, 1991.
- [47] JONES, D. G., AND MALIK, J. Determining three-dimensional shape from orientation and spatial frequency disparities II - using corresponding image patches. Tech. Rep. UCB/CSD 91/657, University of California, Berkeley, CA, 1991.
- [48] KANADE, T., AND KENDER, J. R. Mapping image properties into shape constraints: Skewed symmetry, affine-transformable patterns, and the shape-from-texture paradigm. In *Human and Machine Vision*, J. B. et al, Ed. Academic Press, NY, 1983, pp. 237-257.
- [49] KOENDERINK, J. J., AND VAN DOORN, A. J. Local structure of movement parallax of the plane. *Journal of the Optical Society of America A* 66 (1976), 717-723.
- [50] KUMAR, R., AND HANSON, A. Determination of camera location and orientation. In *Proc. IEEE Wkshp. on Interpretation of 3D Scenes* (1989), pp. 52-60.

- [51] KUMAR, R., AND HANSON, A. Sensitivity of the pose refinement problem to accurate estimation of camera parameters. In *Proc. 3rd Intl. Conf. on Computer Vision* (1990), pp. 365–369.
- [52] LAMDAN, Y., SCHWARTZ, J. T., AND WOLFSON, H. J. Object recognition by affine invariant matching. In *Proc. Computer Vision and Pattern Recognition Conference* (1988), pp. 335–344.
- [53] LECLERC, Y. G. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision* 3, 1 (1989), 73–102.
- [54] LEE, M. Recovering the affine transformation of images by using moments and invariant axes. In *Proc. Topical Meeting of the Optical Society of America on Image Understanding* (1991).
- [55] LENZ, R. K., AND TSAI, R. Y. Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, 5 (1988), 713–719.
- [56] LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature* 293 (1981), 133–135.
- [57] LONGUET-HIGGINS, H. C., AND PRAZDNY, K. The interpretation of a moving retinal image. In *Proc. Royal Society of London B* (1980), pp. 385–397.
- [58] MAHALANOBIS, P. C. On the generalized distance in statistics. *Proceedings of the National Institute of Science, India* 12 (1936), 49–55.
- [59] MANMATHA, R. Draft Dissertation Proposal, Univ. of Massachusetts., 1992.
- [60] MARIMONT, D. H. *Inferring Spatial Structure from Feature Correspondence*. PhD thesis, Stanford University, Stanford, CA, 1986.
- [61] MATTHIES, L., AND KANADE, T. The cycle of uncertainty and constraint in robot perception. In *Proc. The Fourth International Symposium on Robotics Research* (1988), pp. 327–335.
- [62] MATTHIES, L., SZELISKI, R., AND KANADE, T. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision* 3 (1989), 181–208.
- [63] MEHRA, R. K. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control* 15, 2 (1970), 175–184.
- [64] MURRAY, D. W., AND BUXTON, B. F. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9, 2 (1987), 220–228.

- [65] NALWA, V. S., AND PAUCHON, E. Edgel-aggregation and edge-description. In *Proc. The Eighth International Conference on Pattern Recognition* (1986), pp. 604-609.
- [66] NELSON, R. C., AND ALOIMONOS, J. Towards qualitative vision: Using flow field divergence for obstacle avoidance in visual navigation. In *Proc. 2nd Intl. Conf. on Computer Vision* (1988), pp. 188-196.
- [67] OLIENSIS, J., AND THOMAS, J. I. Incorporating motion error in multi-frame structure from motion. In *Proc. IEEE Wkshp. on Visual Motion* (1991), pp. 8-13.
- [68] PAPANIKOLOPOULOS, N., KHOSLA, P. K., AND KANADE, T. Vision and control techniques for robotic visual tracking. In *Proc. IEEE Conf. on Robotics and Automation* (1991), pp. 857-864.
- [69] PORRILL, J. Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter. *Image and Vision Computing* 8 (1990), 37-41.
- [70] RANGARAJAN, K., AND SHAH, M. Establishing motion correspondence. In *Proc. Computer Vision and Pattern Recognition Conference* (1991), pp. 103-108.
- [71] REHG, J. M., AND WITKIN, A. P. Visual tracking with deformation models. In *Proc. IEEE Conf. on Robotics and Automation* (1991), pp. 844-849.
- [72] RIEGER, J. H., AND LAWTON, D. T. Processing differential image motion. *Journal of the Optical Society of America A* 2, 2 (1985), 354-360.
- [73] ROBERTS, K. S. A new representation for a line. In *Proc. Computer Vision and Pattern Recognition Conference* (1988), pp. 635-640.
- [74] SAMPSON, P. D. Fitting conic sections to very scattered data. *Computer Graphics and Image Processing* (1982), 97-108.
- [75] SAWHNEY, H. S., AND HANSON, A. R. Comparative results of some motion algorithms on real image sequences. In *Proc. DARPA Image Understanding Workshop* (1990).
- [76] SAWHNEY, H. S., AND HANSON, A. R. Identification and 3D description of 'shallow' environmental structure in a sequence of images. In *Proc. Computer Vision and Pattern Recognition Conference* (1991), pp. 179-186.
- [77] SAWHNEY, H. S., AND OLIENSIS, J. Description and interpretation of rotational motion from images trajectories. In *Proc. DARPA Image Understanding Workshop* (1989), pp. 992-1003.

- [78] SAWHNEY, H. S., AND OLIENSIS, J. Description and interpretation of rotational motion from images trajectories. Tech. Rep. COINS TR 89-90, University of Massachusetts at Amherst, MA, 1989.
- [79] SAWHNEY, H. S., OLIENSIS, J., AND HANSON, A. R. Description and reconstruction from images trajectories of rotational motion. In *Proc. 3rd Intl. Conf. on Computer Vision* (1990), pp. 494-498.
- [80] SETHI, S. K., AND JAIN, R. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9, 1 (1987), 56-73.
- [81] SHARIAT, H. *The Motion Problem: How to Use More than Two Frames*. PhD thesis, University of Southern California, Los Angeles, CA, 1986.
- [82] SPETSAKIS, M. E., AND ALOIMONOS, J. Optimal computing of structure from motion using point correspondences in two frames. In *Proc. 2nd Intl. Conf. on Computer Vision* (1988), pp. 449-453.
- [83] SPOERRI, A., AND ULLMAN, S. The early detection of motion boundaries. In *Proc. 1st Intl. Conf. on Computer Vision* (1987), pp. 209-218.
- [84] STEVENS, K. A. Computation of locally parallel structure. *Biological Cybernetics* 29 (1978), 19-28.
- [85] STRANG, G. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, MA, 1986.
- [86] THOMPSON, D., AND MUNDY, J. Three-dimensional model matching from an unconstrained viewpoint. In *Proc. IEEE Conf. on Robotics and Automation* (1987), pp. 208-220.
- [87] THOMPSON, W. B., MUTCH, K. M., AND BERZINS, V. A. Dynamic occlusion analysis in optical flow fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1985), 374-383.
- [88] TODD, J. Visual information about rigid and non-rigid motion : A geometric analysis. *Journal of Experimental Psychology : Human Perception and Performance* 8, 2 (1982), 238-252.
- [89] TOMASI, C., AND KANADE, T. Shape and motion from image streams: A factorization method; 2. point features in 3D motion. Tech. Rep. CMU-CS-91-105, Carnegie Mellon University, 1991.
- [90] TSAI, R. Y., AND HUANG, T. S. Uniqueness and estimation of 3D motion parameters and surface structures of rigid objects. In *Image Understanding 1984*, W. Richards and S. Ullman, Eds. Ablex Corporation, NJ, 1984, pp. 135-171.

- [91] WEBB, J. A., AND AGGARWAL, J. K. Structure from motion of rigid and jointed objects. *Artificial Intelligence* 19 (1982), 107-130.
- [92] WENG, J., AHUJA, N., AND HUANG, T. S. Optimal motion and structure estimation. In *Proc. Computer Vision and Pattern Recognition Conference* (1989), pp. 144-152.
- [93] WENG, J., HUANG, T. S., AND AHUJA, N. 3D motion estimation, understanding and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9, 3 (1987), 370-389.
- [94] WENG, J., HUANG, T. S., AND AHUJA, N. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 5 (1989), 451-476.
- [95] W.H.PRESS, B.P.FLANNERY, S.A.TEUKOLSKY, AND W.T.VETTERLING. *Numerical Recipes in C*. Cambridge University Press, 1986.
- [96] WILLIAMS, L. R., AND HANSON, A. R. Translating optical flow into token matches and depth from looming. In *Proc. 2nd Intl. Conf. on Computer Vision* (1988), pp. 441-448.
- [97] WILLIAMS, L. R., AND HANSON, A. R. Perceptual organization of subjective contours. In *Proc. 3rd Intl. Conf. on Computer Vision* (1990), pp. 133-137.
- [98] YASUMOTO, Y., AND MEDIONI, G. Robust estimation of three-dimensional motion parameters from a sequence of image frames using regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 4 (1986), 464-471.
- [99] ZHANG, Z., AND FAUGERAS, O. D. Tracking and grouping 3D line segments. In *Proc. 3rd Intl. Conf. on Computer Vision* (1990), pp. 577-580.
- [100] ZHUANG, X., HUANG, T. S., AND HARALICK, R. M. Two-view motion analysis: A unified algorithm. *Journal of the Optical Society of America A* 3 (1986), 1492-1500.