



PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

**Image Sequence Analysis: Motion and Structure Estimation
with Transitory Sequences and Recognition of Hand Signs.**

By

Yuntao Cui

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science Department

1996

ABSTRACT

Image Sequence Analysis: Motion and Structure Estimation with Transitory Sequences and Recognition of Hand Signs.

By

Yuntao Cui

This thesis reports two pieces of work related to image sequence analysis. First, we investigate the problem for a calibrated stereo camera system traveling in an unknown environment to automatically build a 3D range map of the scene. Due to the dynamic sensing process, the obtained image sequence is *transitory* by our definition in that no component of the scene is visible through the entire sequence. We show that the integration of a transitory sequence has properties that are very different from those of a non-transitory one. Two representations, world-centered (WC) and camera-centered (CC), behave very differently with a transitory sequence. The asymptotical error properties derived indicate that one representation is significantly superior to the other, depending on whether one needs camera-centered or world-centered estimates. Based on these properties, we introduce an efficient “cross-frame” estimation technique for the CC representation. For the WC representation, our analysis indicates that a good technique should be based on global pose of the camera instead of inter-

frame motions. In addition to testing with synthetic data, rigorous experiments have been conducted on a real-image sequence taken by a fully calibrated camera system. The experimental results have demonstrated good accuracy.

Secondly, we focus on the problem of recognizing hand signs from intensity image sequences. We present a new framework to recognize hand signs from intensity image sequences. The framework has two major components: segmentation and recognition. A prediction-and-verification scheme using attention images from multiple fixations is presented to segment hands from input images. A major advantage of this scheme is that it can handle a large number of different deformable and articulated objects presented in complex backgrounds. The scheme is also relatively efficient since the segmentation is guided by the past knowledge through a prediction-and-verification scheme. During the recognition, the system uses multiclass, multidimensional discriminant analysis to automatically select the most discriminating features for gesture classification. A recursive partition tree approximator is presented to do classification. The framework has been tested to recognize 28 different hand signs. The experimental results show that the system can achieve a 93% recognition rate for test sequences that have not been used in the training phase.

To my wife Yu Zhong and my parents

ACKNOWLEDGMENTS

I would like to acknowledge all the people who have assisted me throughout my studies at Michigan State University. I am particularly grateful to my adviser, Dr. John J. Weng, for his advice and encouragement. He provided me with numerous ideas, suggestions and comments, and has been very beneficial to my development as a researcher. Dr. Anil K. Jain provided critiques and encouragement that were very useful for this work. I am grateful to Dr. George Stockman for offering an excellent course on deformable objects and image databases which I took and learned so much from it. My gratitude also goes to my other committee members, Drs. Dennis Gilliland and Richard Hallgren, for their critiques of my research and for their personal support.

My sincere thanks go to my dear wife Yu Zhong for her patience, encouragement, support, and understanding. She has provided many insightful suggestions and useful references.

In my work, I used several pieces of software developed by Dan Swets and Shaoyun Chen, my fellow labmates working with Dr. Weng. Thanks also go to Yu Zhong, Kal Rayes, Doug Neal, and Valerie Bolster for making themselves available for the

experiments.

While working in the PRIP Laboratory, I also met a number of knowledgeable people from different parts of the world. Jinlong Chen, Shaoyun Chen, Yao Chen, Chitra Dorai, Hansye Dulimarta, Lin Hong, Sally Howden, Qian Huang, Wey Shiuan Hwang, Kalle Karu, Yonghong Li, Jianchang Mao, Karissa Elizabeth Miller, Aniati Murni, Tim Newman, Sharathcha Pankanti, Nalini Ratha, Dan Swets, Oivind Due Trier, Patchrawat Uthaisombut, Aditya Vailaya, Gang Wang, Marilyn Wulfekuhler, and Yu Zhong, are among them.

I greatly appreciate Dr. Ray R. Brodeur and Ms. Maria Eppler. They have edited and proofread the thesis and made it more readable.

TABLE OF CONTENTS

LIST OF FIGURES	x
------------------------	----------

LIST OF TABLES	xiii
-----------------------	-------------

1 Introduction	1
1.1 Motion and Structure Estimation from Image Sequences	4
1.1.1 Feature-based and flow-based motion estimation approaches	4
1.1.2 Monocular and stereo image sequence	6
1.1.3 Perspective and parallel projection	7
1.1.4 Two-view vs. multiple-view	8
1.2 Integration of Transitory Image Sequences	9
1.3 Interpretation of Human Action	12
1.4 Hand Sign Recognition	14
2 Feature-Based Approaches for Motion Estimation	19
2.1 General Problem	20
2.2 Motion Model	22
2.3 2D-to-2D Correspondence	24
2.4 3D-to-3D Correspondences	29
2.4.1 Point features	29
2.4.2 Motion from stereo images	30
2.5 2D-to-3D Correspondences	34
2.6 Motion from Long Sequences	38
2.6.1 Kalman filter	38
2.6.2 Other approaches	40
3 Transitory Image Sequences, Asymptotic Properties, and Estimation of Motion and Structure	44
3.1 Basic Concepts	45
3.2 Asymptotic Error Properties of Integrations	49
3.2.1 World-centered representation	50
3.2.2 Camera-centered representation	56
3.2.3 The tightness of the error rates	59
3.3 Methods and Algorithms	68
3.3.1 Cross-frame approach with CC representation	69
3.3.2 World-centered representation	75
3.4 Experiments	77

3.4.1	Simulation Data	78
3.4.2	Experiments with a real setup	83
3.5	Conclusions	95
4	Hand Sign Recognition	97
4.1	Glove-Based Systems	98
4.1.1	Glove devices	98
4.1.2	Interpreting hand sign with gloved-based devices	99
4.2	Vision-Based Approach	100
4.2.1	Segmentation	100
4.2.2	Recognition	103
5	Overview of the Approach	110
5.1	Time as a Dimension	111
5.2	Recognition of Spatiotemporal Pattern	112
6	Hand Segmentation from Attention Images Based on Eigen-subspace Learning	116
6.1	Learning	118
6.1.1	Karhunen-Loeve projection	118
6.1.2	Simulated fovea image	121
6.2	Segmentation	122
6.2.1	Reconstruction	122
6.2.2	Dynamic Deformation	124
6.3	Experiments	127
6.3.1	Training	127
6.3.2	Testing	128
6.4	Conclusions	129
7	Hand Segmentation Using a Prediction-and-Verification Scheme	132
7.1	Valid Segmentation	135
7.1.1	Karhunen-Loeve projection	137
7.1.2	Approximation as function interpolation	138
7.1.3	Valid segmentation	140
7.2	Predication for Valid Segmentation	140
7.2.1	Overview	140
7.2.2	Organization of attention images from fixations	143
7.2.3	Prediction as querying the training set	145
7.3	Experiments	153
7.3.1	Training	153
7.3.2	Hand segmentation	155
7.4	Conclusions and Future Work	159

8	View-Based Hand Sign Recognition from Intensity Image Sequences	160
8.1	Nearest Neighbor Approximator in the MEF Space	163
8.2	Approximation Using Recursive Partition Tree in the MDF Space	164
8.2.1	The Most Discriminating Features (MDF)	164
8.2.2	Curse of dimensionality and the DKL projection	167
8.2.3	Recursive partition tree	167
8.3	Convergence of the Approximators	171
8.4	k Nearest Neighbors	175
8.5	Experimental Results	175
8.5.1	Results of the nearest neighbor approximator in the MEF space	176
8.5.2	Results of the recursive partition tree approximator in the MDF space	178
8.5.3	k nearest neighbors	181
8.5.4	Experiments related to MDF	181
8.6	Conclusions	185
9	Summary and Future Work	187
9.1	Summary	187
9.1.1	Integration of transitory image sequences	187
9.1.2	Hand Sign Recognition	189
9.2	Future Work	191
9.2.1	Integration of transitory image sequences	191
9.2.2	Hand sign recognition	192

LIST OF FIGURES

1.1	Perspective projection.	7
1.2	The twenty eight different signs used in the experiment. (1) sign of “angry”; (2) “any”; (3) “boy”; (4) “yes”; (5) “cute”; (6) “fine”; (7) “funny”; (8) “girl”; (9) “happy”; (10) “hi”; (11) “high”; (12) “hot”; (13) “later”; (14) “low”; (15) “no”; (16) “nothing”; (17) “of course”; (18) “ok”; (19) “parent”; (20) “pepper”; (21) “smart”; (22) “sour”; (23) “strange”; (24) “sweet”; (25) “thank you”; (26) “thirsty”; (27) “welcome”; (28) “wrong” (Bornstein and Saulnier 1989).	16
2.1	Basic perspective geometry. Lower case letters refer to coordinates in the object space and upper case letters refer to coordinates on the image plane. f is the focal length.	20
2.2	Constraint on the motion parameters derived from point matches.	26
2.3	Stereo triangulation and elongated uncertainty shape. The true point can lie anywhere inside the shaded uncertainty region.	31
2.4	Imaging geometry for 2D-to-3D correspondence problem. A point p in coordinate system xyz is imaged at location P' on the image plane which is specified in coordinate system $x'y'z'$	34
3.1	Two systems of reference: (a) world-centered and (b) camera-centered.	46
3.2	Transitory and non-transitory sequences. (a) non-transitory. (b) simple transitory. (c) general transitory.	48
3.3	Using cross-frame motions to integrate many views. Each elongated ellipse indicates the uncertainty in 3-D point position transformed from a single previous stereo view to the current view. The integrated uncertainty is greatly reduced using the multiple cross-frame motions instead of interframe motions.	71
3.4	Simulation environment, where 7000mm distance is covered by the 31 frames.	78
3.5	Camera global pose error versus time. The CC representation on the left column and WC on the right. (a) and (e): Error in rotation matrix. (b) and (f): Error in translation vector. (c) and (g): Error in xy -component of the translation vector. (d) and (h): Error in z -component of the translation vector.	80
3.6	Structure error versus time. The CC representation on the left column and WC on the right. (a) and (e): Global structure error. (b) and (f): xy -component of the global structure error. (c) and (g): Local structure error. (d) and (h): xy -component of the local structure error.	82

3.7	The robot and a few stereo frames in the 151-frame sequence. (a) Robot. (b) Left image of frame 0. (c) Left image of frame 50. (d) Left image of frame 100. (e) Left image of frame 150.	84
3.8	The tracking record of the feature points through the 151 frames in the sequence. If a point k is successfully tracked from frame i to frame j , a vertical line is shown at point number k from frame i to frame j . (Due to the limit of the printer resolution, lines are merged in the plot.)	86
3.9	Stereo matching and temporal matching-and-tracking. (a) An example of stereo matching (frame 0). (b) An example of temporal matching and tracking (frame 24 to 69). A needle is draw from the feature point to its position in the target frame. Due to camera vergence, the orientation of the needles in (a) is correct.	88
3.10	Sample test points on one frame. Each cross shows the location of a test point	89
3.11	Camera rotation error versus time. (a) and (d): Yaw error. (b) and (e): Roll error. (c) and (f): Pitch error.	91
3.12	Camera position error versus time. (a) and (e): Camera position (y - and z -components). (b) and (f): Position error (x -component). (c) and (g): Position error (y -component). (d) and (h): Position error (z -component).	92
3.13	Structure error. (a) and (d): Global structure error. (b) and (e): Global structure error (x - and y -components). (c) and (f): Global structure error (z -component).	93
3.14	Reconstructed 3D surface integrated from many partial views in the sequence, shown with original intensity viewed from an arbitrary direction.	94
3.15	Weighted texture mapping from pixel to the reconstructed surface.	95
5.1	The sign “no” and its image sequence representation. (a) The sign of “no”, snap middle finger, index, and thumb together. (b) The sequence representation of the sign “no”.	111
5.2	The three-stage framework for spatiotemporal event recognition	113
6.1	A 2D illustration of Karhunen-Loeve projection.	120
6.2	An illustration of the reconstruction.	124
6.3	Twenty five different hand shapes used in the experiments.	127
6.4	The SFIs of the training samples.	127
6.5	Segmentation results of the fovea images which are not used in the training. (a) Input fovea images. (b) The results of the reconstruction. We also blur the results of the reconstruction. (c) The results of applying masks to the original input fovea images. The masks were derived using the dynamic spring network model to the reconstructed images. (d) The SFIs of the images in (c). (e) The contours of the nearest neighbors are superimposed onto the input images. (f) Segmentation results.	131
7.1	An illustration of two level fixations of an input hand image.	133
7.2	The illustration of constructing attention images.	136

139		
7.4	Overview of the segmentation scheme.	142
7.5	A 2-D illustration of a hierarchical quasi-Voronoi diagram and the corresponding recursive partition tree. (a) In partition, the label indicates the center of a cell. The label of the child to which its parent's center belongs is not shown due to the lack of space. (b) The corresponding recursive partition tree.	144
7.6	A 2D illustration of nearest neighbor query theorems.	147
7.7	A representative subset of hand shapes used in the experiment.	153
7.8	The attention images from 19 mechanical fixations of a training sample.	155
7.9	Eight sample sequences. From top to bottom, they represent the signs "happy", "hot", "nothing", "parent", "pepper", "smart", "welcome", and "yes".	156
7.10	Results of motion-based attention are shown using dark rectangular windows.	157
7.11	The results of the segmentation are shown after masking off the background.	158
8.1	A 2D illustration of the most discriminating features (MDF). The MDF is projection along z_1 . The MEF along y_1 can not separate the two subclasses.	164
8.2	A 2-D illustration of a hierarchical Voronoi diagram and the corresponding recursive partition tree. (a) The partition, where the circles and rectangles indicate the samples used to create the Voronoi diagram. (b) The corresponding recursive partition tree.	169
8.3	Illustration of components in the Mixture Distance in a 3D original space.	171
8.4	Top ten MEF's	177
8.5	Performance of the nearest neighbor approximator in the MEF space. The performance is given as a function of the number of MEF's used.	178
8.6	Performance of the two different nearest neighbor query approaches: linear vs. quasi-Voronoi diagram.	179
8.7	Top ten MDF's	180
8.8	The difference between MEF and MDF in representing samples. (a) Samples represented in the subspace spanned by the first two MEFs. (b) Samples represented in the subspace spanned by the first two MDFs. The numbers in the plot are the class labels of the samples.	183
8.9	Two sample sequences of signs "of course" (a) and "wrong" (b).	184
8.10	The difference between the MEF and MDF. (a) Reconstruction based on the first MDF. (b) Reconstruction based on 95% MEFs.	185

LIST OF TABLES

3.1	Asymtotic rate for error covariance matrix in integration	58
3.2	Some Data for The Real Setup	89
3.3	Average Image Plane Residual	90
6.1	Summary of the segmentation results	129
7.1	The list of fixation scale and position	154
7.2	Summary of the experimental data	158
8.1	The number of MEF's vs. the variation ratio	177
8.2	Summary of the experimental data for RPTA	180
8.3	Summary of the experimental data for k NN	182

Chapter 1

Introduction

The goal of a machine vision system is to recover useful information about a scene from its two-dimensional projections. Early computer vision systems were concerned primarily with static scenes. Recently, some computer vision systems are being designed to analyze dynamic scenes for different applications.

The input to a dynamic scene analysis system is a sequence of image frames taken from moving objects. The camera used to acquire the image sequence may also be in motion. Mounting evidence, accumulated over the past century and especially of late, leaves no doubt that motion is indeed a fundamental visual dimension like color and stereopsis [138]. The functional benefits of human image motion processing are listed as follows.

- *Depth reconstruction.* The motion of objects can reveal their shapes. Ullman [181] has an ingenious demonstration of this point: a set of dots is projected onto a screen. When the dots are stationary, an observer sees merely a screenful

of randomly distributed dots. When they move, however, the display springs to life, and the observer sees two cylinders rotating in opposite directions.

- *Image segmentation.* Related to the problem of depth measurement is the need to parse the complex pattern of illumination in the optic array into different physical objects and to distinguish “figure” from “ground”. Motion is eminently suited for this job because of the mathematical relation between neighboring points in the optical velocity field at the edges of objects. Points which are well within the boundaries of a visual object, for example, generally have the same or very similar velocities between neighboring points whereas this is not necessarily the case at the boundary of an object in an image.
- *Motion in a proprioceptive sense.* Gibson [72] suggested that visual motion was one of the primary sources of information for the moving organism to know about its own motion in relation to its environment. Early work by Lee and Anderson [112] measured the importance of optical flow information for postural control. Standing infants could be made to lose their balance and fall as a result of movement in the surrounding visual environment. Environmental visual motion also destabilized the posture of adults, suggesting that visual motion information can override information obtained from stretch receptors in the limbs and gravity receptors in the inner ear. Visual motion can also lead to a profound sensation of self-motion, either as a rotation about a vertical axis or as a horizontal or vertical translation [27].
- *Stimulus to drive eye movement.* Ever since the important experiments of Rash-

bass [155], it has been recognized that the oculomotor pursuit system is driven by a velocity signal. Rashbass simultaneously stepped a visual target in one direction and initiated a constant velocity motion in the opposite direction. Thus position information (in the form of a step) was pitted against velocity information (in the form of a ramp). Surprisingly, the eye movement system responded separately to each, generating a smooth eye movement in response to velocity ramp and an oppositely directed saccade in response to the position step. Later work has suggested some additional contribution from a visual position encoding system [148], but theoretical discussions of the oculomotor pursuit system still hinge directly on the notion that the visual system can indeed read velocity.

Broadly speaking, there are two groups of scientists studying motion. The first group is studying human/animal vision with the goal of understanding the operation of biological vision systems including their limitations and diversity. The second group of scientists includes computer scientists and engineers conducting research in computer vision with the objective of developing vision system. A vision system with the ability to navigate, recognize and track objects, and estimate their speed and direction is the ultimate goal of the research. The proposed work falls into the latter category. We report two pieces of works in this thesis. One is the motion and structure estimation with transitory sequences and the other one is the recognition of hand signs.

1.1 Motion and Structure Estimation from Image Sequences

Motion estimation is a long-standing problem in computer vision, with many different applications. A proper formulation of the motion estimation problem requires the identification of several characteristics of the image acquisition process and the scene, including: the camera model, the number of cameras, the number of views.

Two distinct approaches have been developed for the computation of motion from image sequences [3], namely, the feature based approach and the flow based approach. These two approaches can be further classified based on: 1) The number of images in each frame: monocular or stereo; 2) Projection model: perspective or parallel projection; 3) The number of views the system is designed to handle: two-view or multiple-view.

1.1.1 Feature-based and flow-based motion estimation approaches

The feature-based motion estimation from image sequences can be divided into two steps. The first step is to establish feature correspondences for all pairs of consecutive image frames in a sequence. The features are a set of relatively sparse, but highly discriminatory, two-dimensional geometric shapes in the images corresponding to three-dimensional object features in the scene, such as corners, occluding boundaries of surfaces, and boundaries demarcating changes in surface reflectivity. Such

points, lines and/or curves are extracted from each image. The inter-frame correspondence is then established between these features. The second step is to estimate motion parameters. Constraints are formulated based on assumptions such as rigid body motion, i.e., the three dimensional distance between two features on a rigid body remains the same after object/camera motion. Such constraints usually result in a system of nonlinear equations. The observed displacement of the 2-D image features are used to solve these equations leading ultimately to the computation of motion parameters of the objects in the scene.

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow can arise from relative motion of objects and the viewer. Consequently, optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement. The flow-based approach is based on computing the optical flow or the two-dimensional field of instantaneous velocities of brightness values (gray levels) in the image plane. Instead of considering temporal changes in image brightness values in computing the optical flow field, it is possible to also consider temporal changes in values that are the result of applying various local operators such as contrast, entropy, and spatial derivatives to the image brightness values. In either case, a relatively dense flow field is estimated, usually at every pixel in the image. The optical flow is then used in conjunction with added constraints or information regarding the scene to compute the actual three-dimensional relative velocities between scene objects and camera.

There is as yet no clearcut evidence recommending one general approach over the other. The feature-based approach allows a relatively large interframe motion. The

disadvantages of this approach include: 1) The features are expensive to extract; 2) The correspondence problem is nontrivial; 3) Features are relatively sparse. On the contrary, the flow-based approach does not have the above disadvantages, however, 1) it is expensive to compute due to the sheer number of pixels; 2) it can only handle relatively small interframe motions; 3) it can not integrate long sequences effectively due to the absence of anchors (features).

1.1.2 Monocular and stereo image sequence

The monocular sequence is a set of images obtained by a single camera. Given a sequence of monocular images of the scene, the motion and structure of an object can be estimated by both feature-based approach as well as flow-based approach. The solutions for structure and motion remain ambiguous with respect to the absolute value of the distance between the camera and the scene. In other words, structure and motion parameters are unique only up to a scaling factor. The use of stereoscopy can provide this additional parameter to uniquely determine depth and hence the absolute values for the structure and motion parameters.

The fusion of stereo and motion may be effected with different objectives in mind. Stereoscopic processing may be used to aid motion recovery, or conversely, motion analysis may be used to establish feature correspondence in stereo image pairs. The fusion of these two processing modules in human and other biological visual systems has been detected via neurobiological and psychophysiological investigation [146].

1.1.3 Perspective and parallel projection

In general, projections transform points in a coordinate system of dimension n into points in a coordinate system of dimension less than n . *Perspective projection* is the fundamental model for the transformation wrought by our eyes, by cameras, or by numerous other imaging devices. To a first-order approximation, these devices act like a pinhole camera in that the image results from projecting scene points through a single point onto an image plane (see Figure 1.1).

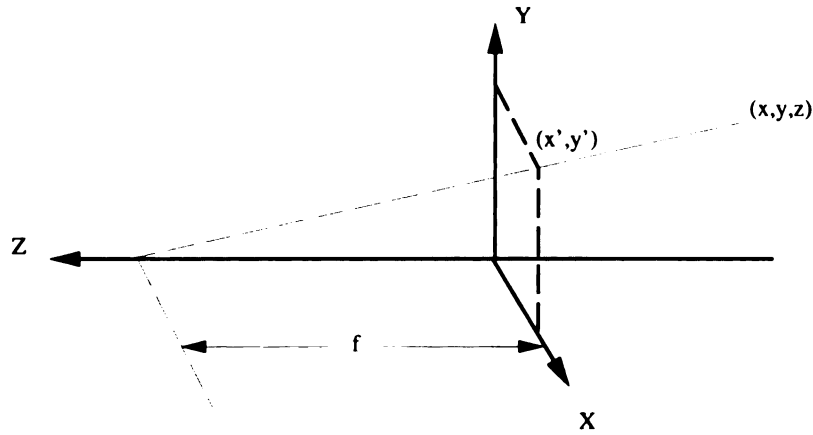


Figure 1.1: Perspective projection.

The mathematical equations for a perspective projection are given by:

$$\begin{aligned}\frac{x}{f - z} &= \frac{x'}{f} \\ \frac{y}{f - z} &= \frac{y'}{f}\end{aligned}\tag{1.1}$$

where f is the *focal length*.

The perspective transformation yields *parallel projection* as a special case when the viewpoint is the *point at infinity* in the z direction. Then all objects are projected

onto the viewing plane with no distortion of their x and y coordinates. Although perspective projection is a more precise model, it has its problems. First, when camera motion is small, effects of camera rotation and translation can be confused with each other. Second, the shape is computed as relative depth, and it is very sensitive to noise. These difficulties are especially magnified when the objects are distant from the camera relative to their size. So, there are still cases where the parallel projection model is used [8, 178].

1.1.4 Two-view vs. multiple-view

The presence of noise in the image leads to inaccuracy in the resulting estimates of the object motion parameters. In order to combat the noise, the algorithm needs to exploit redundancy in the available data to improve the accuracy of the solution. The over determination of the estimation equations is achieved by using a larger number of feature points in a two-view case. The alternative is to use a larger number of image frames. If the interframe motion remains constant, then an arbitrary number of frames can be used since the number of unknown (motion parameters) is not a function of the number of the image frames. If the interframe motion is not constant, which is probably more realistic in the navigation problem where the robot may constantly need to change direction to avoid obstacles, the problem of preventing global motion deviation is more important since the error accumulates from each inaccurate interframe motion.

1.2 Integration of Transitory Image Sequences

In the case of motion estimation from image sequences (multiple-view), there is a special case in which the image sequence is transitory. In this thesis, we propose a new approach to deal with the transitory image sequence. The proposed approach is a feature based method. Our method uses stereo images and the perspective projection model. Next, we define the transitory image sequence.

If a system needs to sense a large 3-D rigid scene which cannot be covered by a single view, the system may actively move and scan the scene [5]. For example, to automatically build a 3-D map of a floor in a building, a camera system moves from one room to the next on the floor. To obtain information about all the facets of a 3-D object, a camera system needs to actively circle around the object or let the object rotate. In general, during a dynamic sensing process, any component of the scene is visible only in a subsequence, and thus the resulting image sequence is *transitory*.

Issues with transitory nature of scene components have mostly not yet been investigated. Current most of the research deals with non-transitory image sequences, and successful improvements have been achieved in this type of fusion [28, 10, 111, 131, 164, 177]. Experiments for scene construction from transitory image sequence only started recently. In Cui *et al* [44], the relative accuracy was reported from a transitory image sequence, which indicated that the accuracy was not further reduced once incoming and exiting feature points are comparable. Tomasi and Kanade [178] conducted experiments with transitory image sequences and discussed how to expand the measurement matrix by filling in “hallucinated” projections. The

results showed that the object structure and camera pose constructed from transitory sequences “Ball” and “Hand” contained larger error than that from a non-transitory sequence “Hotel” [178].

Most questions related to the integration of transitory sequences are still open. Some of them are:

1. With a transitory image sequence, is it still reasonable to expect “the more images one has, the better the accuracy” as with a non-transitory sequence?
2. How should transitory image sequences be integrated?
3. What representation(s) should one use? World-centered (WC), camera-centered (CC) or some other representation?
4. What is the asymptotic error behavior from a transitory image sequence? In other words, how does the error in an estimated object structure and camera pose change with time (or frame number), for a good estimation method?
5. Are the asymptotic error rates for the method the lowest possible?
6. What kind of accuracy one can reach in a real setup with transitory image sequences?

The new contributions of our work includes

1. We show that from a transitory sequence it is inherently not possible to get better estimates with a longer sequence. The later a scene part comes into the sequence, generally the worse its global accuracy is compared to that in the first view.

2. We introduce different techniques for two different usages of the results: global and local (e.g., global corresponds to visual map generation and global pose determination while local corresponds to obstacle avoidance and object manipulation belong to the latter).
3. It is demonstrated that different representations result in very different stabilities. In general, WC is better for a global usage and CC is superior for a local usage.
4. We establish asymptotic error rates with respect to the number of frames, which indicates how the error estimate evolves with time as well as how to minimize the pace of error accumulation. Some concise expressions have been derived in terms of asymptotic error rate for different representations, processing methods, and image sequence types.
5. We establish that the asymptotic error rates are in fact the lowest possible based on the Cramér-Rao error bound.
6. In order to provide actual accuracy with a real system setup, careful experiments have been conducted with a fully calibrated camera system. The algorithm includes feature selection, stereo matching, temporal matching and tracking, 3-D structure integration, and motion and pose estimation.

1.3 Interpretation of Human Action

It has to be admitted that the human visual system is capable of carrying out a wide variety of image analysis and interpretation tasks with what appears to be utmost ease. You walk off the plane after a long boring travel, and sight your friend near the gate who is waving his hand to you. You lighten up a little after seeing his smiling face. There are three problems that your visual system solves: (i) it identifies the objects on the basis of their shapes; (ii) it detects their movement, if there is any; (iii) it recognizes the motion patterns, e.g., hand waving and smiling.

There has been significant interest in the human action recognition from time-sequential images. Model-based recognition consists of the recognition of objects or motions directly from the motion information extracted from the sequence of images. The knowledge about the object or motion is used to construct models that will serve in the recognition process. There are two main steps in this approach. The first consists of finding an appropriate representation for the objects or motions, from the motion cues of the sequence, and then organizing them into useful representations. The second step consists of the matching of some unknown input with a model.

The work on interpretation of human motion can be classified into three categories according to their applications, that is, interpretation of facial expressions, interpretation of hand gestures, and recognition of movements of other body parts.

- **Facial expression**

The human face has attracted much attention in several disciplines, including psychology, computer vision, and computer graphics. Psychophysical investi-

gations clearly indicate that faces are very special visual stimuli. Psychologists have studied various aspects of human face perception and recognition [30]. They have also examined facial expression - the result of a confluence of voluntary muscle articulations that deform the neutral face into an expressive face. The facial pose space is immense. The face is capable of generating on the order of 55,000 distinguishable facial expressions with about 30 semantic distinctions. Studies have identified six primary expressions that communicate anger, disgust, fear, happiness, sadness, and surprise in all cultures [58].

Automatic facial recognition had an early start in image understanding, but work on the problem has been sporadic over the years, evidently due to the difficulty of extracting meaningful information from facial images. Facial classification systems based on measurements derived from interactively selected fiducial points (eye and mouth corners, nose, top of head, etc.) go back to the early 1970's [99]. Recent works concentrate on the deformation of facial features [118, 175, 199].

- **Hand sign**

Humans use gestures in daily life as a means of communication (e.g., waving hands to say good bye to a friend, pointing to an object to bring someone's attention to the object etc). The best example of communication through gesture is given by sign language. American sign language (ASL) incorporates the entire English alphabet along with many gestures representing words and phrases [41], which permits people to exchange information in a nonverbal manner. It

was shown that requirements in precision and resolution for ASL are relatively low as compared to greyscale images [167].

- **Body movement**

Body movement is a broad term. There are several ways to view this task. The first one is to recognize the action performed by a person in a scene, among a database of human action models. The second way is to be able to recognize the different body parts like arms, legs, etc. throughout a sequence, using motion. The third way is to define motion as a sequence of object configurations or shapes through time. The knowledge of the shape and motion of an object, in this case, is used to guide the interpretation of an image sequence in order to analyze the motion between frames, to determine the most plausible configuration of the body or to recognize and label different parts of the body. This approach has been used mostly with humans, and sometimes is called the *tracking of human motion*.

The work presented in this thesis focuses on the hand sign recognition.

1.4 Hand Sign Recognition

The evolution of computer technology has seen a corresponding evolution of computer input and human-machine interaction technologies. At first, humans had to prepare punch cards and paper tapes. Later, the machine was capable of reading keyboards and providing “real time” feedback on tele-type terminals. Recently, ad-

vances in memory and computation technology have permitted machines to have two-dimensional pointing devices such as mice. The next step is to build machines which can interact with the human users in a natural way. That means that machines not only need to understand speech but also *gestures*. One of the most general definitions from the *Lexis* dictionary says that gestures are “movements of body parts, particularly the arms, the hands or the head conveying, or not conveying, meaning”. Among them, hand gestures are the most important, having different sign languages such as American Sign Language.

Since its first known dictionary was printed in 1856 [29], American Sign Language (ASL) is widely used in the deaf community as well as by handicapped people who are not deaf [22]. The general hand sign interpretation needs a broad range of contextual information, general knowledge, cultural background and linguistic capabilities, which are beyond the capabilities present-day computer. In our current research, we extract a set of hand gestures which have meaning in human communication. Twenty-eight different signs are extracted from [23], which are illustrated in Fig. 1.2. These hand signs have the following characteristics: 1) they represent a wide variation of hand shapes; 2) they include a wide variation of motion patterns; 3) these hand signs are performed by one hand; 4) recognition of these signs does not depend on contextual information. The gestures which require the hand to perform in a certain environment or point to a specific object are excluded.

In this thesis, we present a new vision-based framework which allows the computer to interact with users through hand signs. Vision-based analysis of hand signs is one of the most natural ways of constructing a human-computer gesture interface since it is

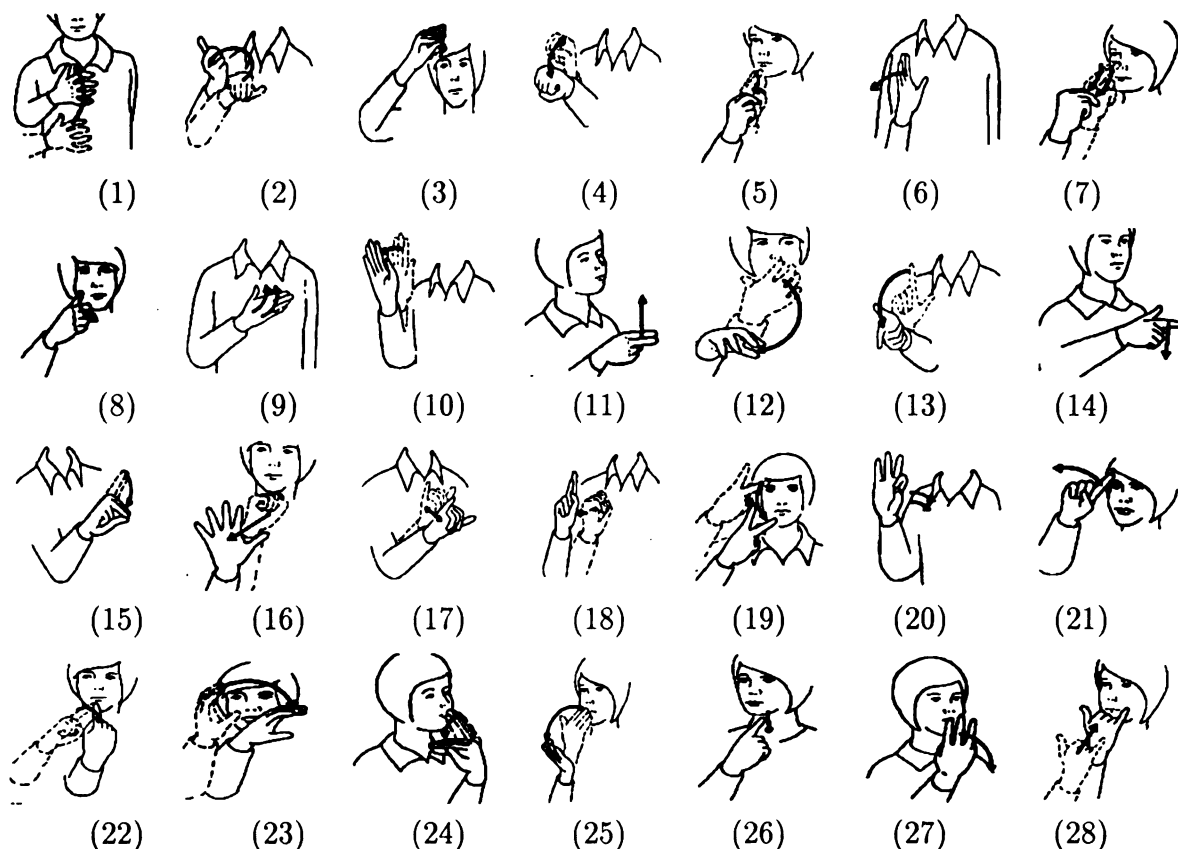


Figure 1.2: The twenty eight different signs used in the experiment. (1) sign of “angry”; (2) “any”; (3) “boy”; (4) “yes”; (5) “cute”; (6) “fine”; (7) “funny”; (8) “girl”; (9) “happy”; (10) “hi”; (11) “high”; (12) “hot”; (13) “later”; (14) “low”; (15) “no”; (16) “nothing”; (17) “of course”; (18) “ok”; (19) “parent”; (20) “pepper”; (21) “smart”; (22) “sour”; (23) “strange”; (24) “sweet”; (25) “thank you”; (26) “thirsty”; (27) “welcome”; (28) “wrong” (Bornstein and Saulnier 1989).

based on the major way humans perceive information about their surroundings. However, it is also the most difficult one to implement in a satisfactory manner because of the limitations in machine vision today. The approach faces two difficult problems: 1) segmentation of the moving hand from a sometimes complex background, and 2) analysis and recognition of hand motion. In order to avoid the segmentation problem, some of the existing systems rely on markers or marked gloves (e.g. [38, 54, 168]). The others simply assume a uniform background (e.g. [19, 46, 53]).

The patterns of hand motion appearing in our gesture vocabulary are extremely complicated. It can be a local deformation such as the change of the hand posture. It can also be a global motion which means the change of the hand position, or it involves both local and global motion. Many existing systems use simplified models to characterize hand motion such as 2D fingertip trajectory [54] and 3D joint angles [110]. These motion models are hand-crafted by algorithm designers. The system does not have the ability to learn models, neither can it improve or alter the model in case the model does not fit. This results in a *brittle* system, since a model cannot fit in all the cases in reality and typically the system cannot tell whether the model will fail given an unknown input. Another problem of the existing approaches is that they typically isolate temporal understanding from spatial understanding. For example, a system can probably tell something is moving, but cannot tell what is moving. Isolation of temporal understanding from spatial understanding makes it impossible to apply the system to unstructured (unknown) environments.

In this thesis, we present a new general framework to learn and recognize hand signs. The new contributions of this framework are listed as follows.

1. *Segmentation.* A prediction-and-verification scheme using attention images from multiple fixations is presented to segment hands from complex backgrounds. A major advantage of this scheme is that it can handle a large number of different deformable objects presented in complex backgrounds. The scheme is also relatively efficient since the segmentation is guided by the past knowledge through a prediction-and-verification scheme.

- *Prediction.* A hierarchical quasi-Voronoi diagram which organizes training attention images for prediction of the segmentation masks.
- *Verification.* A learning-based function approximation scheme to verify the segmentation result.

2. *Recognition.* We propose a new framework in which Motion understanding is tightly coupled with spatial recognition.

- *Automatic feature selection.* In the framework, the discriminant analysis is used to automatically select the most discriminating features (MDF) for recognition.
- *Inference and generalization.* In practice, the system is unable to learn unlimited number of samples. An interpolation scheme is introduced to generalize to other variations from a limited number of learned samples.

Chapter 2

Feature-Based Approaches for Motion Estimation

In this chapter, we review the motion estimation problem which deals with observing some features on the surface of an object in the environment at different points in time and using this information to derive the three dimensional motion and structure of these objects. Let us assume that we have one or more cameras that are moving continuously in a static environment and following some unknown trajectory. We will consider the images obtained at a number of time instants t_0, t_1, \dots, t_{n-1} and assume that we can extract from these n images a number of features and match them between the images. We are interested in the set of finite rigid displacements D_i that bring the camera from its position and orientation at time t_i to its new position and orientation at time t_{i+1} .

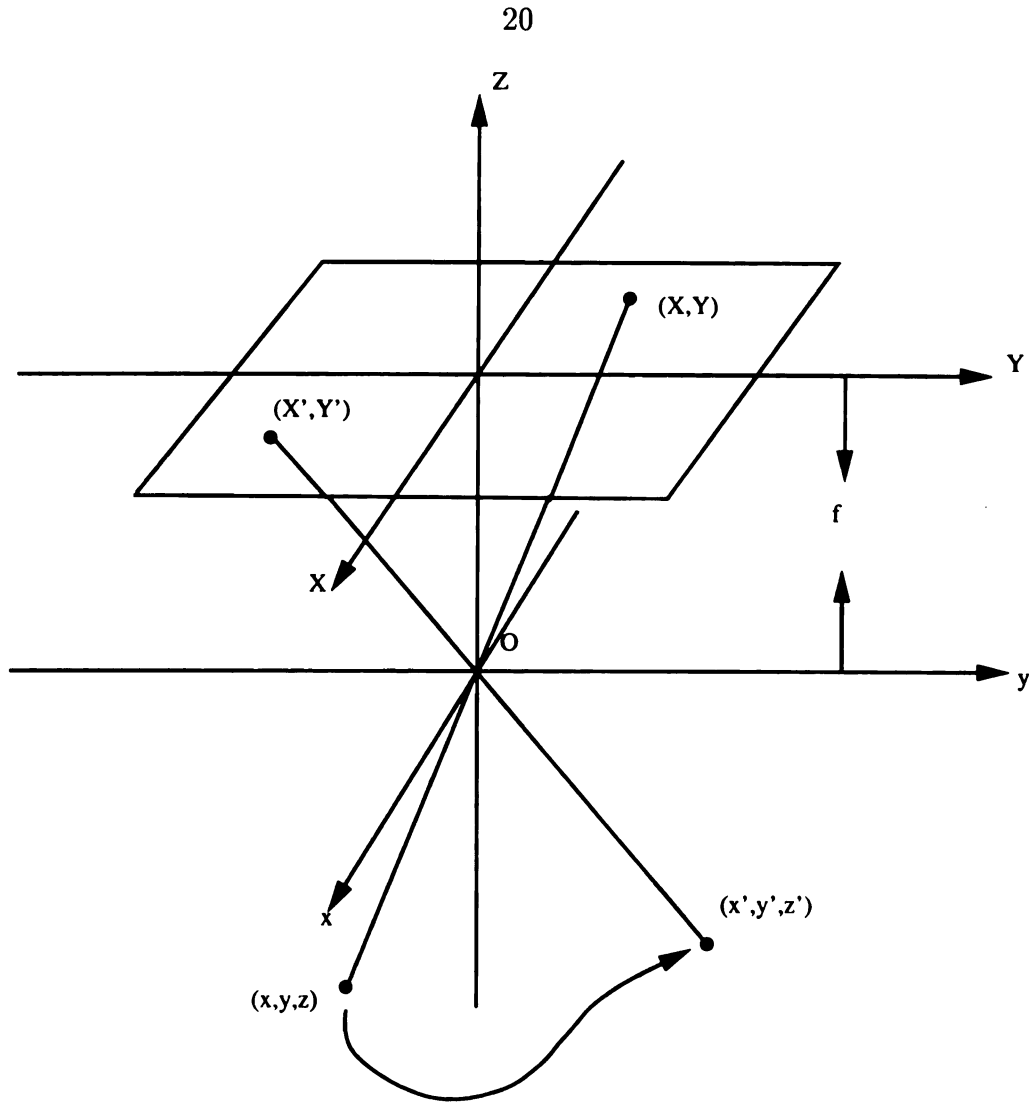


Figure 2.1: Basic perspective geometry. Lower case letters refer to coordinates in the object space and upper case letters refer to coordinates on the image plane. f is the focal length.

2.1 General Problem

The basic geometry of the two view case is sketched in Figure 2.1. The object-space coordinates are denoted by lowercase letters and the image-space coordinates are denoted by uppercase letters. The optical center of a pin-hole camera coincides with the origin of a cartesian coordinate system (xyz) and the positive z -axis is the direction of view. The image plane is located at a distance equal to the focal length f

(which is assumed to be unity) from o along the direction of view. Using a perspective projection model, a point $p = (x, y, z)$ on the surface of an object is projected at a point $P = (X, Y)$ on the image plane, where

$$\begin{aligned} X &= \frac{x}{z} \\ Y &= \frac{y}{z}. \end{aligned} \tag{2.1}$$

Given N corresponding features (points, lines, corners, conics etc.) on the same rigid object and our problem is to infer motion and structure of this rigid body with respect to the imaging system. In general, there are three different cases:

1. *2D-to-2D Correspondence*: Here, N corresponding features are specified on the 2D image planes either at two different times for a single camera or at the same instant of time but from two different cameras. In the former case, the problem is to determine 3D motion and structure of the rigid object and in the latter case, the problem is to determine the relative orientation and location of the two imaging cameras.
2. *3D-to-3D Correspondence*: We are given 3D locations of N corresponding features at two different times and we need to estimate the 3D motion of the rigid body. Thus the problem has application in either motion estimation using 3D information which can be obtained by stereo or other range-finding techniques.
3. *2D-to-3D Correspondence*: In this case, we are given correspondence of N features (f_i, f'_i) such that f_i are specified in three dimensions and f'_i are their

projection on the 2D image plane. The problem is to find location and orientation of the imaging camera. This has applications in either calibration of a single camera or passive navigation through known 3D landmarks.

2.2 Motion Model

Let the two views be taken at t_1 and t_2 , respectively. Consider a particular physical point on the surface of a rigid body in the scene. Let

(x, y, z) = object-space coordinates of the point at time t_1 ,

(x', y', z') = object-space coordinates of the point at time t_2 .

It is well known in kinematics that

$$\begin{aligned} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} &= R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \end{aligned} \quad (2.2)$$

where R is a 3×3 orthonormal matrix, i.e., $R^t R = R R^t = I$ (I is a 3×3 identity matrix) and $\det(R) = 1$, T is a 3×1 vector. Rotation can be specified in a number of equivalent ways. For example, R can be specified as three successive rotation around the x -, y -, and z -axis, by angles θ , ψ , and ϕ , respectively, and can be written as a

product of these three rotations

$$R = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}. \quad (2.3)$$

Alternatively, the rotation matrix can be expressed in terms of a rotation axis, $\hat{n} = [n_x, n_y, n_z]^t$, and a rotation angle, ϕ , about \hat{n} . The rotation matrix can be written as

$$R = \begin{bmatrix} (n_x^2 - 1)c + 1 & n_x n_z c - n_z s & n_x n_z c + n_y s \\ n_y n_x c + n_y s & (n_y^2 - 1)c + 1 & n_y n_z c - n_x s \\ n_z n_x c - n_y s & n_z n_y c + n_x s & (n_z^2 - 1)c + 1 \end{bmatrix} \quad (2.4)$$

where $c = (1 - \cos\phi)$ and $s = \sin\phi$.

A rotation around an axis with direction cosines (n_1, n_2, n_3) and rotation angle ϕ can also be represented by the unit quaternion

$$q = (s; l, m, n) = [\cos\frac{\phi}{2}; n_1 \sin\frac{\phi}{2}, n_2 \sin\frac{\phi}{2}, n_3 \sin\frac{\phi}{2}] \quad (2.5)$$

specifically

$$(0; p'_i) = q(0; p_i)q^* \quad (2.6)$$

where $*$ denotes complex conjugation. In terms of q , the rotation matrix becomes

$$R = \begin{bmatrix} s^2 + l^2 - m^2 - n^2 & 2(lm + sn) & 2(ln + sm) \\ 2(lm + sn) & s^2 - l^2 + m^2 - n^2 & 2(mn - sl) \\ 2(ln - sm) & 2(mn + sl) & s^2 - l^2 - m^2 + n^2 \end{bmatrix}. \quad (2.7)$$

2.3 2D-to-2D Correspondence

Consider the problem in which a point $p_i = (x_i, y_i, z_i)$ on a rigid body moves to a point $p'_i = (x'_i, y'_i, z'_i)$ with respect to a camera fixed coordinate system. Let the perspective projection of p_i be $P_i = (X_i, Y_i, 1)$ and that of p'_i be $P'_i = (X'_i, Y'_i, 1)$. Due to the rigid body motion, p_i and p'_i are related by

$$p'_i = Rp_i + T \quad (2.8)$$

where R and T are the rotation and translation respectively. Given N correspondences $(P_i, P'_i), i = 1, 2, \dots, N$, it is impossible to determine the magnitude of translation. If the rigid body were two times farther away from the image plane, but twice as big, and translated at twice the speed, we would get exactly the same two images. Therefore, the translation T and object-point ranges (z_i) can only be determined to within a global positive scale factor.

Roach and Aggarwal [158] proposed an algorithm that solves for motion parameters directly from nonlinear equations. The equations that relate the three-dimensional coordinates of a point (x, y, z) and its image plane coordinates (X, Y)

are

$$\begin{aligned} X &= F \frac{r_{11}(x - x_0) + r_{12}(y - y_0) + r_{13}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)} \\ Y &= F \frac{r_{21}(x - x_0) + r_{22}(y - y_0) + r_{23}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)}. \end{aligned} \quad (2.9)$$

where F is the focal length, (x_0, y_0, z_0) is the projection center and $r_{11}, r_{12}, \dots, r_{33}$ are functions of (θ, ψ, ϕ) , as shown in equation (2.3). Roach and Aggarwal showed that five points in two views are needed to recover these parameters. The five-points algorithm has a very long history [65]. The problem was solved in 1913 by Kruppa [109]. In [158], Roach and Aggarwal related the number of points to the number of equations available for the solution of 3-D coordinates and motion parameters as follows. The global coordinates of each point are known so the five points produce 15 variables. The camera position and orientation parameters $(x_0, y_0, z_0, \theta, \psi, \phi)$ in two views contribute another 12 variables yielding a total of 27 variables. Each 3-D point produces two projection equations thus forming a total of 20 nonlinear equations. To make the number of unknowns equal to the number of equations, the six camera parameters of the first view are set to be zero and the Z -component of one of the five points is set to be an arbitrary positive constant to fix the scaling factor. An iterative finite difference Levenberg-Marquardt algorithm was used to solve these nonlinear equations. Nonlinear equations generally have to be solved through iterative methods with an initial guess or through a global search. Searching is computationally expensive. Also the iterative methods may diverge or convert to a local minima.

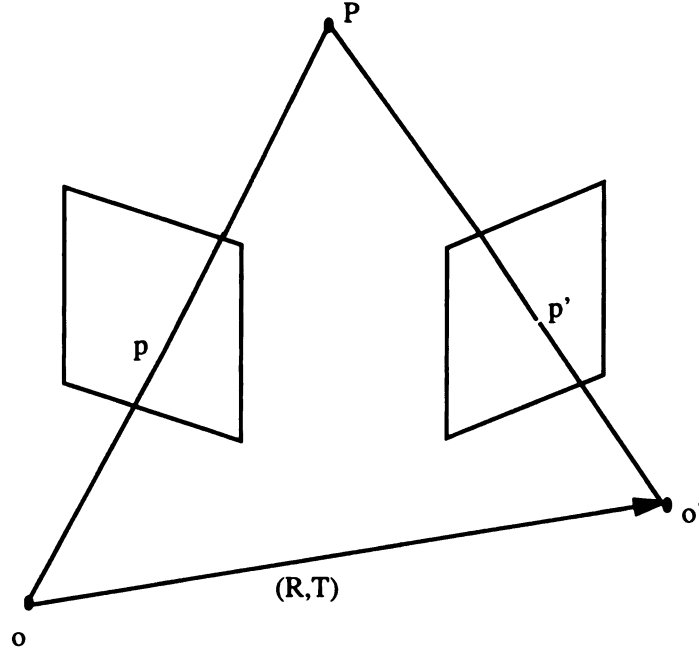


Figure 2.2: Constraint on the motion parameters derived from point matches.

Given eight or more point correspondences, a linear algorithm can be derived [120, 179]. As shown in Figure 2.2, the 3-D point P has images p and p' at two consecutive time instants. From the figure, it is clear that p and p' form a correspondence if and only if the three vectors PO , OO' and PO' are coplanar. The constraint can be written in the coordinate system of the first camera as

$$p' \cdot (T \times Rp) = 0. \quad (2.10)$$

Introduce the antisymmetric matrix G :

$$G = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \quad (2.11)$$

Matrix G is such that $Gx = t \times x$ for all vectors x . Let $E = GR$, equation (2.10) can be rewritten as

$$(p')^t E p = 0 \quad (2.12)$$

Dividing both sides of equation (2.12) by the positive quantity zz' (i.e., dividing p by z and p' by z') gives

$$\begin{bmatrix} X' & Y' & 1 \end{bmatrix} E \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = 0. \quad (2.13)$$

Equation (2.13) is linear and homogeneous in terms of nine unknowns of elements of E , $e_{i=1,2,\dots,9}$. Given N correspondences, we can write equation (2.13) in the form

$$B[e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^t = 0. \quad (2.14)$$

With eight point correspondences, if the rank $(B) = 8$, E can be uniquely determined to within a scale factor. Once E is determined, R and T can be obtained.

The existing linear algorithms essentially consider noise-free data. High sensitivity to noise is reported in [179] [63]. To handle noise, Yasumoto and Medioni [200] used a regularization approach. Under the assumption of small rotation angles, rotation matrix R is expressed by

$$R = \begin{bmatrix} 1 & -\Omega_z & \Omega_y \\ \Omega_z & 1 & -\Omega_x \\ -\Omega_y & \Omega_x & 1 \end{bmatrix} \quad (2.15)$$

where Ω_x , Ω_y , Ω_z denote the rotation angle around the x , y and z axis, respectively.

The objective function is

$$\sum_{i=1}^n [(\alpha_i - \hat{\alpha}_i)^2 + (\beta_i - \hat{\beta}_i)^2] + \lambda(\Omega_x^2 + \Omega_y^2 + \Omega_z^2) \quad (2.16)$$

where the first and second term are the square of the difference between the predicted and computed displacement, and the third term is the regularization function. A consequence of their approach is that the estimated motion parameters are biased towards nonrotational interpretations if the regularization factor is not zero. In their approach, the search for the global minimum of the objective function in motion parameter space is computationally expensive.

Weng *et. al.* [189] used least squares techniques to make use of the redundancy in the data to combat noise. The algorithm first solves for the essential matrix E . Then the motion parameters are obtained from E . Finally the spatial structure is derived from the motion parameters. All the steps of the algorithm use the redundancy in the data to combat noise. The linear equations in the linear algorithms [179] [120] [203] are converted to least squares minimization problems. Due to the coupling between different parameters in the motion vector, the large magnitude of translation is needed to obtain stable estimation of translation direction and structure of the scene.

If the rank $(B) < 8$, the above linear algorithms cannot be used. However, if rank $(B) = 5, 6$ or 7 , then the linear equations (2.14) can be solved along with the polynomial constraints on the components of matrix E [88] [90]. Specifically, E is equal to a skew-symmetric matrix post multiplied by a rotation matrix only if

$\{e_i\}_{i=1,2,\dots,9}$, satisfy the following three constraint equations. Let $\varepsilon_{i=1,2,3}$ be the i th row of E , then

$$\begin{aligned}\varepsilon_3 \cdot (\varepsilon_1 \times \varepsilon_2) &= 0 \\ (\|\varepsilon_2\|^2 + \|\varepsilon_3\|^2 - \|\varepsilon_1\|^2)(\varepsilon_2 \cdot \varepsilon_3) + 2(\varepsilon_1 \cdot \varepsilon_3) &= 0 \\ \|\varepsilon_3\|^4 &= (\|\varepsilon - 2\|^2 - \|\varepsilon\|^2)^2 + 4(\varepsilon_1 \cdot \varepsilon_2)^2.\end{aligned}\tag{2.17}$$

Thus we get three polynomial equations in $\{e_i\}_{i=1,2,\dots,9}$ of degree 3, 4, 4, respectively.

2.4 3D-to-3D Correspondences

2.4.1 Point features

Suppose we are given N corresponding points (p_i, p'_i) which obey the relationship of

$$p'_i = Rp_i + T\tag{2.18}$$

The problem is: given (2.18), find R and T . The (p_i, p'_i) are 3D coordinates of points on the surface of the rigid body in motion. It is well known that three noncollinear-point correspondences are necessary and sufficient to determine R and T uniquely.

Equation (2.18), when expanded represents three scalar equations in six unknown motion parameters. With three point correspondences, we will get nine nonlinear equations. Iterative methods can be used to obtain the ‘best’ fits of the six unknowns. However, it is possible to get stuck in the local minima.

Blostein and Huang [20] used linear algorithms by observing that equation (2.18) is linear in components of R and T . Given four correspondences, $(p_i, p'_i)_{i=1,2,3,4}$, we have the following linear equation:

$$\begin{bmatrix} p_{1x} & p_{1y} & p_{1z} & 1 \\ p_{2x} & p_{2y} & p_{2z} & 1 \\ p_{3x} & p_{3y} & p_{3z} & 1 \\ p_{4x} & p_{4y} & p_{4z} & 1 \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ t_1 \end{bmatrix} = \begin{bmatrix} p'_{1x} \\ p'_{2x} \\ p'_{3x} \\ p'_{4x} \end{bmatrix}. \quad (2.19)$$

Similar equations can be obtained to solve $(r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}, t_2, t_3)$. The linear method uses four points instead of the minimum of three required for uniqueness. To overcome the problem of supplying the linear method with this extra point correspondence, a “pseudo-correspondence” can be artificially constructed on the basis of rigidity of the body.

2.4.2 Motion from stereo images

A common method for determining the three-dimensional structure of the surrounding environment is through stereo vision, and in fact, the human stereo system is remarkably adept at this computation, under a wide variety of conditions. Stereo vision can be characterized by three steps as shown in Figure 2.3: 1) The point in one image corresponding to the projection of a point on a surface; 2) The point in the other image corresponding to the projection of the *same* surface point; 3) The difference in projection of the corresponding points is used, together with estimates

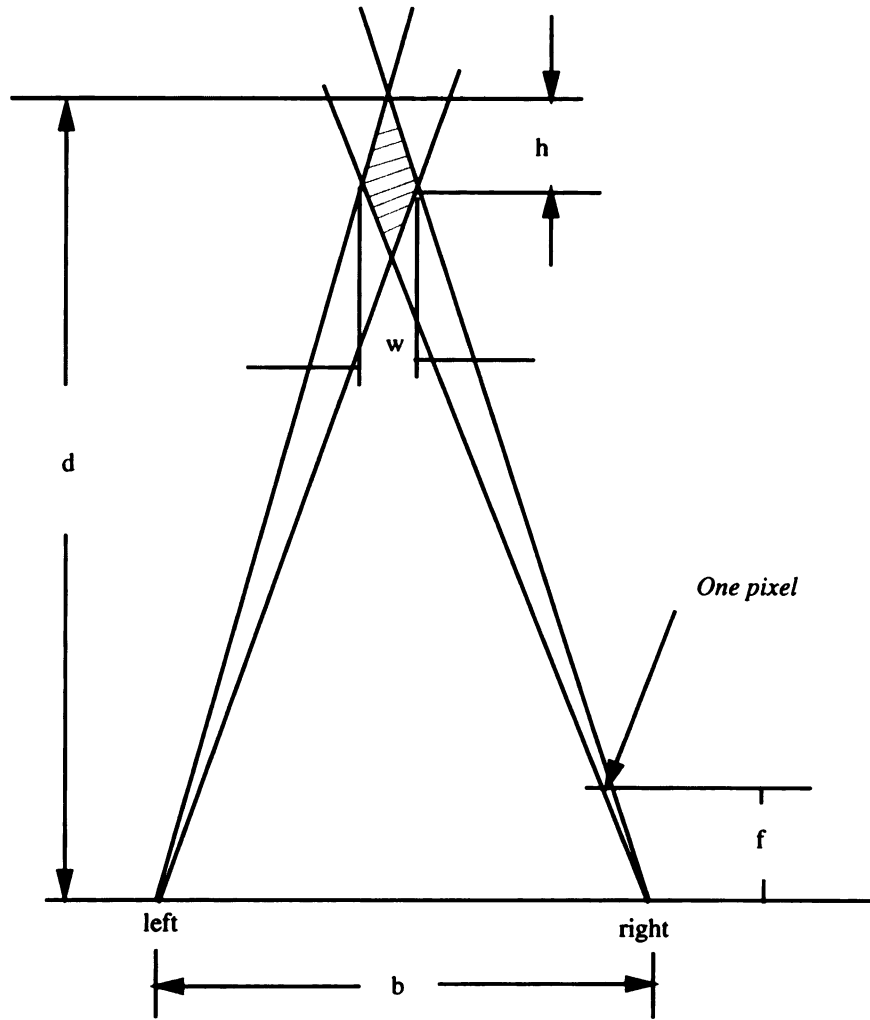


Figure 2.3: Stereo triangulation and elongated uncertainty shape. The true point can lie anywhere inside the shaded uncertainty region.

of the parameters of the imaging geometry to determine a measure of the distance to the surface point. Marr and Poggio have proposed a feature-point based model of human stereopsis [126]. A computer implementation of their algorithm was then developed and tested [78, 79].

In the problem of motion from stereo images, $(p_i, p'_i)_{i=1, \dots, N}$ are measured by stereo triangulation. Since the measurements are subject to error, one prefers to work with more than three point correspondences. In this case, R and T can be obtained as a

solution to the following least squares problem:

$$\min_{w.r.t. R, T} \left[\sum_{i=1}^N \|p'_i - (Rp_i + T)\|^2 \right] \quad (2.20)$$

subject to the constraint that R is a rotation matrix. where $\| \cdot \|$ represents the Euclidean norm. Such a constrained least square problem can be solved with linear procedures using quaternions [87] or by singular value decomposition [8].

One of the most important advantages of the closed-form solutions is that the corresponding algorithms are fast and the solutions are guaranteed. However, the least squares solution is not optimal in that it equally trusts all components with different reliabilities. In the three-dimensional position of a point determined by a typical stereo triangulation, the depth component is much less reliable than the lateral components, as shown in Figure 2.3. Matthies and Shafer [131] studied some related issues of error modeling in stereo navigation. They modeled the error of a three-dimensional point, constructed through stereo triangulation, by using a three-dimensional random vector with a Gaussian distribution (called an ellipsoidal model). Given a set of corresponding three-dimensional points $\{p_i\}$ before motion and $\{p'_i\}$ after motion, the interframe motion represented by a rotation matrix R and a translation vector T were determined to minimize

$$\sum_{i=1}^N (Rp_i + T - p'_i)^t V_i (Rp_i + T - p'_i) \quad (2.21)$$

where the weighting matrix V_i is the inverse of $R\Gamma_{x_i}R^{-1} + \Gamma_{x'_i}$ (Γ with a subscript

denotes the error covariance matrix of the variable represented by the subscript). A closed-form solution to this problem was not found. They iteratively minimized (2.21) using a least squares solution as an initial guess.

Kiang, *et. al.* [101] replaced the matrix V_i in (2.21) by a scalar w_i^2 . The distribution of error in the three-dimensional position of a point was simplified into an uncertainty line segment. From a least squares solution, a few iterations gave improved motion parameters. Moravec [134] has also used a simpler scalar weight, which is inversely proportional to the depth of the point. A scalar weight indiscriminately treats the uncertainties in different components. This implies that either reliable components are undertrusted or unreliable components are overtrusted. Furthermore, the correlation between errors in the components of a three-dimensional point cannot be taken into account by the scalar weight.

Weng *et. al.* [191] also used matrix-weighted least squares. They presented a closed-form approximate matrix-weighted least squares solution for motion parameters from three-dimensional point correspondences, which minimizes (2.21) with the weighting matrix V_i simplified so that it does not vary with the unknown rotation matrix R . The result of the algorithm approximates the optimal solution if the rotation is small. With large rotation, the result can be used as a better initial guess for optimal iterative approach.

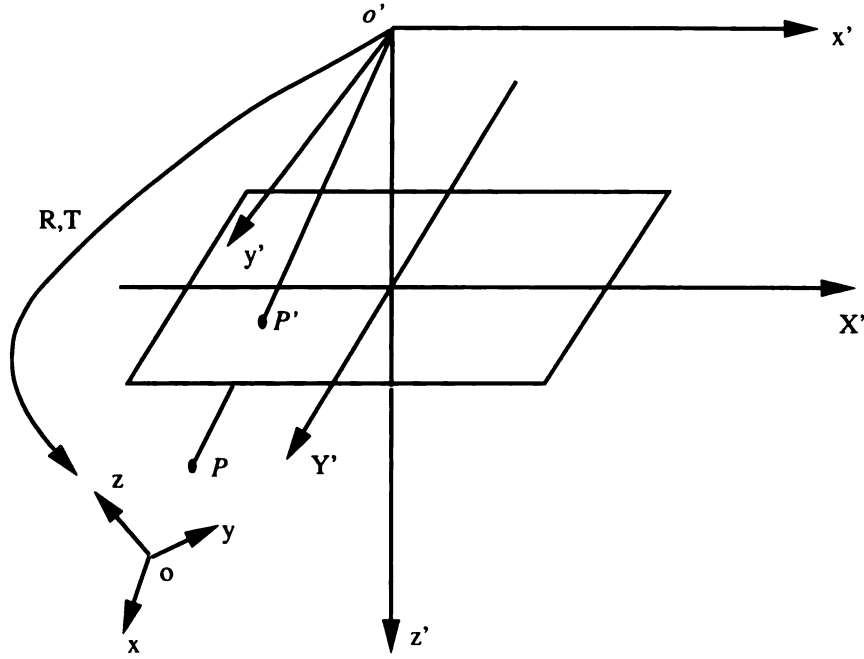


Figure 2.4: Imaging geometry for 2D-to-3D correspondence problem. A point p in coordinate system xyz is imaged at location P' on the image plane which is specified in coordinate system $o'x'y'z'$.

2.5 2D-to-3D Correspondences

This situation arises when each of the features is 3D and their corresponding features are 2D. If 3D locations of features are known along with their projection on the camera plane which is at unknown location, then the algorithms described in this section allow us to determine the attitude, i.e., the location and the orientation of the camera (known as the camera calibration problem).

As shown in Figure 2.4, consider two coordinate systems. xyz is a coordinate system in which the 3D point features are located. Thus p_i are points in this coordinate system with coordinates (x_i, y_i, z_i) . The camera is referenced to the other coordinate system $(o'x'y'z')$. Image coordinates on the camera plane are obtained by perspective projection and denoted by (X', Y') . Thus the image of point p_i is at P'_i .

whose coordinates are given by

$$\begin{aligned} X'_i &= \frac{x'_i}{z'_i} \\ Y'_i &= \frac{y'_i}{z'_i} \end{aligned} \quad (2.22)$$

Coordinate system xyz is obtained by a rotation R and translation T of the coordinate system $x'y'z'$ and the goal in camera calibration is to determine R and T , knowing the N point correspondences $(p_i, P'_i)_{i=1,\dots,N}$.

The 3D coordinates of p'_i are related to those of p_i by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T. \quad (2.23)$$

Combining equations (2.22) and (2.23), we get

$$\begin{aligned} X'_i &= \frac{r_{11}x_i + r_{12}y_i + r_{13}z_i + t_1}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3} \\ Y'_i &= \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_1}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3}. \end{aligned} \quad (2.24)$$

There are six unknowns (three for rotation and three for translation) and therefore with three point correspondences, we have enough (i.e. six) equations. Unfortunately, these are nonlinear transcendental equations, since r_{ij} are related to the three unknowns of the rotation matrix in a transcendental manner. Iterative methods are required to solve these nonlinear algebraic equations. In practice, the data (i.e. p_i

and P'_i) are known only approximately and therefore one may use more than three point correspondences. In such a case, the following nonlinear least squares problem may be solved iteratively:

$$\min_{w.r.t. R, T} \left[\sum_{i=1}^N \left(X'_i - \frac{r_{11}x_i + r_{12}y_i + r_{13}z_i + t_1}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3} \right)^2 + \left(Y'_i - \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_1}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_3} \right)^2 \right] \quad (2.25)$$

subject to R being a rotation matrix. The disadvantages of these approaches is that unless one starts with a good initial guess, the iterative procedure may not converge to the right solution.

If three nonlinear point correspondences are given, then without loss of generality we can assume that three points lie in the plane $z = 0$. Then (2.24) becomes

$$\begin{aligned} X'_i &= \frac{r_{11}x_i + r_{12}y_i + t_1}{r_{31}x_i + r_{32}y_i + t_3} \\ Y'_i &= \frac{r_{21}x_i + r_{22}y_i + t_1}{r_{31}x_i + r_{32}y_i + t_3}. \end{aligned} \quad (2.26)$$

Thus the three point correspondences give six linear and homogeneous equations in nine unknowns $r_{11}, r_{12}, r_{21}, r_{22}, r_{31}, r_{32}, t_1, t_2, t_3$. Assuming $t_3 \neq 0$, we can divide by t_3 , to get six linear equations in eight unknowns

$$\frac{r_{11}}{t_3}, \frac{r_{12}}{t_3}, \frac{r_{21}}{t_3}, \frac{r_{22}}{t_3}, \frac{r_{31}}{t_3}, \frac{r_{32}}{t_3}, \frac{t_1}{t_3}, \frac{t_2}{t_3}$$

Additional constraints on $\{r_{ij}\}$ can be obtained based on the definition that R is a

rotation matrix as

$$\begin{aligned} \left(\frac{r_{11}}{t_3}\right)^2 + \left(\frac{r_{21}}{t_3}\right)^2 + \left(\frac{r_{31}}{t_3}\right)^2 &= \left(\frac{r_{12}}{t_3}\right)^2 + \left(\frac{r_{22}}{t_3}\right)^2 + \left(\frac{r_{32}}{t_3}\right)^2 \\ \left(\frac{r_{11}}{t_3}\right)\left(\frac{r_{12}}{t_3}\right) + \left(\frac{r_{21}}{t_3}\right)\left(\frac{r_{22}}{t_3}\right) + \left(\frac{r_{31}}{t_3}\right)\left(\frac{r_{32}}{t_3}\right) &= 0. \end{aligned} \quad (2.27)$$

Thus we have six linear and two quadratic equations in eight unknowns. According to Bezout's theorem [184], the maximum number of solutions (real or complex) is four, assuming that the number of solutions is finite. Solutions can be obtained by computing the resultant which in this case will be a fourth-degree polynomial in one of the unknowns.

With four coplanar point correspondences, (2.26) yields eight linear homogeneous equations in nine unknowns. If this system of equations is nonsingular, R, T can be obtained uniquely.

If four or five point correspondences are known, then one can either solve a linear least squares problem or use the above method by taking three point correspondences at a time.

If six point correspondences are known, then (2.26) gives twelve linear homogeneous equations in twelve unknowns $\{r_{ij}\}_{i,j=1,\dots,3}$, $\{t_i\}_{i=1,\dots,3}$. R, T can be determined uniquely, if the system is nonsingular.

2.6 Motion from Long Sequences

In the previous section, we have concentrated on motion/structure determination using only two time instants or frames. This is sufficient for some applications (e.g., passive navigation, pose determination, camera calibration), but not for others (e.g., motion prediction). For motion prediction and general understanding, it is necessary to work with longer image sequences. Furthermore, using longer image sequences is potentially a way of combating noise in the data.

2.6.1 Kalman filter

Broida and Chellappa [28] considered the case of a two-dimensional object undergoing one-dimensional motion. They assume that the object structure is known and attempt to recover the motion parameters. The unknown model parameters are represented as a vector:

$$[xc, \dot{xc}, zc, \dot{zc}, p1, p2, w]^t \quad (2.28)$$

where (xc, zc) is the location of the center of mass of the object, (\dot{xc}, \dot{zc}) is the object translational motion, $p1$ and $p2$ are unknown phase angles of moment arms $r1$ and $r2$ that connect the two feature points to the center of mass. Here $r1$ and $r2/r1$ are assumed known. The differential equation describing unforced motion is written in terms of the above vector as:

$$\dot{x}(t) = [\dot{xc}, 0, \dot{zc}, 0, w, w, 0]^t \quad (2.29)$$

with arbitrary initial conditions $xc(t), zc(t), p1(t)$, and $p2(t)$. This system yields the following state equation:

$$x(k+1) = F(k)x(k). \quad (2.30)$$

The measurement model is given by

$$\begin{aligned} X1 &= L[xc + r1\cos(p1)]/[zc + r1\sin(p1)] = h1[x(k)] \\ X2 &= L[xc + r2\cos(p2)]/[zc + r2\sin(p2)] = h2[x(k)] \end{aligned} \quad (2.31)$$

where $X1$ and $X2$ are the images of the two feature points and L is the focal length of the sensor. The vector representation is given by

$$X(k) = [X1(k)X2(k)]^t = h[x(k)] + n(k) \quad (2.32)$$

where $h[x] = [h1(x)h2(x)]$ and $n(k)$ is the term corresponding to zero mean, Gaussian, spatially correlated, and temporally white noise.

The above formulation is then used to design an iterated extended linear Kalman filter that solves for the state variables-in this case the translation and rotation parameters.

Ayache and Faugeras [10] also used an extended Kalman filter to deal with noisy stereo image sequences. An observation x that depends on a parameter a in a non-linear fashion that can be expressed as a relation

$$f(x, a) = 0. \quad (2.33)$$

Assume that the observation x is corrupted with noise which can be modeled as an additive zero mean Gaussian noise:

$$x = x' + \varepsilon \quad (2.34)$$

with $E(\varepsilon) = 0$ and $E(\varepsilon\varepsilon^t) = \Lambda$.

The problem is, given a number of stereo observations x_i and start with an initial estimate \hat{a}_0 of a and its associated covariance matrix $S_0 = E((\hat{a}_0 - a)(\hat{a}_0 - a)^t)$, the Kalman filtering approach can deduce recursively an estimate \hat{a}_n of a and its covariance matrix $S_n = E((\hat{a}_n - a)(\hat{a}_n - a)^t)$ after taking into account n observations. The \hat{a}_n is the parameter vector that minimizes the criterion:

$$(a - \hat{a}_0)^t S_0^{-1} (a - \hat{a}_0) + \sum_{i=1}^n (y_i - M_i a)^t W_i^{-1} (y_i - M_i a) \quad (2.35)$$

where

$$M_i = \frac{\partial f(x_i, a)}{\partial a}$$

$$W_i = \frac{\partial f(x_i, a)}{\partial x} \Lambda \frac{\partial f(x_i, a)^t}{\partial x} \quad (2.36)$$

2.6.2 Other approaches

For a linear problem, theoretically, the result of Kalman filtering is the same as that of a batch method. However, for a nonlinear problem, the result of Kalman filtering is not as good. The key problem with Kalman filtering for the nonlinear problem is

that the system Jacobiaan matrix for each old observation cannot be used for future observations. This is a fundamental structure of sequential processing. To see that clearly, we consider a time-invariant problem $y = f(m) + \delta_y$ where the parameter to be estimated dose not vary with time and the noise is white. In stead of minimizing the desired objective function, $\sum_{i=1}^n [y_i - f_i(m)]^2$, the iterated extended Kalman filter minimizes

$$\sum_{i=0}^n [y_i - J_i(\hat{m}^{(i)})]^2 = \sum_{i=0}^n [y_i - \frac{\partial f_i(\hat{m}^{(i)})}{\partial m} \hat{m}]^2 \quad (2.37)$$

where y_i and $f_i(m)$ are the components of y and $f(m)$, respectively, and $\hat{m}^{(i)}$ is the sequentially estimated parameter vector based on the first i observations. For small i , $\hat{m}^{(i)}$ has a large error since just i observations are available. Therefore, $J_i(m)$ evaluated at $\hat{m}^{(i)}$ gives a system matrix that is evaluated far from the true parameters. This results in inaccurate system equations. Once those inaccurate system equations are generated, they are included in the objective function (2.37), further preventing the estimated parameter m from approaching the correct parameters while new data are obtained.

To overcome the above problems, Cui *et. al.* [45] have proposed a recursive batch approach to deal with long image sequences. Using batching processing, $\hat{m}^{(i)}$ in (2.37) is replaced by \hat{m} , which takes all the observations into account, instead of just the first i observations. Computationally, when all observations are processed in a batch fashion, the modification of parameters is reliable, and the system matrix of every observation is updated at each iteration. In other words, with a sequential algorithm, the contribution or influence of the later observations to the evaluation of

the system matrices for the earlier observations is neglected. In order to achieve good performance without suffering from excessive computational cost, a recursive-batch approach is used. In this approach, the observation sequence is processed in relatively small batches. For each batch of data, estimates are determined in a batch fashion from old estimates and the current batch of data. The approach is recursive because the processing step is repeated for each batch of data and the newly estimated result depends on the previous result.

Tomasi and Kanade [178] have presented a factorization method to recover shape and motion from image sequences. Assume P feature points are tracked over F frames in an image stream. The trajectories of image coordinates are $\{(u_{fp}, v_{fp}) | f = 1, \dots, F, p = 1, \dots, P\}$. Write the horizontal feature coordinates u_{fp} into an $F \times P$ matrix U with one row per frame and one column per feature point. Similarly, an $F \times P$ matrix V is built from the vertical coordinates v_{fp} . The combined matrix of size $2F \times P$

$$W = \begin{bmatrix} U \\ V \end{bmatrix} \quad (2.38)$$

is called the *measurement matrix*. The rows of the matrices U and V are then registered by subtracting from each entry the mean of the entries in the same row:

$$\begin{aligned} \tilde{u}_{fp} &= u_{fp} - a_f \\ \tilde{v}_{fp} &= v_{fp} - b_f \end{aligned} \quad (2.39)$$

where

$$\begin{aligned} a_f &= \frac{1}{P} \sum_{p=1}^P u_{fp} \\ b_f &= \frac{1}{P} \sum_{p=1}^P v_{fp} \end{aligned} \tag{2.40}$$

This produces two new $F \times P$ matrices $\tilde{U} = [u_{fp}]$ and $\tilde{V} = [v_{fp}]$. The matrix

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix} \tag{2.41}$$

is called the *registered measurement matrix*. Without noise, the registered measurement matrix \tilde{W} is at most of rank three. With noise, the best possible shape and rotation estimate is obtained by considering only the three greatest singular values of \tilde{W} , together with the corresponding left and right eigenvectors. The approach is only applicable to orthographic projection.

Chapter 3

Transitory Image Sequences, Asymptotic Properties, and Estimation of Motion and Structure

If a system needs to sense a large 3-D rigid scene which cannot be covered by single view, the system may actively move and scan the scene [5]. For example, to automatically build a 3-D map of a floor in a building, a camera system moves from one room to the next on the floor. To obtain information about all the facets of a 3-D object, a camera system needs to actively circle around the object or let the object rotate. In general, during a dynamic sensing process, any component of the scene is visible only in a subsequence, and thus the resulting image sequence is transitory.

A transitory image sequence is one in which no scene element is visible through the *entire* sequence. When a camera system scans a scene which cannot be covered by a single view, the image sequence is transitory. This chapter deals with some major theoretical and algorithmic issues associated with the task of estimating structure and motion from transitory image sequences.

3.1 Basic Concepts

We consider a rigid scene and a sensing system (we will call it camera system). They undergo a motion relative to each other. No matter which is actually moving, or both are moving, what we need to consider for the kinematics here is just the relative motion between the two.

We first define the system of reference. Because we are considering two entities: the scene and the camera system, it does not help us to place the system of reference on any object other than these two. If the system of reference is placed on the scene, the representation with respect to this system is called world-centered (WC) (also called object-centered). If the system is placed on the camera system, the representation is called camera-centered (CC). Fig. 3.1 shows these two representations. In the WC representation, the camera is moving with respect to a static scene, while in the CC representation, the scene is moving relative to a static camera. To be specific in discussion, we say that the scene is static and camera is moving. Thus, the world-centered reference system is fixed (with the scene) and the camera-centered reference system is moving (with the camera).

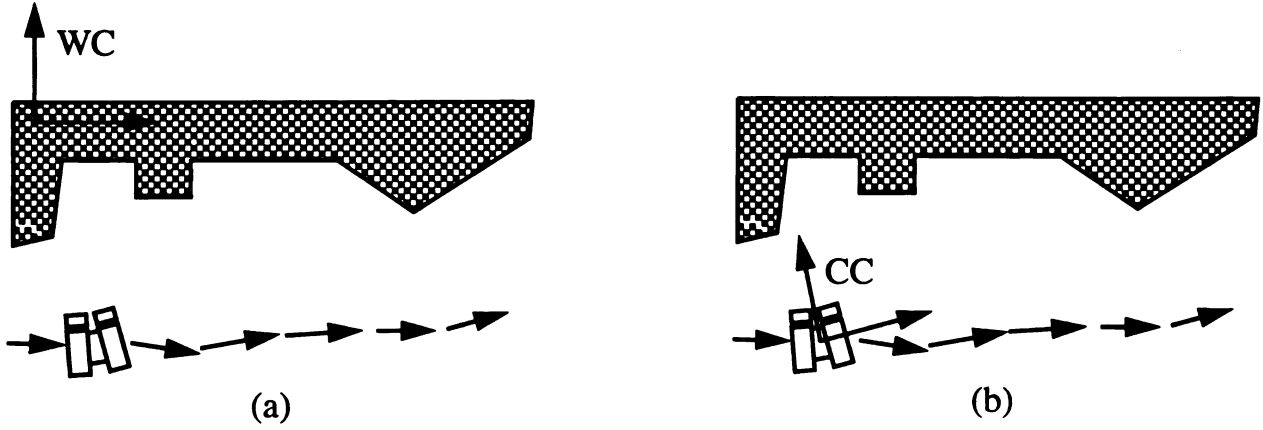


Figure 3.1: Two systems of reference: (a) world-centered and (b) camera-centered.

A view u of a 3-D feature point x is a 2-vector (two dimensional vector) in the monocular case and a 4-vector in the stereo case (left and right views). With random error in the image measurement u , the 3-D position of the point x determined from u becomes a probability distribution whose extent can be characterized by its error covariance matrix Γ_x . The covariance matrix of a 3-D point from a monocular view can be represented by

$$\Gamma_x = H \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix} H^t$$

where, σ_3 is an extremely large number or infinity and the orthogonal matrix H specifies the orientation of the major axes of the covariance. By using a covariance matrix also for monocular view, we can treat monocular and multi-ocular cases in a unified way. Our analysis is applicable to both perspective and orthographic projections. As a notation, we write a small perturbation of a vector v by δ_v , and the error covariance matrix of a vector v by Γ_v .

First, we examine the error from determination of the pose m of a camera system

in a system of reference, where p is a 6-vector. For example,

$$m = (\theta_x, \theta_y, \theta_z, p_x, p_y, p_z) \quad (3.1)$$

where p_x, p_y, p_z specifies the position of the camera projection center and $\theta_x, \theta_y, \theta_z$ specify the orientation of the pose represented by a rotation matrix

$$R(\theta_x, \theta_y, \theta_z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

The pose is estimated from x , a set of 3-D points, represented in that reference system and u , their image observations in the camera. Therefore, the pose is a function of x and u : $m(x, u)$. We can express the error in m in terms of that in x and u :

$$\delta_m = \frac{\partial m}{\partial x} \delta_x + \frac{\partial m}{\partial u} \delta_u \quad (3.3)$$

and for its covariance matrix:

$$\Gamma_m = \frac{\partial m}{\partial x} \Gamma_x \frac{\partial m^t}{\partial x} + \frac{\partial m}{\partial u} \Gamma_u \frac{\partial m^t}{\partial u} \quad (3.4)$$

assuming that the correlation between x and u is negligibly small.

Next, we investigate the error in determining 3-D position of a set of 3-D points y visible by a camera system whose estimated pose is m . These points in y have

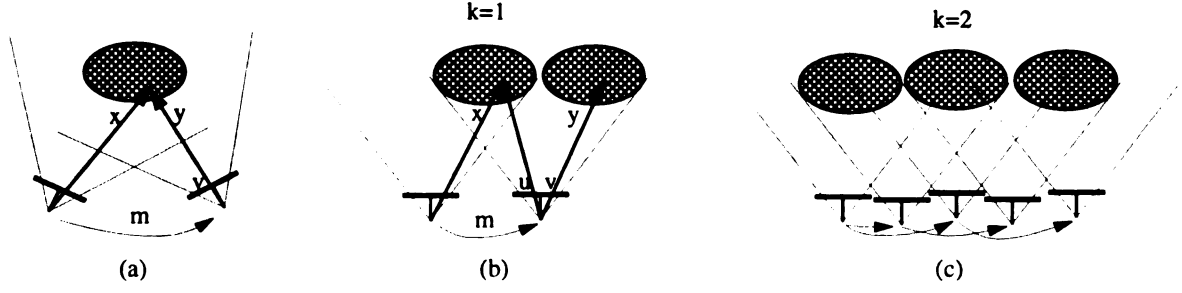


Figure 3.2: Transitory and non-transitory sequences. (a) non-transitory. (b) simple transitory. (c) general transitory.

been viewed as a set of image points v . The estimated 3-D position of points y in the above system of reference is then a function $y(m, v)$. We can express the error in the estimated y by that of m and v as

$$\delta_y = \frac{\partial y}{\partial m} \delta_m + \frac{\partial y}{\partial v} \delta_v \quad (3.5)$$

and for its covariance matrix:

$$\Gamma_y = \frac{\partial y}{\partial m} \Gamma_m \frac{\partial y^t}{\partial m} + \frac{\partial y}{\partial v} \Gamma_v \frac{\partial y^t}{\partial v} \quad (3.6)$$

assuming that the correlation between error in m and v is negligibly small. The above equation indicates that the error covariance of the 3-D points has two components, one is caused by the error in the pose estimate, the other results from error in the feature measurements.

Now, we use the above result to analyze pose determination from x and the use of estimated pose m to determine y . We consider two cases. (i) x and y correspond to the same set of scene points, as shown in Fig: 3.2(a). Thus, u and v are the same

in (3.3) and (3.5), which gives

$$\delta_y = \frac{\partial y}{\partial m} \frac{\partial m}{\partial x} \delta_x + \frac{\partial y}{\partial m} \frac{\partial m}{\partial u} \delta_u + \frac{\partial y}{\partial v} \delta_v = \frac{\partial y}{\partial m} \frac{\partial m}{\partial x} \delta_x + \left(\frac{\partial y}{\partial m} \frac{\partial m}{\partial u} + \frac{\partial y}{\partial v} \right) \delta_v \quad (3.7)$$

which gives

$$\Gamma_y = A\Gamma_x A^t + D\Gamma_v D^t \quad (3.8)$$

where A and D are the appropriate Jacobians.

(ii) x and y correspond to different scene points as shown in Fig: 3.2(b). Substituting Γ_m in (3.4) for that in (3.6), it following that

$$\Gamma_y = A\Gamma_x A^t + B\Gamma_u B^t + C\Gamma_v C^t \quad (3.9)$$

where A , B and C are the appropriate Jacobians. The first term is caused by the error in the 3-D structure x from which the pose is computed. The second term is due to error in u , the observation of x . The third term results from error in the observation of y .

3.2 Asymptotic Error Properties of Integrations

In this section, we derive how the amount of error in the estimate changes with integration of various sequences. We assume that the algorithm obtains a linear minimum variance estimate in the sense of Gauss-Markov [122], which is the minimum variance estimate with Gaussian noise.

In order to investigate the best possible result, the processing method is assumed to be batch unless stated otherwise. This means that all the observed data are available for processing and the estimate is computed with all the data as a single batch. In contrast to batch processing is recursive processing [122] where data items are used one at a time, each giving an updated estimate for the result, and once an data item is used for updating it is discarded. In other words, recursive processing imposes a restriction on the way data are available and thus it might not be able to compute all the estimates that batch processing techniques can compute. Thus, recursive processing may have a worse asymptotic error behavior than the batch processing, unless the model is actually linear [122].

3.2.1 World-centered representation

In the WC representation, every new observation about object structure is transformed into the WC system of reference using the estimated camera pose. Then all the transformed structure observations are fused together according to each's error covariance matrix.

Ideal non-transitory sequence

Consider that a set of feature points y is visible in all the views in the image sequence, as shown in Fig. 3.2(a). Suppose that from $t = 1$ to $t = n$, n observations are made for structure y :

$$y = y_t + \delta_{y_t} \tag{3.10}$$

Without loss of generality, we can assume that the pose m is relative to the pose at $t = 1$. The correlation of error in δ_{y_t} between different t 's is weak because error is random. According to the Gauss-Markov Theorem [122], the linear minimum variance estimator of z in the linear equation $Az = b + \delta$, where the noise term δ has a covariance matrix Γ_δ , is $z = (A^t \Gamma_\delta^{-1} A)^{-1} A^t \Gamma_\delta^{-1} b$ with an error covariance matrix $\Gamma_z = (A^t \Gamma_\delta^{-1} A)^{-1}$. Thus, the minimum variance linear estimate for y in (3.10) is

$$y = \left(\sum_{t=1}^n \Gamma_{y_t}^{-1} \right)^{-1} \left(\sum_{t=1}^n \Gamma_{y_t}^{-1} y_t \right) \quad (3.11)$$

where Γ_{y_t} is given in (3.6). The error covariance matrix of y in (3.10) is given by

$$\Gamma_y = \left(\sum_{t=1}^n \Gamma_{y_t}^{-1} \right)^{-1} \quad (3.12)$$

Theorem 1 *Let A and B be real $n \times n$ positive definite matrices. Then, $A - (A^{-1} + B^{-1})^{-1}$ is positive definite. Particularly, $\text{trace}(A) > \text{trace}((A^{-1} + B^{-1})^{-1})$.*

Proof of Theorem 1.

$$\begin{aligned} A - (A^{-1} + B^{-1})^{-1} &= A - (A^{-1}(B + A)B^{-1})^{-1} \\ &= A - B(B + A)^{-1}A \\ &= (I - B(A + B)^{-1})A \\ &= ((A + B)(A + B)^{-1} - B(A + B)^{-1})A \\ &= A(A + B)^{-1}A \end{aligned}$$

Thus, it is clear that $A(A + B)^{-1}A$ is positive definite because A, B are all positive definite. \square

Using the result of Theorem 1, we know from Equation (3.12) that any observation y_t decreases the expected error in the structure. In order to give a concise and intuitive expression about error covariance matrix, we need to assume some uniformity in the sense that the difference in the error covariance matrix from each view is neglected and each is replaced by the average error covariance matrix. Here, we assume that difference of Γ_{y_t} among different t is neglected. Thus,

$$\Gamma_y = (n\Gamma_{y_t}^{-1})^{-1} = \frac{1}{n}\Gamma_{y_t} = O(1/n) \quad (3.13)$$

Thus, it is clear that the expected error variance in the structure is *inversely proportional* to the number of frames n . We call the factor $1/n$ error rate. The situation discussed here applies to that of “Hotel” sequence in Tomasi and Kanade [178].

A point should be mentioned here. As indicated in (3.7), the first term on the right-hand side is presented in all the observations y_t . This implies that the observations y_t ’s are not exactly uncorrelated. But, if the structure x is re-estimated using all the observations, the correlation between this re-estimated x and y_t is weak and it can be neglected especially when n is large.

Simple transitory sequence

In a simple transitory sequence, each scene point is visible in two consecutive frames.

In this case, the pose m estimated from point set $x_t = x$ and its observation $u_t = u$ is

used to estimate the new structure $x_{t+1} = y$ whose observation is $u_{t+1} = v$, as shown in Fig. 3.2(b). From (3.9), we can estimate the error covariance of the structure x_t :

$$\Gamma_{x_t} = A_t \Gamma_{x_{t-1}} A_t^t + B_t \Gamma_{u_t} B_t^t + C_t \Gamma_{v_t} C_t^t \quad (3.14)$$

Thus, using the above expression recursively, we get

$$\Gamma_{x_n} = \sum_{t=1}^n (B_t \Gamma_{u_t} B_t^t + C_t \Gamma_{v_t} C_t^t)$$

where, B_t and C_t are the products of the appropriate Jacobian and we have neglected error in the re-estimated structure represented in the WC reference system, just as we did in the last subsection. Now, we assume a uniformity in which the difference among the terms under the summation is neglected. Thus,

$$\Gamma_{x_n} = n(B_t \Gamma_{u_t} B_t^t + C_t \Gamma_{v_t} C_t^t) \quad (3.15)$$

In other words, the error covariance in the structure is *proportional* to the number of frames. This implies that error is accumulated with the number of frames.

General transitory sequence

The general situation with a transitory sequence is shown in Fig.3.2(c), where a point can be visible in any number of frames (except the entire sequence). Detailed formulation for this general case is tedious and the resulting complex expression will not give us an insight. Because we are interested in asymptotic error behavior, we

may make some assumption about uniformity. Assume that every feature point is visible in $2k$ frames. Thus, we regard the entire sequence $F = \{f_t \mid t = 0, 1, 2, \dots, n\}$ as k subsequences $F_l = \{f_{pk+l} \mid p \geq 0 \text{ is an integer}\}$, $l = 0, 1, 2, \dots, k-1$, so that in each F_l each point is visible by two frames and each F_l is then a simple transitory sequence. k is called visibility span. The entire sequence consists of k subsequences each is a simple transitory sequence and is of n/k long. According to the result of simple-transitory case with the uniformity assumption, the error covariance matrix of the linear minimum variance estimate based on each F_l is proportional to the length n/k :

$$\Gamma_{x_n} = \frac{n}{k}(B_t \Gamma_{u_t} B_t^t + C_t \Gamma_{v_t} C_t^t) = O(n/k) \quad (3.16)$$

where the factor in the parentheses should be that for a simple transitory subsequence. On the other hand, we have k subsequences, each gives an independent observation of structure x_t . Thus, we can use the result for ideal non-transitory sequence we obtained when we derive (3.13), which says that the error covariance matrix is reduced by a factor of $1/k$:

$$\Gamma_{x_n} = \frac{n}{k^2}(B_t \Gamma_{u_t} B_t^t + C_t \Gamma_{v_t} C_t^t) = O(n/k^2) \quad (3.17)$$

which gives an error rate n/k^2 for error covariance matrix Γ_{x_n} . This is a very interesting rate. If the temporal sampling density is increased by a factor 2 with the same scan trajectory, n and k are both doubled, and the error covariance matrix is reduced by a factor $1/2$! We can see that when $k = 1$, the general rate n/k^2 becomes the rate for the simple transitory case and when $k = n$ it gives that for the non-transitory case.

The error rate n/k^2 in (3.17) implies that the later a part of scene enters the view, the larger the number n , and thus the larger the variance of the error in its position with respect to the WC reference system fixed at the first view.

In the above subsequence decomposition, motion is expressed as interframe motions in each subsequence, which is a motion that cross k frames in the original sequence. Of course, the above subsequence decomposition is not necessarily what is done by an actual estimation algorithm. This decomposition is in fact just a way to derive error rate. By grouping structure observations into the defined subgroups F_l , errors in the estimate from different groups are uncorrelated. Thus, the decomposition is to make derivation of error rate more concise and simpler. It does not affect the asymptotic error rate n/k^2 , because the best estimate is still derived by processing the entire set of structure observations as a single batch.

Global pose error

In a non-transitory case, the error covariance matrix is given in (3.4) which is almost independent of n .

Now, consider the transitory case. According to (3.4), the error variance of camera pose estimate is the sum of two terms, that from Γ_u and that from $\Gamma_{x_{n-l}}$ where $n-l$ is the past time frame that shares sufficient features with the current view at n . Therefore, we have

$$\Gamma_{x_n} = \frac{n-l}{l^2} (B_t \Gamma_{u_t} B^t + C_t \Gamma_{v_t} C^t) + \frac{\partial m}{\partial u} \Gamma_u \frac{\partial m^t}{\partial u} \quad (3.18)$$

Since l is in the same order of k , the asymptotic error rate is n/k^2 . Denote the last term in the above equation by $O(1)$ indicating it is caused by a single view of u vector. Thus, the pose error with a transitory sequence has the same asymptotic error rate as that of the structure estimate:

$$\Gamma_{x_n} = O(n/k^2) + O(1) \quad (3.19)$$

This is an interesting yet expected result. When n/k^2 increases without bound, the error rate of the camera global pose error is n/k^2 , i.e., the first term in (3.19). However, Suppose one increases the temporal density of the sequence which covers the same trajectory, i.e., letting $n = ck$ for some constant c and increase k without bound. Then, the first term in (3.19) becomes $O(ck/k^2) = O(c/k)$ which approaches zero, and thus the second becomes dominant. The expression in (3.17) is a single term of $O(n/k^2)$. This is because the pose of a camera system is determined by a single view. while the structure can be determined by many camera views.

3.2.2 Camera-centered representation

In the CC representation, object structure is represented in the camera reference system. In other words, every previous observation about object structure must be transformed into the camera reference system at the current frame and be fused according to the Gauss-Markov Theorem.

An important difference between the WC and CC representations is the following

- In the WC representation, every part of the scene that has been observed but is

not currently visible does not need to be updated with the current view, because the WC reference system does not change with respect to the scene.

- In the CC representation, every part of the scene that has been observed but is not currently visible must be transformed to the current camera centered system because the CC reference system moves with respect to the scene.

The update for every part of the past structure can be computationally expensive. We will address this point when we describe our cross-frame approach.

With the CC representation, the pose m to be computed is from the past time $t - p$ to the current time t , $p = 1, 2, 3, \dots, t - 1$. After fusing all the past views with the current view at t , the resulting structure is called the CC structure. Theoretically, the structure error should be the same as that with the WC representation if all the past frames are treated in a batch fashion. Thus the behavior of the error covariance matrix for the CC structure is the same as that of the WC structure estimated with the WC representation, except that time t is now reversed: the older the frame, the worse the structure accuracy in the CC representation.

However, the local structure, i.e., that is visible in the current frame, does not have the above transitory problem, simply because it is visible at current time n . Therefore, it can take the advantage of the situation enjoyed by the ideal non-transitory sequence. If the CC structure only take past b frames into account as a batch, and those b frames share a considerable number of features with the current view at n , then, according to the result (3.13) derived with uniform ideal non-transitory sequence, the CC structure

of the currently visible part is of order

$$\Gamma_{x_n} = \frac{1}{b} \Gamma_{y_t} \quad (3.20)$$

where Γ_{y_t} is the error covariance matrix of the past structure transformed to frame n , and b is the batch size. For the above expression to hold true, b should be small enough so that the past b frames share the structure x_n with the view at time n .

Now, we are ready to summarize the asymptotic error rates using Table 3.1. In

Table 3.1: Asymptotic rate for error covariance matrix in integration

Representation	Estimate	Non-transitory	Simple transitory	General transitory
WC	structure	$O(1/n)$	$O(n)$	$O(n/k^2)$
WC	pose	$O(1)$	$O(n)$	$O(n/k^2) + O(1)$
CC	structure	$O(1/n)$	$O(1)$	$O(1/b)$
CC	pose	0	0	0

Table 3.1, n is current time (or frame number), k the visibility span, and b is the batch size $b \leq k$. All the structure error is that for the visible part at the current n -th frame. The camera pose error in CC representation should be zero in all the cases, because it is defined directly in the camera system itself instead of measure. In the table, “0” is used to indicate this fact.

As can be seen from the table, with a general transitory sequence, for global structure representation which is necessary for extended scene reconstruction, one should increase the visibility span k as much as possible. For the camera-centered local structure which is useful in grasping or collision avoidance, one should increase

the batch processing size $b \leq k$ for the best possible accuracy.

3.2.3 The tightness of the error rates

The error rates we obtained in Table 3.1 are achievable rates, using the methods explained in the derivation. However, it is necessary to answer the tightness question. How tight are those rates? In other words, are those rates the best one can possibly achieve? If they are too loose, they are of little value. If they are the best one can possibly achieve theoretically, they give an important insight into the nature of the problem. To answer this important tightness question, we need to look into theoretical bounds with respect to parameter estimation.

In general, the observation model of our problem can be expressed as

$$\hat{u} = u(\alpha) + \delta_u \quad (3.21)$$

where \hat{u} is a vector of image-plane observations, contaminated by noise vector δ_u , and $u(\alpha)$ is the noise-free image plane vector which depends on the parameter vector α . In our problem, u consists of image coordinates of all the features in all the image frames. δ_u is the error vector which takes into account a wide variety of errors, such as errors in spatial digitization, feature detection, stereo matching and temporal matchings, etc. The vector α is the parameter vector one wants to estimate, such as structure of the currently visible scene, camera pose, motion parameters, etc.

Suppose that $\hat{\alpha}$ is an unbiased estimator of α from \hat{u} in (3.21), the noise vector δ_u has a zero mean and covariance matrix Γ_u , and the probability distribution density

of the noise factor is $p(u, \alpha)$. In reality our estimator is not exactly unbiased and the noise mean does not have to be exactly zero. We assume that the absolute bias and the noise mean are negligibly small. The multi-dimensional version of the Cramér-Rao error bound [154, 192] gives

$$E(\hat{\alpha} - \alpha)(\hat{\alpha} - \alpha)^t \geq F^{-1} = \text{CRB}(\alpha) \quad (3.22)$$

where E denotes expectation operator, and F is the Fisher information matrix:

$$F = \left[\frac{\partial \ln p(u, \alpha)}{\partial \alpha} \right]^t \left[\frac{\partial \ln p(u, \alpha)}{\partial \alpha} \right] \quad (3.23)$$

The inequality in (3.22) means that the difference matrix of the two sides is nonnegative definite. In particular, the diagonal elements and the trace of a nonnegative definite matrix are all nonnegative. Therefore Cramér-Rao bound gives a lower error bound for the error covariance of every component of the parameter vector α . As indicated in (3.23), such a bound is evaluated with noise-free observation u and true parameter vector α . It is worth noting that the bound depends on the problem itself and is independent of actual algorithm that is used to estimate α . Thus, the bound is *algorithm independent*. It indicates that no matter what algorithm is used to estimate α , the resulting error covariance matrix of α cannot be lower than that specified by the bound.

Next, we investigate the Cramér-Rao bound of the global pose of the camera system in WC representation. We consider a general transitory sequence of length n ,

$F = \{f_t \mid t = 0, 1, 2, \dots, n-1\}$ with a visibility span k . Since we are investigating asymptotic behavior in which n goes to infinity, without loss of generality, we consider n to be an integral multiple of k , i.e., $n = (j+1)k$, for some positive integer j . $j+1$ is the length of k subsequences $F_l = \{f_{pk+l} \mid p = 0, 1, \dots, j\}$, $l = 0, 1, 2, \dots, k-1$, each of them is a simple transitory sequence.

The simple transitory case

Consider the subsequence F_0 , of length j . As explained in (3.1), the global position of the camera at the i -th frame of F_0 , with respect to its global position at 0-th frame F_0 , can be specified by a column vector

$$m(i) = (p_x(i), p_y(i), p_z(i), \theta_x(i), \theta_y(i), \theta_z(i))^t$$

where $p(i) = (p_x(i), p_y(i), p_z(i))^t$ and $\theta(i) = (\theta_x(i), \theta_y(i), \theta_z(i))^t$ specify the global position and orientation, respectively. Define incremental interframe displacement

$$d(i) = m(i) - m(i-1) \tag{3.24}$$

$i = 1, 2, \dots, j$. From (3.24), we have the relation $m(i) = \sum_{t=1}^i d(t) + m(0)$, or

$$\begin{bmatrix} m(0) \\ m(1) \\ \vdots \\ m(j) \end{bmatrix} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \begin{bmatrix} m(0) \\ d(1) \\ \vdots \\ d(j) \end{bmatrix} \tag{3.25}$$

with I denoting the identity matrix. Alternatively, we write

$$m_{j,0} = M_j d_{j,0}$$

where we denote the left side of (3.25) by $m_{j,0}$ and the right side by the product of the matrix M_j and vector $d_{j,0}$. Geometrically, $m_{j,0}$ is the global attitude trajectory of the camera system while $d_{j,0}$ is the interframe displacement vector, plus the initial attitude $m(0)$. According to the definition of the Cramér-Rao bound, we have

$$\text{CRB}(d_{j,0}) = \left\{ \left[\frac{\partial \ln p(u, d_{j,0})}{\partial d_{j,0}} \right]^t \left[\frac{\partial \ln p(u, d_{j,0})}{\partial d_{j,0}} \right] \right\}^{-1}$$

and

$$\text{CRB}(m_{j,0}) = \left\{ \left[\frac{\partial \ln p(u, m_{j,0})}{\partial m_{j,0}} \right]^t \left[\frac{\partial \ln p(u, m_{j,0})}{\partial m_{j,0}} \right] \right\}^{-1}$$

Since

$$\frac{\partial \ln p(u, d_{j,0})}{\partial d_{j,0}} = \frac{\partial \ln p(u, m_{j,0})}{\partial m_{j,0}} \frac{\partial m_{j,0}}{\partial d_{j,0}} = \frac{\partial \ln p(u, m_{j,0})}{\partial m_{j,0}} M_j$$

it follows that

$$\text{CRB}(m_{j,0}) = M_j \text{CRB}(d_{j,0}) M_j^t \tag{3.26}$$

For our purpose of investigating the asymptotic behavior of the Cramér-Rao bound, we need the uniformity condition of the motion sequence as we did earlier, since the behavior of an otherwise arbitrarily changing motion trajectory can depend more on a particular local motion instead of the temporal trend of the error behavior. Now, we assume a uniformity with which the differences among the interframe motions

$d(i), i = 1, 2, \dots, j$ are neglected. In other words, the Cramér-Rao bound of interframe motions $\text{CRB}(d_{j,0})$, which is a symmetric matrix, is now a band matrix:

$$\text{CRB}(d_{j,0}) = \begin{bmatrix} C_0 & C_1 & C_2 & & 0 \\ C_1 & C_0 & C_1 & \ddots & \\ C_2 & C_1 & C_0 & \ddots & C_2 \\ & \ddots & \ddots & \ddots & C_1 \\ 0 & & C_2 & C_1 & C_0 \end{bmatrix} \quad (3.27)$$

In other words, denoting $\text{CRB}(d_{j,0}) = [C_{pq}]$, then $C_{pq} = C_{qp} = 0$ whenever $|p - q| \geq h$, for some constant h . The unfirmity condition requires that the error bounds for estimating interframe motions d_i and d_j , respectively, are not correlated when the interframe motions are farther than h frames apart. This is a reasonable condition because although interframe motion depends mostly on the two image frames that defined the interframe motion. Although the information about the scene structure may contribute to the estimation of interframe motion to some degree, two far apart interframe motions do not share any common scene element when h is large enough in a general transitory sequence. With a simple transitory sequence, two interframe motions do not share any common scene element as soon as $|p - q| \geq 2$.

Without loss of generality, we can consider $h = 2$ for a simple transitory sequence.

Thus, (3.26) and (3.27), give

$$\text{CRB}(m_{j,0}) = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \begin{bmatrix} C_0 & C_1 & & 0 \\ C_1 & C_0 & \ddots & \\ & \ddots & \ddots & C_1 \\ 0 & & C_1 & C_0 \end{bmatrix} \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix}^t \quad (3.28)$$

The last element in the bottom row of $\text{CRB}(m_{j,0})$ is the Cramér-Rao bound (CRB) for the global pose of the camera at frame j of F_0 , which gives

$$\text{CRB}(m_{j,0}) = \begin{bmatrix} I & I & \cdots & I \end{bmatrix} \begin{bmatrix} C_0 & C_1 & & 0 \\ C_1 & C_0 & \ddots & \\ & \ddots & \ddots & C_1 \\ 0 & & C_1 & C_0 \end{bmatrix} \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} = (j+1)C_0 + 2jC_1 = O(j) \quad (3.29)$$

In other words, we have proved

Theorem 2 *Under the uniformity condition, the Cramér-Rao bound (CRB) of the global pose error at frame j in a simple transitory sequence is of the order $O(j)$.*

It is worth noting that we have not imposed any particular distribution type on noise in the observation other than the uniformity condition, the above conclusion holds for any noise distribution provided the uniformity condition is satisfied.

In (3.27), we regarded $m(0)$ the same as $d(1), d(2), \dots, d(j)$ in the vector $d_{j,0}$ just for notation convenience. It will not affect the order of the Cramér-Rao bound we derived in (3.29) since error in $m(0)$ affects just a constant in $\text{CRB}(m_{j,0})$.

The Cramér-Rao bound of the global position of the structure can be investigated in a similar manner. Each scene element visible by the last frame in F_0 is a function of global pose coupled with the structure estimate from the global pose, which results in an additive constant error covariance in addition to that of the global pose. Therefore, the Cramér-Rao bound of the global position of the structure has the same order in error rate as the global pose.

The general transitory case

First, we extend the above result for F_0 to the other subsequences F_l . We extend our notation from $m_{j,0}$ and $d_{j,0}$ to $m_{j,l}^{(l)}$ and $d_{j,l}^{(l)}$, respectively, to denote the corresponding trajectories of F_l , starting from frame f_l to frame f_{j_k+l} , $l = 0, 1, \dots, k-1$. Given F , the above discuss still holds for F_l , except that the meaning of C_i in (3.27) is the Cramér-Rao bound of the corresponding component based on the entire F instead of just F_0 . Therefore, the Cramér-Rao bound of the error rate of the global pose $m_{j,l}^{(l)}$ is still of order $O(j) = O(n/k)$:

$$\text{CRB}(m_{j,l}^{(l)}) = O(n/k) \quad (3.30)$$

Consider each scene element x_n that is visible from f_{j_p+l} , the last frame of F_l , $l = 0, 1, \dots, k-1$. Since Cramér-Rao bound is a lower bound and Table 3.1 means that $\text{CRB}(x_n) \leq O(n/k^2)$. To establish $\text{CRB}(x_n) = O(n/k^2)$, all we need to prove is $\text{CRB}(x_n) \geq O(n/k^2)$. To do the latter, we can neglect some errors. For $l = 1, 2, \dots, k-1$, we neglect the interframe pose error between frame f_0 and f_l , and the

error in the process of constructing x_n from frame f_{jp+l} . Thus, x_n is determined by the camera global position at f_{jp+l} by a function g :

$$x_n = g(m_{j,0}^{(0)}, m_{j,1}^{(1)}, \dots, m_{j,k-1}^{(k-1)}) = g(m_j)$$

where we define

$$m_j = (m_{j,0}^{(0)}, m_{j,1}^{(1)}, \dots, m_{j,k-1}^{(k-1)})^t$$

Since all subsequences F_l are independent with each other, the Cramér-Rao bound of m_j is a block diagonal matrix:

$$\text{CRB}(m_j) = \text{diag}\{\text{CRB}(m_{j,0}^{(0)}), \text{CRB}(m_{j,1}^{(1)}), \dots, \text{CRB}(m_{j,k-1}^{(k-1)})\} \quad (3.31)$$

Using the variable change technique as we used before, we have

$$\begin{aligned} \text{CRB}(x_n) &= \left\{ \left[\frac{\partial \ln p(u, x_n, m_j)}{\partial x_n} \right]^t \left[\frac{\partial \ln p(u, x_n, m_j)}{\partial x_n} \right] \right\}^{-1} \\ &\geq \left\{ \left[\frac{\partial m_j}{\partial x_n} \right]^t \left[\frac{\partial \ln p(u, x_n, m_j)}{\partial m_j} \right]^t \left[\frac{\partial \ln p(u, x_n, m_j)}{\partial m_j} \right] \left[\frac{\partial m_j}{\partial x_n} \right] \right\}^{-1} \\ &= \left\{ \left[\frac{\partial m_j}{\partial x_n} \right]^t \text{CRB}(m_j)^{-1} \left[\frac{\partial m_j}{\partial x_n} \right] \right\}^{-1} \end{aligned} \quad (3.32)$$

Since $\text{CRB}(m_j)$ is block diagonal, then so is its inverse, the above inequality gives

$$\text{CRB}(x_n) \geq \left\{ \sum_{l=0}^{k-1} \left[\frac{\partial m_{j,l}^{(l)}}{\partial x_n} \right]^t \text{CRB}(m_{j,l}^{(l)})^{-1} \left[\frac{\partial m_{j,l}^{(l)}}{\partial x_n} \right] \right\}^{-1}$$

Under the uniformity condition, $\frac{\partial m_{j,l}^{(l)}}{x_n}$ and $\text{CRB}(m_{j,l}^{(l)})$ are treated as constant with respect to l . Thus

$$\begin{aligned} \text{CRB}(x_n) &\geq \left\{ k \left[\frac{\partial m_{j,0}^{(0)}}{x_n} \right]^t \text{CRB}(m_{j,0}^{(0)})^{-1} \left[\frac{\partial m_{j,0}^{(0)}}{x_n} \right] \right\}^{-1} \\ &= \left[\left[\frac{\partial m_{j,0}^{(0)}}{x_n} \right]^t \right]^{-1} \frac{1}{k} \text{CRB}(m_{j,0}^{(0)}) \left[\frac{\partial m_{j,0}^{(0)}}{x_n} \right]^{-1} = O(n/k^2) \quad (3.33) \end{aligned}$$

the last equation used the result in (3.30). Therefore, we have $\text{CRB}(x_n) = O(n/k^2)$.

The Cramér-Rao bound for global pose can be directly derived from that of the global position of the structure. The derivation for the order of Cramér-Rao bound in nontransitory case is simple and is omitted.

In summary, we have established the following result:

Theorem 3 *The asymptotic error rates in Table 3.1 are not only reachable but also the theoretical lowestest possible specified by the Cramér-Rao lower bound. This is true for any distribution as long as the uniformity condition is satisfied.*

These error rates are determined by the nature of the transitory sequence. Although we have used the uniformity condition so that the rate can be expressed simply, the uniformity condition can be applied to *ensemble average* in terms of random process. Passing without a rigorous proof, the rates stated in Theorems 2 and 3 are probably true for general random motion sequences in the sense that they are average rates as long as the uniformity is true on average.

3.3 Methods and Algorithms

The above analysis motivated our method of keeping two representations, WC for global measurements and CC for local measurements. To be specific, we assume a stereo camera system. The method can be directly extended to monocular case without any major modification.

We first consider estimation with a nonlinear observation function f . Suppose that an observation vector y is related to a parameter vector m by a nonlinear equation

$$y = f(m) + \delta_y$$

where δ_y is a pairwise uncorrelated random noise vector with zero mean, and covariance matrix $\Gamma_y = E\delta_y\delta_y^t$. The maximum likelihood estimate with Gaussian noise δ_y or minimum variance linear estimate with a general noise distribution calls for minimizing

$$(y - f(m))^t \Gamma_y^{-1} (y - f(m)) \quad (3.34)$$

with respect to m . In other words, the optimal parameter vector m is the one that minimizes the matrix-weighted discrepancy between the computed observation $f(m)$ and the actual observation y . At the solution that minimizes (3.34), the estimated \hat{m} has a covariance matrix

$$\Gamma_{\hat{m}} = E(\hat{m} - m)(\hat{m} - m)^t \simeq \left\{ \frac{\partial f(\hat{m})^t}{\partial m} \Gamma_y^{-1} \frac{\partial f(\hat{m})}{\partial m} \right\}^{-1} \quad (3.35)$$

One of the advantage of this minimum variance criterion is that we do not need to know the exact noise distribution.

3.3.1 Cross-frame approach with CC representation

Let X_p denote the 3-D positional vector of a point represented in the CC system at frame p . Point X_q represented in the CC system at frame q is moved to X_p in the CC system at time p :

$$X_p = R_{p,q}X_q + T_{p,q}.$$

where $R_{p,q}$ and $T_{p,q}$ are a rotation matrix and a translation vector, respectively. Let $m_{p,q}$ which is a function of $R_{p,q}$ and $T_{p,q}$, denote the relative pose from q to p .

All the structure observed in the past needs to be transformed to the CC system at frame p and properly fused. There are two basic approaches in the fusion of the past structure.

1. Recursive method: frame by frame. The fused structure at previous frame is transformed to the current frame p and fused with the new observation at p according to the estimated interframe motion $m_{p,p-1}$.
2. Batch method: cross-frame. For each $q \in \{p-1, p-2, \dots, p-b+1\}$, estimate the cross-frame motion $m_{p,q}$ and transform X_q to frame p and fuse with the new observation at p .

The first method involves two frames at a time, $p-1$ and p . As we discussed before, the transformed structure has an error covariance matrix $\Gamma_m + \Gamma_{p-1}$ where Γ_m is for

error in interframe motion and Γ_{p-1} for that in the fused structure up to frame $p - 1$. We can see that the error covariance in the transformed structure is increased due to error in the interframe motion. Thus, according to the similar derivation that leads to (3.12), the fused structure has an error covariance matrix of

$$((\Gamma_m + \Gamma_{x_{p-1}})^{-1} + \Gamma_p^{-1})^{-1}$$

where Γ_p is the error covariance matrix of single observation at p . We can see from the above expression that the error variance will not approach zero with the number of frames integrated, because even if $\Gamma_{x_{p-1}} = 0$, we still have a considerably large $(\Gamma_m^{-1} + \Gamma_p^{-1})^{-1}$. The reason is that the error in interframe motion deteriorates the previous structure estimate. A structure estimate at frame $p - l$ will undergo l such deteriorations under the frame-by-frame recursive method and thus, when $l > 1$, the old structure estimate is hardly useful in the fusion with that in view p .

Under the second cross-frame method, each previous structure estimate at $p - l$ is directly transformed to p under one transformation. Thus the transformed structure deteriorates by the motion error only once. The error covariance matrix of each transformed structure is a sum of two terms, one from single cross-frame motion and the other from the observation error at the frame $p - l$, as long as frame $p - l$ is in the same batch as frame p . Thus, following a derivation similar to that in Section 3.2.1, we know that the second method has an asymptotic error rate of $1/b$ as listed in Table 3.1. Fig. 3.3 graphically explains the advantage of cross-frame motions.

In practice, we define a number K , called extra batch size, to be the number

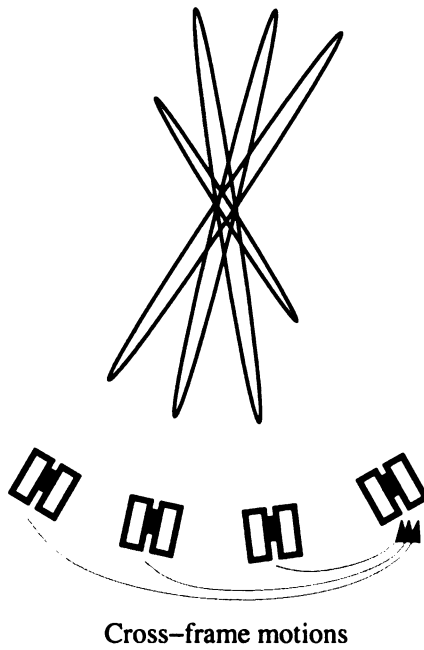


Figure 3.3: Using cross-frame motions to integrate many views. Each elongated ellipse indicates the uncertainty in 3-D point position transformed from a single previous stereo view to the current view. The integrated uncertainty is greatly reduced using the multiple cross-frame motions instead of interframe motions.

of extra image (stereo) frames that are processed as a batch in addition to the last two. Thus, at current frame number p , all the image frame batch consists of frames from $p - K - 1$ to p . Due to the transitory nature of the image sequence, any frame q , with $q < p - K - 1$, with not share any portion of the scene with frame p if K is sufficiently large. According to our discussion about non-transitory and transitory image sequences, it is useful for K to span a subsequence that is nearly nontransitory. Otherwise, the estimate is sequential in nature, as expressed in (3.14), beyond a certain extent.

With a batch at frame p , the current active cross-frame motion set is denoted by

$$W(p) = \bigcup_{i=p-K-1}^{p-1} \{m_{p,i}(R_{p,i}, T_{p,i})\}.$$

The cross-frame motion set completely defines the motion between any two frames within the batch. When $K = 0$, we have just an interframe motion in $W(p)$.

Let N be the total number of feature points being considered; $x_{i,s}$ denote the three dimensional local structure of i -th point in s -th camera-centered system; $u_{i,j,s}$ be the 2-D image coordinate vector of i -th point on the j -th side (left, right) at the s -th frame. Assuming that the noise in the observations ($u_{i,j,s}$) is uncorrelated and has the same variance (σ_u^2, σ_v^2) in the two image coordinates, expression (3.34) that is to be minimized can be written the following form

$$\min_{\forall x_{i,p}, \forall m \in W(p)} f(m, x_{i,p}) = A + B \quad (3.36)$$

where

$$A = \sum_{i=1}^N \{S_i^t (R_{p,p-K-1} \Gamma_{x_{i,p-K-1}^*} R_{p,p-K-1}^t)^{-1} S_i\}$$

with

$$S_i = (x_{i,p} - X(m_{p,p-K-1}, x_{i,p-K-1}^*))$$

and

$$B = \sum_{s=p-K}^p \sum_{j=L}^R \sum_{i=1}^N (\hat{u}_{i,j,s} - u(m_{s,p}, x_{i,p}))^t \begin{bmatrix} \sigma_u^{-2} & 0 \\ 0 & \sigma_v^{-2} \end{bmatrix} (\hat{u}_{i,j,s} - u(m_{s,p}, x_{i,p}))$$

In the above expression, $X(m_{s,p}, x_{i,p})$ is the transformation function to transform the point $x_{i,p}$ from camera coordinate system at frame p to frame s based on the motion parameters $m_{s,p}$. Function $u(m_{s,p}, x_{i,p})$ is the noise-free projection computed from $m_{s,p}$ and $x_{i,p}$, which includes transformation and projection. This objective function has two terms. The first term, A , reflects the integrated 3-D structure in the past up to time $p - K - 1$. The second term, B , is used to minimize the image plane error of the frames within the batch from $p - K$ up to p . The summation bound for i can be modified to include only those points that are visible in each frame so that a point does not have to be visible through the entire batch.

Minimization of the objective function

The objective function in (3.36) is neither linear nor quadratic in terms of cross-frame motion parameters, m , and 3-D feature points, x . An iterative algorithm is required to search for the solution of m and x . The dimension of the unknown parameters is

intractably huge due to a typically large N . Thus, a direct optimization is impractical.

Our two procedures play a central role in resolving this problem:

First, a (suboptimal) closed-form solution for interframe motion from $p - 1$ to p is first computed. This interframe motion is used together with previous pose estimate to compute a preliminary estimate for all the cross-frame motions needed.

The second is to eliminate iteration on the structure. The gradient-based search is only applied to cross-frame motions, because given each candidate set of cross-frame motions the best structure for (3.36) can be directly computed in a closed-form. To show how, let us examine the objective function (3.36). The second term of the objective function corresponds to minimizing the image vector error within the batch. An alternative way to approximate this is to use the matrix-weighted discrepancy of $x_{i,p} - X(m_{p,j}, x_{i,j})$, the 3-D position difference, to give the total discrepancy

$$\min_{x_{i,p}} \sum_{s=p-K}^p \sum_{i=1}^N (x_{i,p} - X(m_{p,s}, x_{i,s}^*))^t (R_{p,s} \Gamma_{x_{i,s}^*} R_{p,s}^t)^{-1} (x_{i,p} - X(m_{p,s}, x_{i,s}^*)) \quad (3.37)$$

where $x_{i,s}^*$ is computed from the triangulation at frame s , $\Gamma_{x_{i,s}^*}$ is the estimated covariance matrix of $x_{i,s}^*$ for triangulation. Substituting the second term of objective function (3.36) with (3.37), we minimize

$$\min_{x_{i,p}} f(x_{i,p}) = \sum_{s=p-K-1}^p \sum_{i=1}^N (x_{i,p} - X(m_{p,s}, x_{i,s}^*))^t (R_{p,s} \Gamma_{x_{i,s}^*} R_{p,s}^t)^{-1} (x_{i,p} - X(m_{p,s}, x_{i,s}^*)) \quad (3.38)$$

given any $W(p)$. The above is a linear minimization problem, for which we just need

to solve the following linear equation [59],

$$\left\{ \sum_{s=p-K-1}^p [R_{p,s} \Gamma_{x_{i,s}}^{-1} R_{p,s}^t] \right\} x_{i,p} = \sum_{s=p-K-1}^p \{ (R_{p,s} \Gamma_{x_{i,s}}^{-1} R_{p,s}^t) X(m_{p,s}, x_{i,s}) \} \quad (3.39)$$

which gives

$$x_{i,p} = \left\{ \sum_{s=p-K-1}^p [R_{p,s} \Gamma_{x_{i,s}}^{-1} R_{p,s}^t] \right\}^{-1} \left\{ \sum_{s=p-K-1}^p \{ (R_{p,s} \Gamma_{x_{i,s}}^{-1} R_{p,s}^t) X(m_{p,s}, x_{i,s}) \} \right\}$$

Its error covariance matrix is estimated by [59]

$$\Gamma_{x_{i,p}} = \left[\sum_{s=p-K-1}^p \Gamma_{i,s}^{-1} \right]^{-1}$$

where

$$\Gamma_{i,s} = \left(\frac{\partial X(m_{p,s}, x_{i,s}^*)}{\partial m} \right) \Gamma_{m_{p,s}} \left(\frac{\partial X(m_{p,s}, x_{i,s}^*)}{\partial m} \right)^t + \left(\frac{\partial X(m_{p,s}, x_{i,s}^*)}{\partial x_{i,s}^*} \right) \Gamma_{x_{i,s}^*} \left(\frac{\partial X(m_{p,s}, x_{i,s}^*)}{\partial x_{i,s}^*} \right)^t$$

3.3.2 World-centered representation

The WC representation follows a similar derivation. The difference is that the structure does not move in WC system. Thus, the structure integrated in the WC system up to any time can be used directly for later WC integration.

Objective function

Without loss of generality, let the world coordinate system coincide with the camera-centered coordinate system of the first frame. $M(m, n) = \cup_{i=m}^n \{m_{i,1}\} =$

$\cup_{i=m}^n \{R_{i,1}, T_{i,1}\}$, is the collection of all the global motions, where $(R_{i,1}, T_{i,1})$ is the rotation matrix and translation vector from frame 1 to i . For each feature points i , we have structure G_i corresponding to the world coordinate system. Now slightly modifying the equation (3.36), we get the appropriate objective function for the WC representation:

$$\min_{\forall G_i, \forall m \in M(p-K-1, p)} f(G_i, m) = A + B \quad (3.40)$$

where

$$A = \sum_{i=1}^N (G_i - G_i^*)^t \Gamma_{G_i^*}^{-1} (G_i - G_i^*) \quad (3.41)$$

and

$$B = \sum_{s=p-K}^p \sum_{j=L}^R \sum_{i=1}^N (\hat{u}_{i,j,s} - u(m_{s,1}, G_i))^t \begin{bmatrix} \sigma_u^{-2} & 0 \\ 0 & \sigma_v^{-2} \end{bmatrix} (\hat{u}_{i,j,s} - u(m_{s,1}, G_i)) \quad (3.42)$$

In the objective function, $u(m_{s,1}, G_i)$ is the noise-free projection computed from $m_{s,1}$ and G_i . The essence of the above objective function is that newly updated global structure G_i takes into account the old observation G_i^* integrated up to frame $p - K - 1$, but it considers all the observations in the batch as image vectors, all properly weighted in the sense of Gauss-Markov.

Similar to computation for the CC representation, no iteration is needed for structure part, and a suboptimal closed-form solution is computed first for motion and structure which is used as the initial guess for minimization. The following equation gives the closed form solution for structure parameters G_i when the motion parame-

ters $M(m, n)$ are given:

$$G_i = \left(\sum_{s=p-K-1}^p \Gamma_{G_i,s}^{-1} \right)^{-1} \left(\sum_{s=p-K-1}^p \Gamma_{G_i,s}^{-1} G_{i,s} \right) \quad (3.43)$$

where $G_{i,s}$ is the estimation based on the single frame s . The estimated error covariance matrix of the newly updated the structure is

$$\Gamma_{G_i} = \left(\sum_{s=p-K-1}^p \Gamma_{G_i,s}^{-1} \right)^{-1} \quad (3.44)$$

This WC based objective function is in essence similar to those of [10] and [131]. The differences are (a) a batch parameter K is used to better deal with the transitory sequence; (b) image-plane discrepancy is minimized to automatically take into account non-symmetrical nature of error distribution in 3-D point positions; (c) the algorithm can automatically handle leaving points and coming points which is required with transitory sequences.

3.4 Experiments

We conducted experiments with synthetic and real world images in order to experimentally exam the error rates listed in Table 3.1 and compare the WC and CC representations.

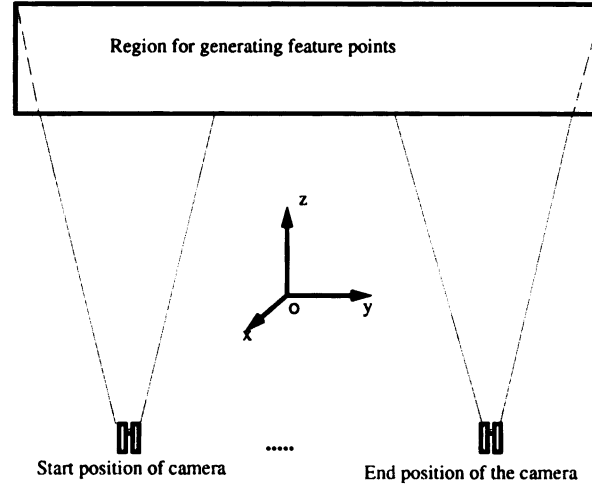


Figure 3.4: Simulation environment, where 7000mm distance is covered by the 31 frames.

3.4.1 Simulation Data

3-D feature points were generated randomly for each trial, between depth 2000mm and depth 3000mm, with a uniform distribution. The entire scene is covered by 31 frames and the distance between consecutive frames is roughly 200mm. A small rotation is added between each pair of two consecutive frames. Fig. 3.4 illustrates the simulation environment. This environment is similar to the real setup to show later. The average errors we will show were obtained through 100 random trials each with a different set of 3-D points.

Measurement

The simulated camera system has a resolution of 512×480 pixels, just like the real cameras we used. Measurement error was simulated by pixel round-off error. This level of measurement accuracy is generally higher than but close to the accuracy of our feature detector, matcher and tracker according to our visual inspection of

the algorithm. The camera's global orientation is determined by a rotation matrix ($R_{i,1}$) and the position by translation vector ($T_{i,1}$). The error of a matrix or vector is measured as the Euclidean norm of the difference between the estimated and true one. The world system is placed at the scene at the first frame, based on which the global structure of the feature points is defined. If a feature point is visible at certain frame, it also has a local structure (i.e., with respect to the CC reference at that frame). In the WC representation, the global structure is directly estimated but its local structure needs to be computed via the estimated global pose of the camera. The situation is just the opposite in the CC representation, where the local structure is directly estimated while the global structure must be computed via camera's global pose.

Batch Size

Visibility span determines the number of frames which share a common view. It can also be used as a criterion to select the maximum batch size. Obviously, a batch size that is beyond the visibility span cannot help much. With our setup, in order to let the first and the last frame in the batch share at least 30% of the scene, the batch size should not be larger than 3.

Results

Fig. 3.5 shows the current camera position error ($R_{i,1}$, $T_{i,1}$) for different frames, where i is the index of the time as shown in the figure. The first column is for the CC representation and second is for the WC representation. The results indicate that the

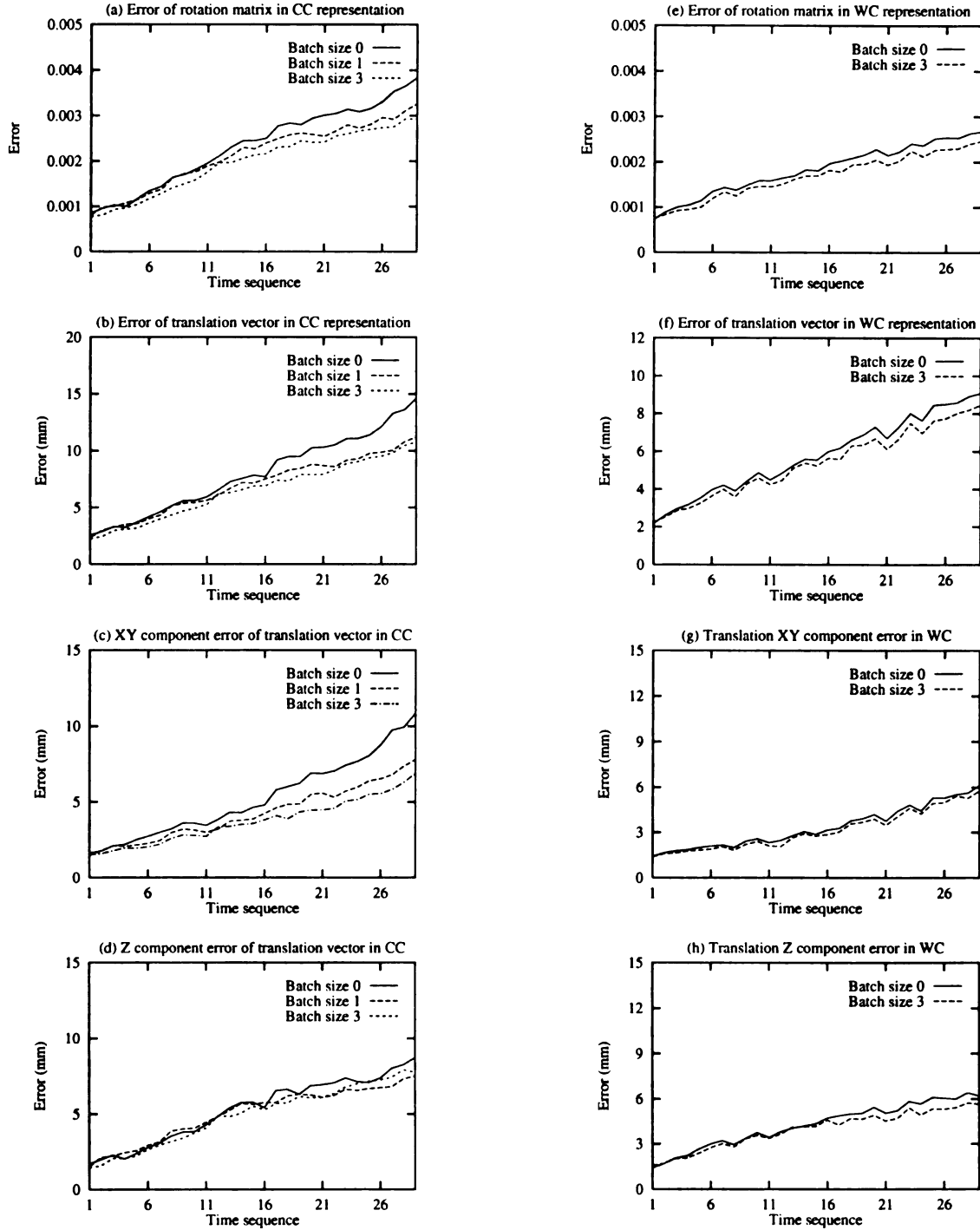


Figure 3.5: Camera global pose error versus time. The CC representation on the left column and WC on the right. (a) and (e): Error in rotation matrix. (b) and (f): Error in translation vector. (c) and (g): Error in xy -component of the translation vector. (d) and (h): Error in z -component of the translation vector.

error increases with the time, an inherent property with transitory image sequences as Table 3.1 shows. It can be seen from the figure that the batch size has more impact in the CC representation than WC. This is because in the CC representation, the reference frame moves, which introduces more nonlinearity than the WC case when the old observation is transformed into the current CC reference system. A larger batch size is more appropriate for such a nonlinear transform, because covariance for error modeling is based on a linear approximation for nonlinear systems. Fig. 3.5 clearly shows that for camera pose estimates the WC representation is a little better, which is consistent with Table 3.1.

Fig. 3.6 shows the local and global structure error. The error is shown as the average error of all the visible feature points at the current frame. The result indicates that a larger batch size is very effective to reduce both the local and global structure errors, for CC representation, as we predicted in Section 3.3.1. Because of the structure is directly estimated in the WC representation and thus the “measurement equation” is linear. Thus, a larger batch size does not improve much for WC representation due to dominantly linear nature of the WC structure fusion. The figure also shows that the WC representation performs better for estimating the global structure while the CC representation does better for local structure, as predicted by Table 3.1. A point worth noting here is the fact the local structure error with the CC representation is constant, while that with the WC representation grows with time, also a property predicted by Table 3.1.

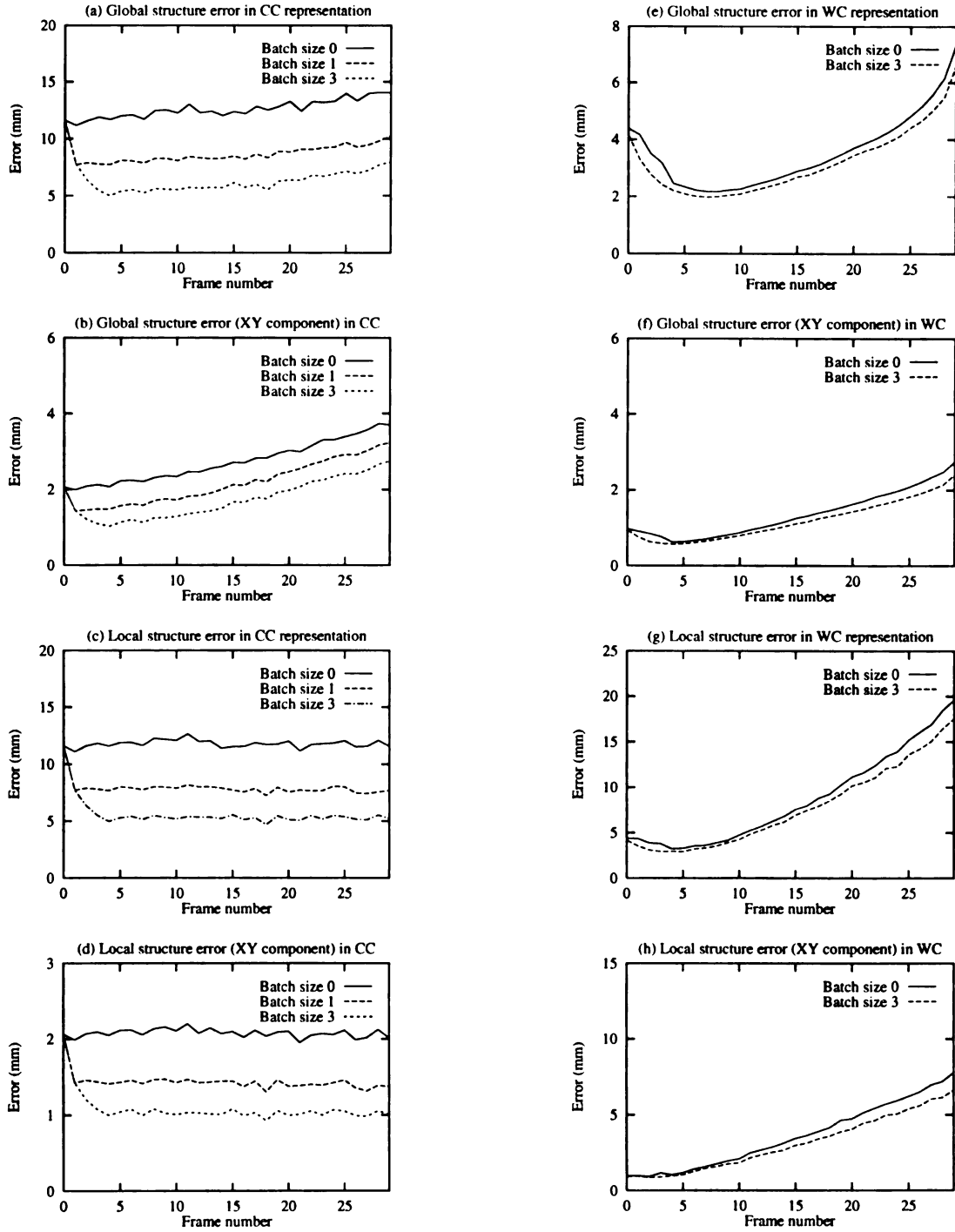


Figure 3.6: Structure error versus time. The CC representation on the left column and WC on the right. (a) and (e): Global structure error. (b) and (f): xy -component of the global structure error. (c) and (g): Local structure error. (d) and (h): xy -component of the local structure error.

3.4.2 Experiments with a real setup

A challenging task facing the area of motion and structure analysis is to provide data from rigorous experiments that verified the actual accuracy of the results, so that we can evaluate whether passive structure sensing is possible and reliable in real world. The result reported here is an effort toward this goal. The data are processed in an off-line fashion.

The setup used for our image acquisition is a Denning MRV-3 mobile robot and a pair of stereo cameras, 265mm apart, mounted on a custom-designed stereo positional setup that allows step-motor controlled pan and tilt for each camera from a computer, as shown in Fig. 3.7. The stereo camera system was calibrated with distortion compensation using an algorithm from Weng *et al* [190]. The field of view of each camera is about 36 degrees diagonally. and each digitized image has 512×480 pixels. An image sequence of 151 frames was acquired from the moving mobile robot. It contains a left-view sequence and a right-view sequence. The entire stereo sequence was used for automatic feature extraction, matching and tracking. A temporally subsampled (one sample every 5 frames) subsequence of 31 frames was used for motion and structure estimation with a consideration that this subsequence is dense enough for estimation and yet enables cross-frame motions to cover more original frames with a relatively small batch size. Fig. 3.7 shows a few images in the 151-frame sequence.

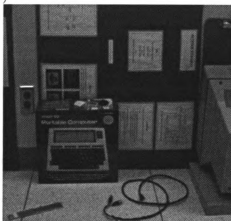
A feature point detector has been developed for this project to automatically detect feature points from images. The feature detector first computes the cornerness measure (the degree a point looks like at a corner) at every pixel. Then the local peaks



(a)



(b)



(c)



(d)



(e)

Figure 3.7: The robot and a few stereo frames in the 151-frame sequence. (a) Robot. (b) Left image of frame 0. (c) Left image of frame 50. (d) Left image of frame 100. (e) Left image of frame 150.

of this cornerness measure are detected to form a peak histogram, ranked with the cornerness measure. The program automatically determines the threshold so that the required number of features are given from top rankings. An area directed analysis is incorporated into the scheme so that the detected feature points evenly spread across the entire image.

Stereo matching was done using an image matching algorithm from Weng *et al* [188], which provides a dense displacement field with a disparity vector for every pixel. The disparity vector at every feature point is extracted from this field.

For efficiency, the algorithm uses tracking mechanism as much as possible. Only when the tracking is not successful based on the closeness measure used by tracking, the matcher is called. For each addition of new points, the stereo and temporal matchings are performed using the same algorithm from Weng *et al* [188]. Once a new feature point is added from a left image, a square template (7×7) centered at this point is recorded as the left template for this point. The stereo matching algorithm gives the matching pixel in the right image, from which a square template is recorded as the right template for the point. Temporal tracking uses a prediction-and-verification scheme for each of the left and right sequences. The temporal interframe displacement is predicted from the previous displacement of the point. Verification is then performed based on the value of template matching: Let $t(i, j)$, $-s \leq i, j \leq s$ be the template, and $f(i, j)$ denote the image value at the point (i, j) . A template

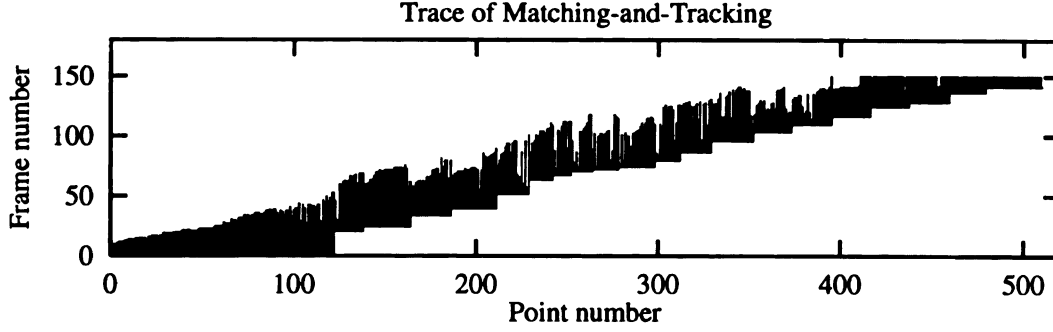


Figure 3.8: The tracking record of the feature points through the 151 frames in the sequence. If a point k is successfully tracked from frame i to frame j , a vertical line is shown at point number k from frame i to frame j . (Due to the limit of the printer resolution, lines are merged in the plot.)

difference value centered at (x, y) is defined by

$$d(x, y) = \sum_{i=-s}^s \sum_{j=-s}^s |[t(i, j) - \bar{t}] - [f(x + i, y + j) - \bar{f}(x, y)]| \quad (3.45)$$

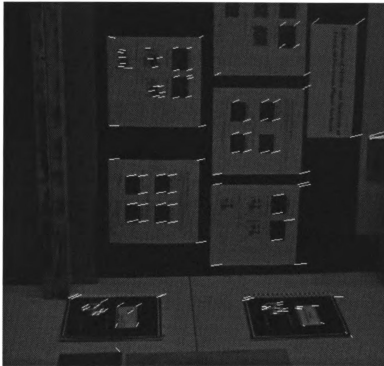
where $\bar{t} = \sum_{i=-s}^s \sum_{j=-s}^s t(i, j)$ and $\bar{f}(x, y) = \sum_{i=-s}^s \sum_{j=-s}^s f(x + i, y + j)$. Namely, first, the template and image are both locally normalized so that their local sum is equal to zero, and then the template difference value is the sum of the absolute difference between the locally normalized template and locally normalized image. The best matching point is the pixel at which the difference value reaches the minimum in a small neighborhood (5×5 in our experiment) centered at the predicted position. A point becomes inactive if the best matching point exceeds the allowable template difference value.

This temporal matching and tracking method was very successful. The trace record of the entire sequence is shown in Fig. 3.8. About 100 feature points were automatically kept at any time. Since some points may go out of view and some

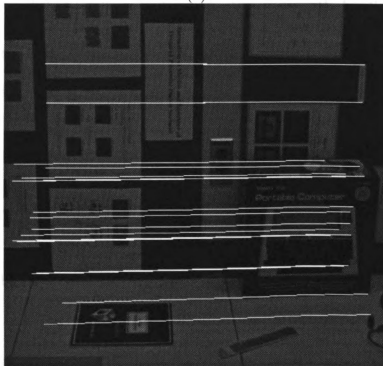
points may become inactive, the number of active points may fall below a tolerable number (90 in our experiment). If this happens, the feature detector is called which provides additional points from the image and then the stereo matcher is called to give stereo matching for these new points. The time when the feature detector was automatically called can be clearly identified in Fig. 3.8. Many points have been successfully tracked until the time they went out of the view. Figure 3.9 presents an example of temporal matching-and-tracking. A careful visual inspection of entire point trace indicates that there was no visible errors.

The measurement of the real setup is similar to the simulation. To verify the accuracy of structure estimates as well as camera pose estimates, the global coordinates of a set of test points were carefully measured to within an error of 1mm. The selection of test points were based on ease of measurement and was not based on automatically selected features. Thus, each test point is not necessarily a part of the feature points used for the automatic algorithms, although many of them are. The image coordinates of the test points are manually measured from digital images. The accuracy of the reconstructed structure error was measured by the following steps.

- (a) Compute the WC and CC representations for feature point position and camera pose using the fully automatic algorithm described above.
- (b) Manually measure the image coordinates of the test points.
- (c) Perform a multiframe triangulation to get the 3-D position of the test points. The number of frame used is according to the corresponding batch size.
- (d) Measure the global position error as the difference between the true and estimated test points. This way of measuring error tests not only the pose of the camera, but also some of the reconstructed feature points, if they are



(a)



(b)

Figure 3.9: Stereo matching and temporal matching-and-tracking. (a) An example of stereo matching (frame 0). (b) An example of temporal matching and tracking (frame 24 to 69). A needle is drawn from the feature point to its position in the target frame. Due to camera vergence, the orientation of the needles in (a) is correct.

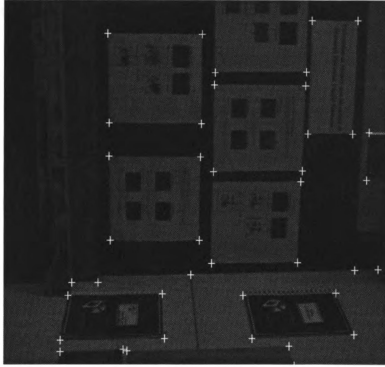


Figure 3.10: Sample test points on one frame. Each cross shows the location of a test point also test points. Table 3.2 lists some data of the real setup. Fig. 3.10 shows the test points of one frame.

Table 3.2: Some Data for The Real Setup

Number of frame	31	Distance traveled (mm)	3097
Number of feature points	387	Number of test points	85

First, to show how well the estimated structure and camera pose agree with the automatically detected feature points, the estimated 3-D feature points were projected onto the image plane according to the pose. The average distance difference between every projected point and actually detected feature point is called image plane residual and is shown in Fig. 3.3. The values are around a fraction of a pixel for both representations. These numbers also indicate that camera distortion compensation

in the calibration was very effective.

Table 3.3: Average Image Plane Residual

Representation	batch size 0	batch size 3
WC	0.76 pixels	0.68 pixels
CC	0.45 pixels	0.51 pixels

Fig. 3.11 shows the actual camera orientation error. Although the image sequence here is transitory, the pitch and row errors are comparable with those in the non-transitory "Hotel" sequence experiment by Tomasi and Kanade [178] over the entire sequence. The visibility span of our setup is about 4. At frame 4, the yaw error has the same magnitude as that in [178]. After frame 4, the error tends to increase due to the transitory nature of the sequence. It is interesting that roll and pitch errors did not increase quickly with time. After traveling about 3000mm, the total orientation error is not more than 0.02° in roll, 0.23° in pitch and 2° in yaw with the WC representation.

Fig. 3.12 shows the camera position error and Fig. 3.13 presents the global error of the test points visible at the current time. As we predicted, the error increases with the time. But the estimates appear good. After traveling about 3000mm, the estimated camera global position error is less than 60mm in depth Z (less than 2.3%), about 43mm horizontally and under 25mm vertically with the WC representation. This seems to indicate that reasonable results can be obtained with a fully automatic algorithm, even with a transitory image sequence.

Fig. 3.14 shows the reconstructed 3D surface. The surface detail is described by mapping intensity of the images onto the reconstructed surface. This approach is

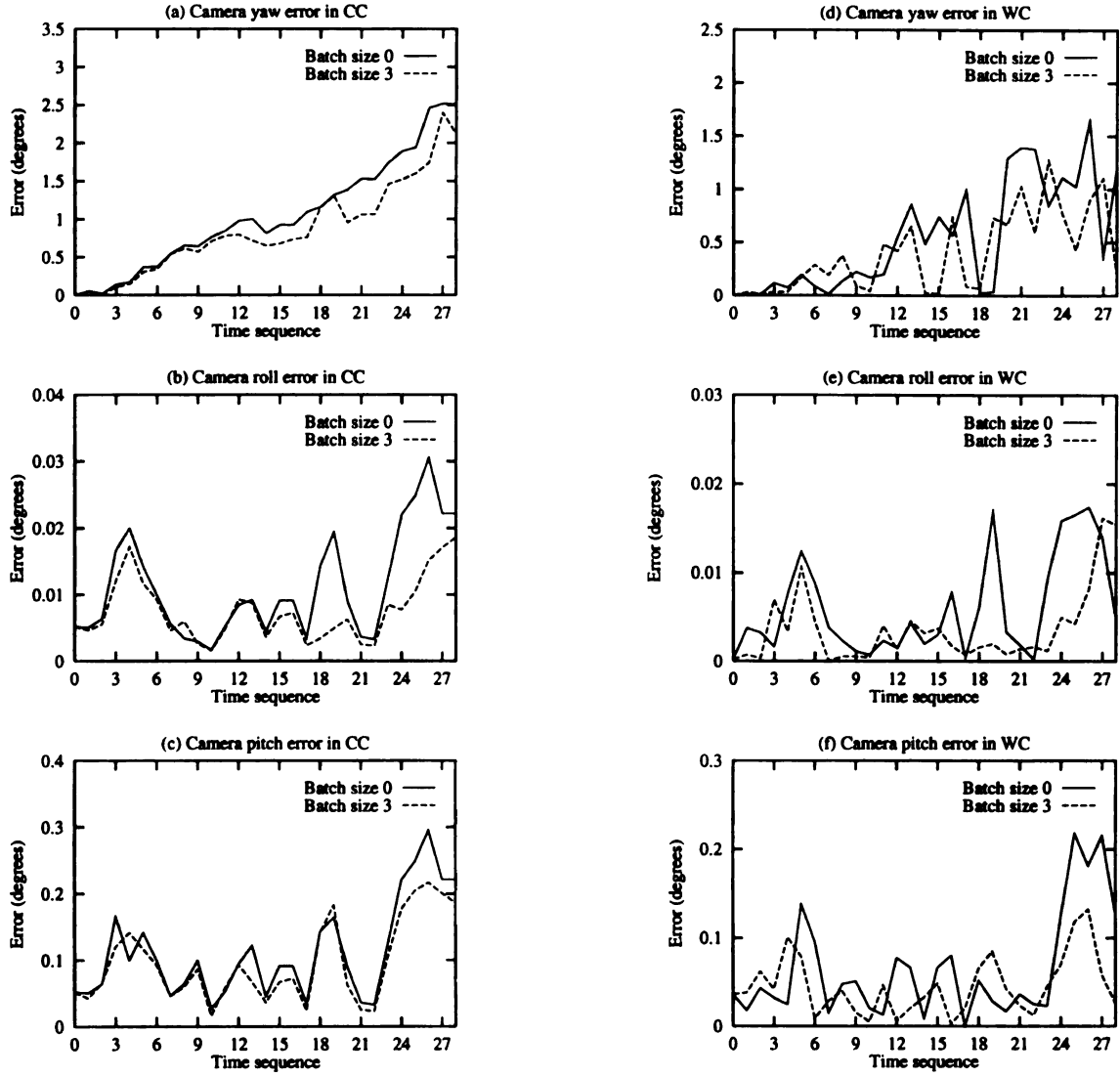


Figure 3.11: Camera rotation error versus time. (a) and (d): Yaw error. (b) and (e): Roll error. (c) and (f): Pitch error.

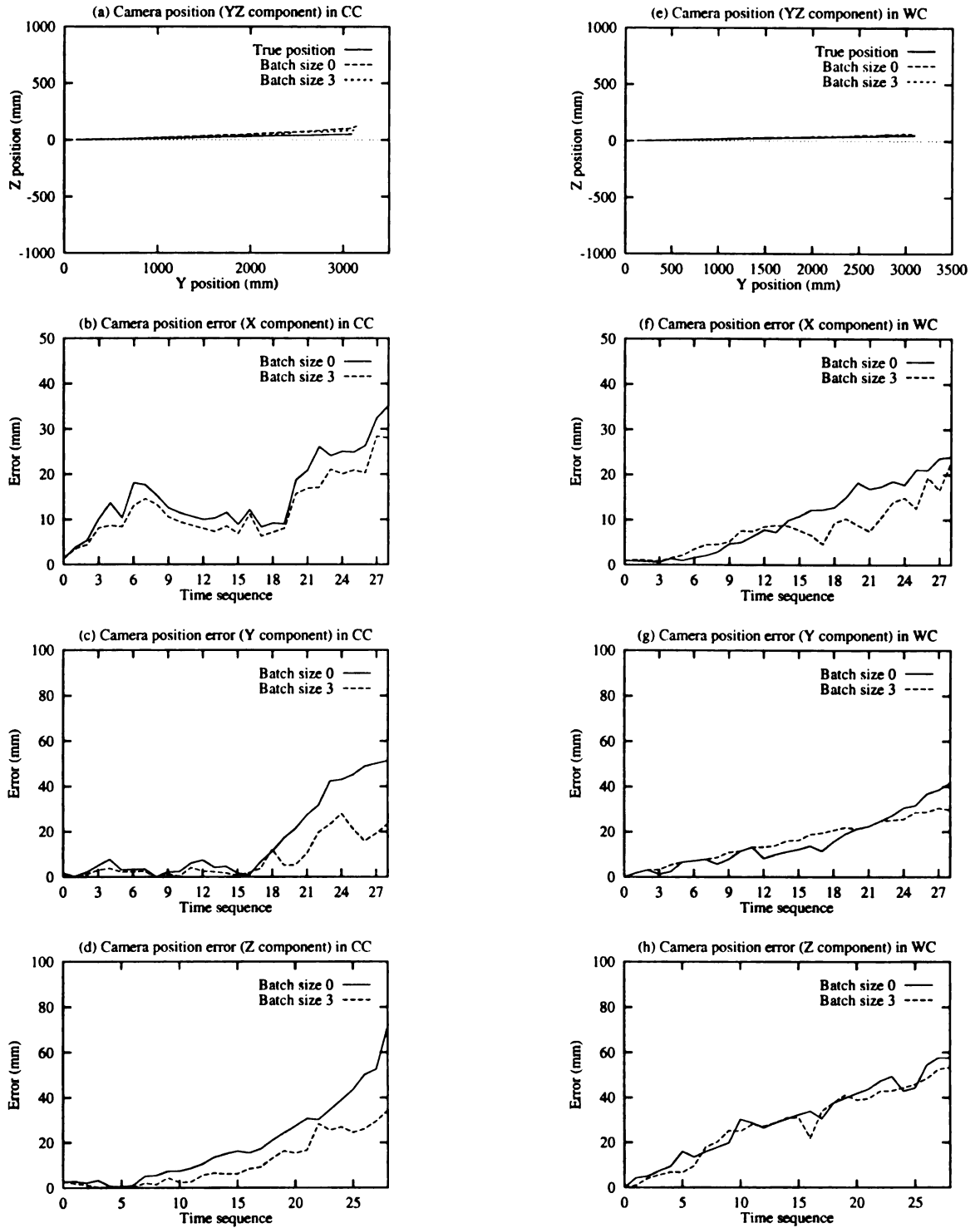


Figure 3.12: Camera position error versus time. (a) and (e): Camera position (y - and z -components). (b) and (f): Position error (x -component). (c) and (g): Position error (y -component). (d) and (h): Position error (z -component).

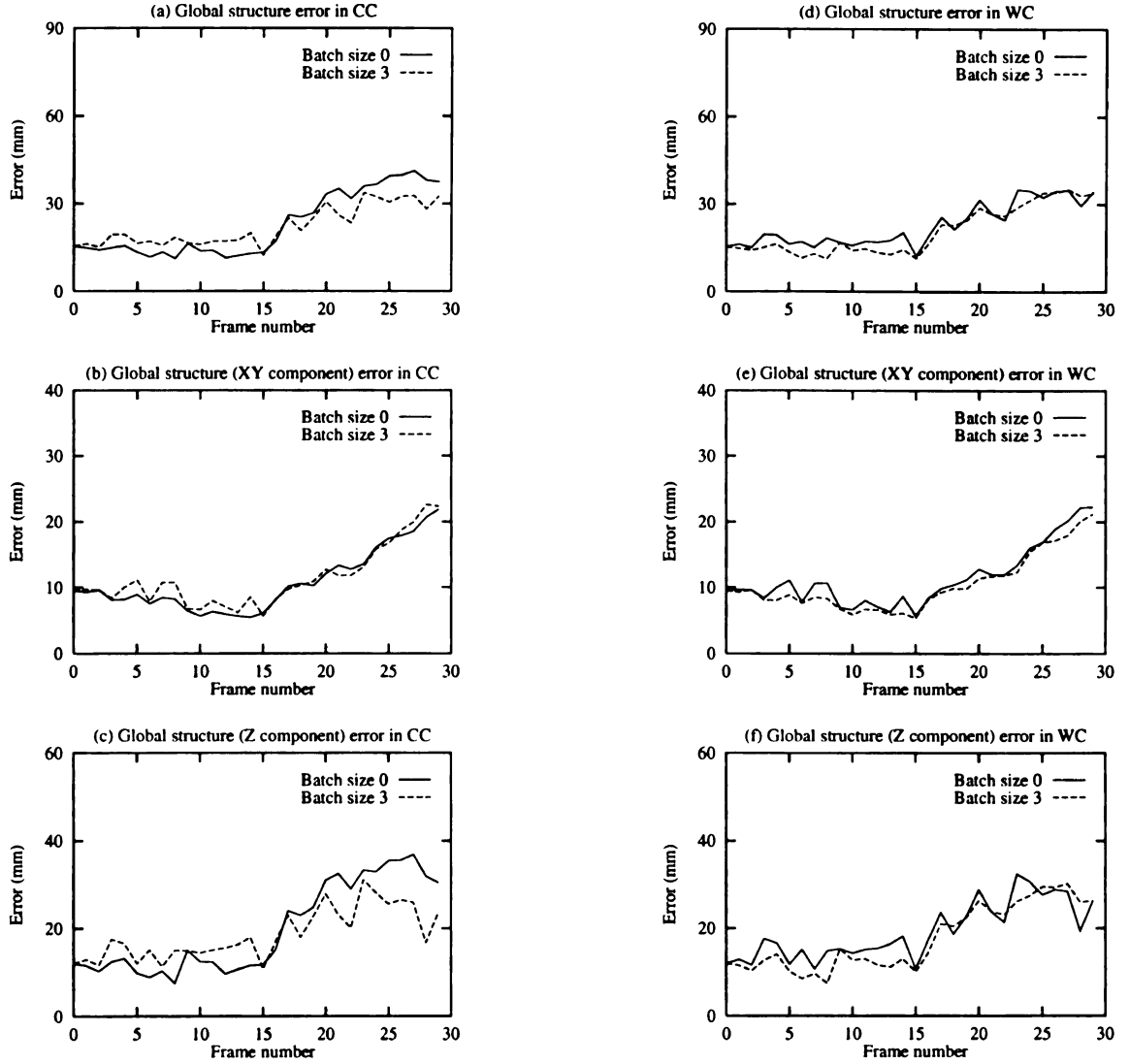


Figure 3.13: Structure error. (a) and (d): Global structure error. (b) and (e): Global structure error (x - and y -components). (c) and (f): Global structure error (z -component).

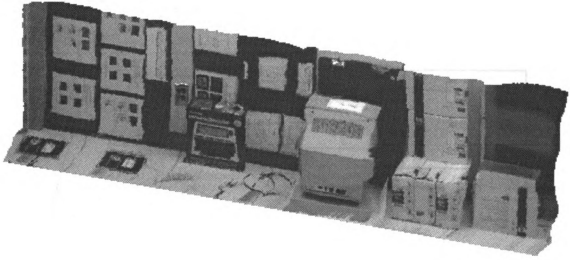


Figure 3.14: Reconstructed 3D surface integrated from many partial views in the sequence, shown with original intensity viewed from an arbitrary direction.

known as *texture mapping* or *pattern mapping* [17]; the image is called a *texture map*, and its individual elements are often called *texels*. The mapping is shown in the Fig. 3.15. The texture map of the tetrahedra of the reconstructed surface is assigned the average intensity value of its vertices in the images. In our transitory case, multiple images may corresponds to the single patch of the surface. A weighted method is used,

$$I_P = \sum_{i=1}^k w_i I_{P_i} \quad (3.46)$$

where the weights w_i 's are the inverse of the structure uncertainty of the points and I_{P_i} is the intensity value of the point P at i th-image.

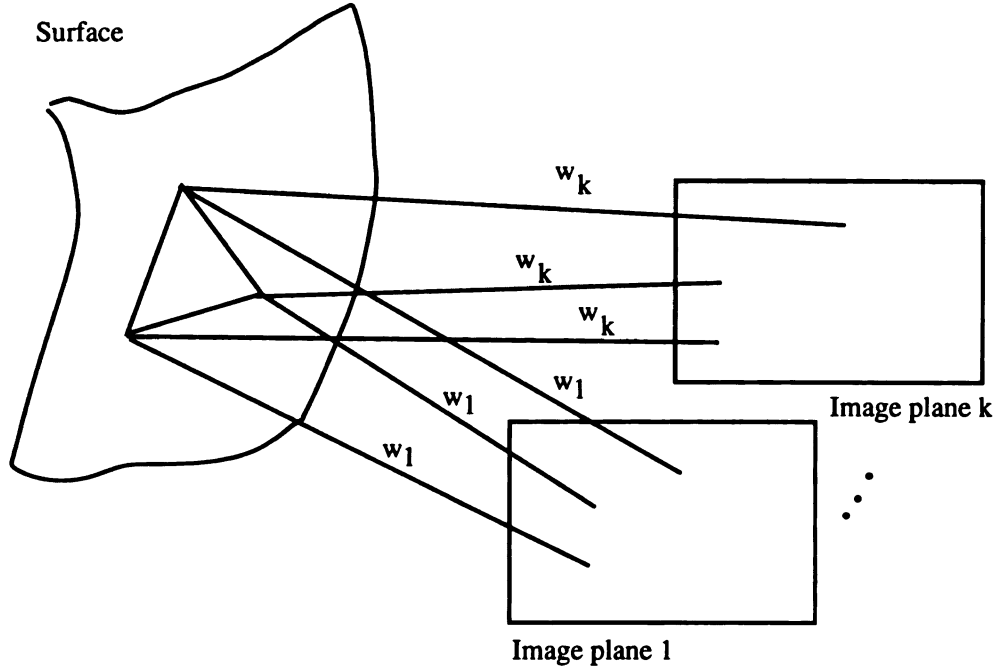


Figure 3.15: Weighted texture mapping from pixel to the reconstructed surface.

3.5 Conclusions

In this article, we introduced the concept of transitory image sequence for structure and motion estimation from long image sequences. It has been shown that integration for transitory sequence has asymptotic error rates that are very different from those with a non-transitory one. The theoretical error rates listed in Table 3.1 indicates that the WC representation is better for global estimates and the CC representation is superior for local estimates.

Based on the analysis, our algorithm keeps both the WC and CC representations. The data shows that the error in the local structure is effectively reduced by a relatively larger batch size and does not increase with time. The global pose of the camera and global structure of the scene is better estimated by a WC representation. Those properties are consistent with the error rates summarized in Table 3.1,

but our sequence is still not long enough to clearly duplicate the theoretically proved asymptotic rates.

The experiment was conducted using a fully automatic algorithm and the accuracy of the result has been verified using the ground truth. The verified accuracy appears to indicate that with off-the-shelf cameras, one can automatically determine the scene structure and pose of the camera with a good accuracy (the depth error is less than 3% compared with the traveling distance), although the image sequence here is of a more different transitory type (compared to the non-transitory one).

Chapter 4

Hand Sign Recognition

Humans have the capability to interpret hand gestures. The study of how humans use and interpret hand gestures has a long and rich history. The first known dictionary of American Sign Language (ASL) was published in 1856 [29]. Today, American Sign Language is widely used in the deaf community, as well as by the people who are not deaf [22].

Recently, there is a significant amount of research which uses hand gestures in the field of human machine interface. There are two types of hand gestures: static hand gestures and dynamic hand gestures. Static gestures are determined by a particular hand posture while dynamic hand gestures are characterized by a dynamic process which includes the initial, the intermediate, and the final hand configuration. Systems which are designed to handle dynamic hand gestures have the full potential of understanding human gestures.

Two major types of approaches exist in the field of hand gesture recognition. The first approach uses glove-based input devices. Glove-based devices employ mechanical

or optical sensor attached to a glove that transduces finger flexion and abduction into electrical signals for the purpose of determining the hand posture. The second approach is the vision-based approach. This approach acquires visual information of a hand gesture by using a single video camera or a pair of cameras. Once the visual information is acquired, the sign is extracted by analyzing the temporal image sequence. In this chapter, we give the literature review of existing hand gesture recognition systems.

4.1 Glove-Based Systems

Glove devices measure the shape of the hand as the fingers and palm flex. Over the past decade, especially in the last few years, many researchers have built hand and gesture measuring devices for computer input. In this section, we briefly describe some significant ones.

4.1.1 Glove devices

In the early 1980s research at MIT used a camera-based LED system to track body and limb position for real-time computer graphics animation, termed “scripting-by-enactment” [74]. This work included a glove studded with LEDs. By focusing the camera on just the hand, they captured finger motion.

Zimmerman *et. al.* developed the DataGlove that monitored 10 finger joints and the six degrees of freedom of the hand’s position and orientation [206]. Commercialization of the DataGlove by VPL Research, at a reasonable cost led to its widespread

use around the world. Most DataGloves have 10 flex sensors, but some have been made with abduction sensors that measure the angle between adjacent fingers.

Kramer and Leifer [107] developed the CyperGlove at Stanford University. It is a custom-made cloth glove with up to 22 thin foil strain gauges sewn into the fabric to sense finger and wrist bending. A small electronics box converts the analog signals into a digital stream that can be read by a computer's standard serial port.

4.1.2 Interpreting hand sign with gloved-based devices

Several projects have investigated various levels of recognizing hand signs from simple finger spelling to analysis of American Sign Language. The MIT Media Lab used their LED glove as part of an experimental system for finger-spelling using lookup tables in software to recognize finger postures [83].

Kramer and Leifer used the CyberGlove to translate ASL into spoken English [107]. They used a Bayesian decision rule-based pattern recognition scheme to map finger positions, represented as a "hand-state vector", into predefined letters or symbols. When the instantaneous hand-state lay close enough to a recognizable state, the corresponding ASL letter or symbol was put in an output buffer. When a word phrase was completed, a special sign caused the result to be spoken by a voice synthesizer.

ATR Research Labs in Japan developed a coding scheme to allow computer recognition of the Japanese kana manual alphabet [173]. Their system used the DataGlove to capture hand posture. It recognized signs through a combination of principal component analysis (to determine the contributions of each finger joint to the differences

between signs) and cluster analysis (to group hand configurations).

Fels used a DataGlove to interpret hand motion to drive a speech synthesizer [66]. His particular approach used a three-stage back-propagation neural network trained to recognize gestural “words”. He divided hand motions among 66 finger positions and 6 hand motions. Finger positions defined the root word, while hand motions modified the meaning and provided expression. These combined to form the 203 words of his “language”, loosely based on conventional gestural languages. Fels reported a high recognition rate once the system was fully trained.

4.2 Vision-Based Approach

The use of computer vision makes it possible to sense human communication unobtrusively and enables human users to interact with computers in a truly natural fashion. There are two major problems which are needed to be solved in vision-based approaches. The first problem is segmentation of the moving hand from sometimes complex background. The second problem is recognition. This involves modeling the hand and the hand motion and then recognition of the gesture.

4.2.1 Segmentation

Segmentation is a very difficult problem. Many existing hand gesture recognition systems avoid this problem by 1) using markers or marked gloves [38, 54, 168]; or 2) assuming uniform backgrounds [19, 46, 53, 110].

In dealing with dynamic gestures and assuming that the hand is moving in a

stationary environment, we can use motion as a visual cue to do segmentation. Several motion segmentation methods have been proposed. These approaches fall into two categories. Approaches in the first category are designed to deal with rigid moving objects (e.g. [24, 56]). This type of approaches achieves a segmentation by either building a reference image of the static background [56], or extracting the motion entity based on 3-D motion models or 2-D velocity-field models [24]. Since hands are highly articulated and non-rigid objects, the above approaches are not suitable for hand segmentation.

The second type of approach fits a shape to deformable objects (e.g. [77, 81, 93, 98, 100, 175]). These models typically need a good initial position to converge. They also need a relatively clean background since the external forces are defined by the image gradient.

There are two classes of deformable models, namely, free-form models and parametric models. In the free-form models, there is no prior information of the global structure of the template; the template is constrained only by local continuity and smoothness constraints [81, 98]. Due to lack of prior information about the global structure, this type of approach relies heavily on good initial positions. When this approach is applied to track moving deformable objects, it requires small interframe deformation to converge. This means that an increase in the sampling rate is needed to capture the deformation in detail since the change of the hand configuration can be dramatic within a single gesture. The computational cost increases as more frames are used to represent a hand sign.

On the other hand, the parametric deformable models use some prior informa-

tion of the geometrical shape of the object. There are some efforts in this category which try to locate hands from input images [77, 93, 100]. Typically, the parametric deformable model includes a prototype template and a deformation model based on a small set of parameters. The final solution is found by minimizing certain type of objective function.

Grenander *et. al.* used a polygon to represent the contour of a human hand [77]. The deformation is described by Markov processes on the edges. A similar scheme was used by Kervrann and Heitz [100] in their work on locating hands from image sequences. The “mean shape” is determined by the salient points of the hand contour and is obtained by the training samples. The deformations, given the “mean shape”, are modeled using linear combinations of the eigenvectors of the variations from the mean shape. Jain *et. al.* used a bitmap to represent the prototype template [93]. The template is then deformed to fit salient edges in the input image by applying a probabilistic transformation on the prototype contour which maintains smoothness and connectedness.

The major drawback of the above approaches is that they only allow very limited deformation. Limited deformation results because the deformation is modeled as a small perturbation of the prototype [100] or because the penalty term is presented in the objective function when the template deforms from the prototype [93]. This drawback makes the model unsuitable for dealing with dynamic gestures where typically deformation is large.

Recently, Moghaddam and Pentland [133] presented a maximum likelihood detection method to locate hands in a cluttered scene. The method uses the training

data to estimate the density of a Mixture-of-Gaussian model (for multimodal distribution). These probability densities are then used to formulate a maximum likelihood estimation framework to detect hands. The training set of hand shapes is obtained against a black background and the contour of the hand is extracted using a Canny edge-operator. Then, a diffusion process is applied to these binary edge maps to broaden and smear the edges. Finally, they are projected to the eigenspace. The density estimation is in the eigenspace instead of the original high-dimensional image space. The major drawback of this method is that on certain illumination condition, edge operator may fail to pick up some hand contour edges. In that case, the system can fail to locate hands.

4.2.2 Recognition

Existing approaches typically include two parts, modeling hands and analysis of hand motion. Models of the human hand include the fingertip model, the three dimensional model, the two dimensional shape model, and the region-based model. Different hand models lead to different models of hand motion. The trajectory of each finger tip is suitable to represent the motion in the case when the fingertip is used to model hands. The system which uses the three dimensional hand model is capable of modeling the real hand kinematics. The two dimensional hand model can describe two dimensional rotation and translation. For the system which uses the region-based model, motion can be modeled as the change of state.

Fingertip model

Cipolla, Okamoto and Kuno [38] presented a real-time structure-from-motion method in which the 3D visual interpretation of hand gestures is used in a man-machine interface. Consider an arbitrary coordinate system with the $x - y$ plane spanning the image plane (f from optical center) and the z -axis aligned with the ray. Assume the fingertip to have a translational velocity with components U_1, U_2, U_3 and an angular velocity with components $\Omega_1, \Omega_2, \Omega_3$. The two components of the image velocity of a point in space, (X, Y, Z) due to relative motion between the observer and the scene under perspective projection are given by [119]:

$$\begin{aligned} u &= \left[\frac{fU_1 - xU_3}{Z} \right] + f\Omega_2 - y\Omega_3 - \frac{xy}{f}\Omega_1 + \frac{x^2}{f}\Omega_2 \\ v &= \left[\frac{fU_2 - yU_3}{Z} \right] - f\Omega_1 + x\Omega_3 - \frac{xy}{f}\Omega_2 - \frac{y^2}{f}\Omega_2. \end{aligned} \quad (4.1)$$

The second component depends only on rotational motion about viewer center. It gives no useful information about the depth of the point or the shape of the visible surface. The rotational component can be removed if, instead of using raw image motion the difference of the image motion of a pair of points, is used. This is called *motion parallax*. The parallax motion vector, divergence, curl, and deformation components of the affine transformation of an arbitrary triangle, with the points at each vertex, determine the projection of the axis of rotation, change in scale, and cyclotorsion.

Davis and Shah [54] have created a system to recognize a sequence of multiple gestures. The library of gestures includes seven gestures. The user must start in

the designated start position upon initialization of the system and is able to make gestures until the termination gesture is recognized by the system. The *moving light display* is obtained by extracting finger tips from each frame. A moving light display labeled by Rashid [156] is a sequence of binary images representing points from a moving object. The positions of the finger tips are marked by the special gloves. The trajectory of each finger tip is found by minimizing the overall proximal smoothness function minimized as much as possible in addition to being fair to each individual assignment [153]. The success of the algorithm needs assumptions of smooth motion in the sequence and small motion between consecutive frames. The system is generally unable to handle occlusion. A finite state machine is used to guide the flow and recognition of gestures based on the motion characteristics of the hand. Owing to the nature of the machine, no warping of image sequences is necessary (i.e. it is not required to have a fixed number of images for each gesture sequence).

Three dimensional model

Kuch and Huang [110] used cubic B-splines to represent the surfaces of the palm, fingers, and thumb. The use of B-splines allows the rendering of smooth surfaces, while allowing the calibration system to keep track of a smaller set of control points versus every vertex in the model. The hand model has a total of 23 degree of freedom (DOF). Each four finger is given four DOF while five DOF is given to the thumb. The palm is given two internal DOF located at the base of the forth and fifth (ring and pinky) metacarpals. The last two DOF reflect the ability of the palm to fold or curve. These DOF determine the overall orientation in space of the entire hand.

The model is used to fit a real human hand, so it can be used in tracking and can be incorporated into a virtual environment or model based compression scheme such as sign language communication over phone line. The model needs to be calibrated before it can be used. The calibration is interactive. It requires three specific views of the real hand. The interactive selection is to locate all the joints and to delineate the portion currently being fitted from the background. Each tracking starts from a person holding his hand in a predefined orientation within the field of view of a camera looking at an uniform background. The DOF of the hand model is then independently and locally perturbed to fit the moving hand.

Two dimensional model

Starner and Pentland used Hidden Markov models (HMM's) to recognize American Sign Language. An eight element feature vector consisting of each hand's x and y position, angle of axis of least inertia, and eccentricity of bounding ellipse is chosen to represent the hand shape. The eccentricity of the bounded ellipse was found by determining the ratio of the square roots of the eigenvalues that correspond to the matrix

$$\begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$$

where a , b , and c are defined as

$$a = \int \int x'^2 dx' dy'$$

$$b = \int \int x'y' dx' dy'$$

$$c = \int \int y'^2 dx' dy'$$

(x' and y' are the x and y coordinates normalized to the centroid.) The work assumes that the order of words in American Sign Language is a first order Markov process. The topology of the HMM used in the paper is a four state HMM with skip transitions determined to be sufficient for the task. Six personal pronouns, nine verbs, twenty nouns, and five adjectives are used in the experiment. These words can form a sentence, however, the structure of the sentence is fixed.

Region-based model

Sometimes the extraction of precise hand and motion information from image sequences is not desirable or it is very difficult if not impossible. In these instances, we can use features generated from the entire image or a relatively large region of the image. These features are called region-based features.

Darrel and Pentland [53] have adopted a view-based representation for learning, tracking and recognizing human gestures from a sequence of images. The method uses an automatic view-based approach to build the set of view models from which gesture models will be created. The model views of an object are built using normalized correlation. The first view is chosen by the user as one of the images from a sequence. The object in the subsequent input images is tracked, and when the correlation score r_m drops below a predetermined threshold, a new model view is created with the current input image. This process is repeated until no more models are necessary.

Once all the views of an object have been gathered, gesture models need to be created. A gesture is a set of views over time. A gesture will be correlated with each stored view of the object (the hand), and the score plotted, for each view, with respect to time. Several examples of the same gesture are used, and the mean $g_m(t)$ and variance $\sigma^2(g_m(t))$ of the correlation scores with respect to model view m will be used to represent that particular gesture \mathbf{g} . To compare a new input gesture, each frame of the new sequence is correlated with a model view. The score results for the whole sequence is plotted with respect to time. The view-based approach is capable of modeling complex, articulated objects for which no simple 3-D model or recovery method is available. The models can be learned by observation rather than needing precise CAD models. The drawback of view-based system is that complex, articulated objects (such as hands) have a very large range of appearances, making traditional approaches to view-based matching difficult. The use of grey level correlation can also be highly sensitive to noise.

Bobick and Wilson [19] considered each hand image in a sequence as a point in the eigenspace. The eigenspace is computed from the training hand images. The sequence of a hand gesture is then defined as an ordered sequence of fuzzy states in the eigenspace. The fuzziness is defined by the variance of the points that fall near it. For a given gesture, these states are used to capture both the repeatability and variability evidenced in a training set of example trajectories. The states are positioned along a prototype of the gesture, and shaped such that they are narrow in the directions in which the ensemble of examples is tightly constrained, and wide in directions in which a great deal of variability is observed. They used Hastie and

Stuetzle [84]’s “principal curves” to compute a prototype trajectory of an ensemble of trajectories. The method is demonstrated at a relatively low dimensional eigenspace (3D). Three eigenvectors may be enough for a simple hand sign such as waving hand in the paper, but for complex signs which have large amount of hand configuration variation, three eigenvectors may not be enough to capture the variance of the pixel intensity values of the training frame. With sparse and high dimensional data, the performance can degrade.

Chapter 5

Overview of the Approach

A hand gesture is a *spatiotemporal event*. A spatiotemporal event involves an object of interest and motion of the object. In the linguistic description of American Sign Language, Stokoe used a structural linguistic framework to analyze sign formation [171]. He defined three “aspects” that were combined simultaneously in the formation of a particular sign - what acts, where it acts, and the act. These three aspects translate into building blocks that linguists describe as - the hand shape, the location, and the movement.

In this thesis, we present a new framework which will deal with the above three “aspects” of hand signs. There are two major components in our framework. We have a prediction-and-verification scheme to locate hands from complex backgrounds. We also have a spatiotemporal recognition component which combines motion understanding (movement) with spatial recognition (hand shape) in an unified framework.

5.1 Time as a Dimension

A natural way to represent a spatiotemporal event is to consider input image sequence as data in space and time [42, 60] by associating the serial order of the pattern with the dimensionality of the pattern vector. The first temporal event is represented in the plane $t = 0$ and the second temporal event by plane $t = 1$, and so on. The entire spatiotemporal pattern vector is considered as a whole by the framework. Figure 5.1 shows an example in which the hand sign “no” is represented by a spatiotemporal sequence (three images).

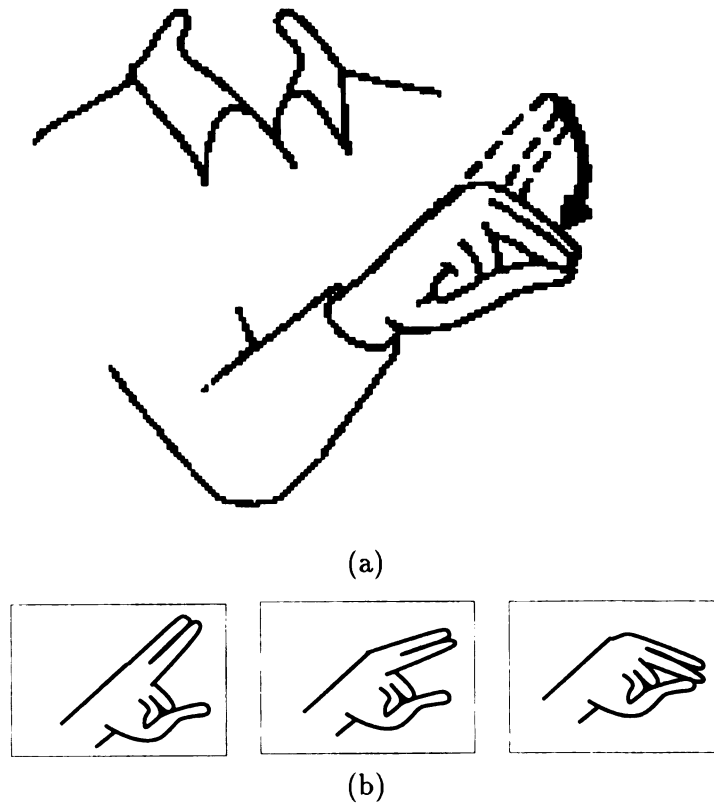


Figure 5.1: The sign “no” and its image sequence representation. (a) The sign of “no”, snap middle finger, index, and thumb together. (b) The sequence representation of the sign “no”.

5.2 Recognition of Spatiotemporal Pattern

As shown in Fig. 5.1, a spatiotemporal event includes two kinds of information: the object of interest, and the movement of the object. The movement of the object can be further decomposed into two components: global and local motions. The global motion capture gross motion in terms of position. The local motion characterizes deformation, orientation and gesture changes. In the case of sign language, the hand is the object of interest. The position change of the hand is a global movement and the change of the hand gesture and orientation is a local movement.

In this thesis, we propose a three-stage framework for spatiotemporal event recognition, as illustrated in Fig. 5.2. The first stage, sequence acquisition, acquires image sequences representing the event. This involves motion detection and motion-based visual attention. The start and end of motion mark the temporal attention window in which the event occurs. We map this temporal window to a standard temporal length (e.g., 5) to form what is called *motion clip*, while the speed information is available from the mapping performed in this stage. In a motion clip, only the temporal dimension is normalized.

The second stage is visual attention and object segmentation. This stage directs the system to focus on the object of interest in the image. Given an image, the object of interest may appear anywhere in the image with certain size and orientation. Besides the object of interest, the image may contain a complex background as well. Therefore, the first step is to determine where to look, or in other words, to select visual attention. If we assume that the object of interest is moving in a stationary

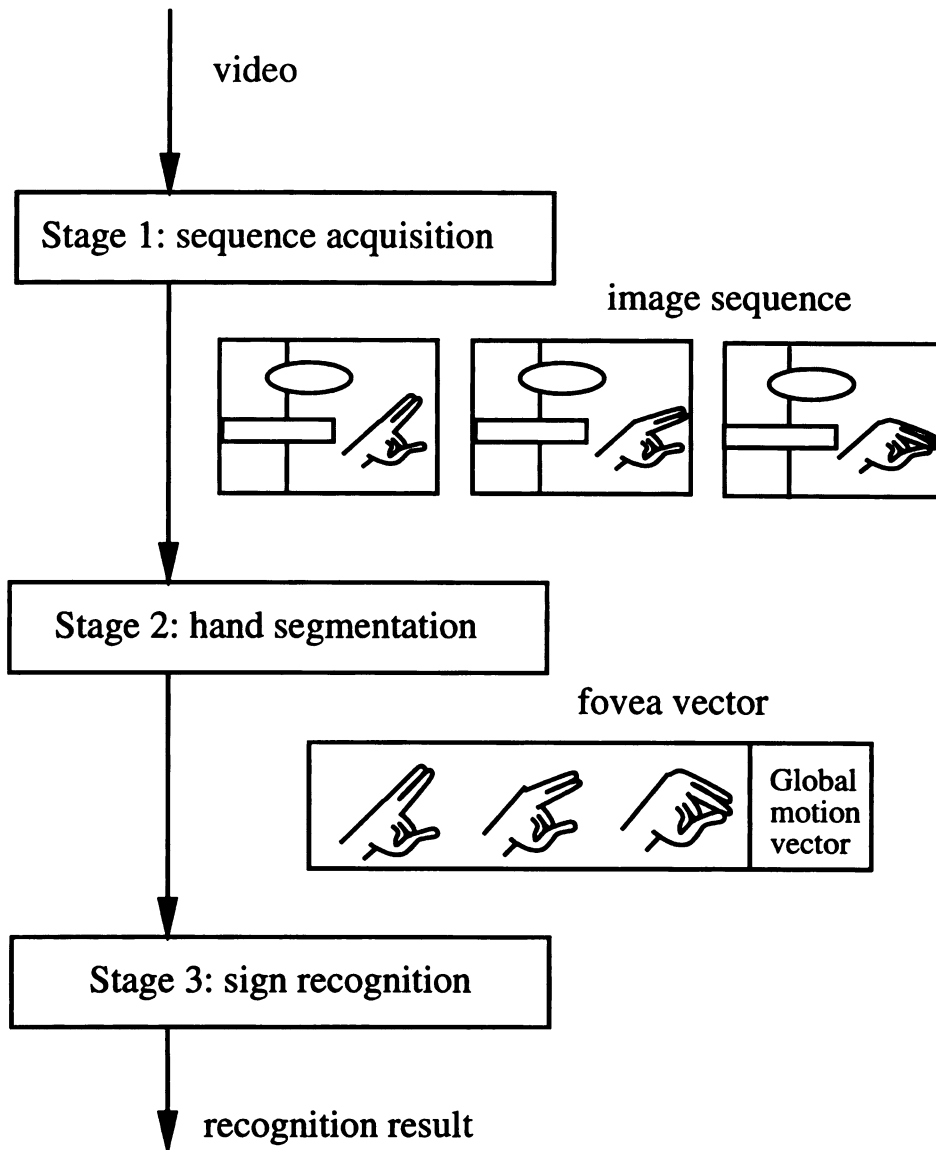


Figure 5.2: The three-stage framework for spatiotemporal event recognition

environment, it is not very difficult to roughly determine the position of a moving object in the image using motion information. However, it is not simple if the task is to extract the contour of the object from various backgrounds.

In Chapter 6, we present an eigen-subspace learning method to segment hands from attention images. In that method, we assume the visual attention is accomplished and the object of interest is centered in the fixed size attention window. In Chapter 7, a prediction-and-verification segmentation scheme is proposed to locate hands from complex backgrounds. The scheme uses the past experience to guide the search of the valid segmentation and is more efficient and effective than other stochastic approaches such as simulated annealing.

After stage two, the object of interest in each image of a sequence is segmented and mapped to a fovea image of a standard fixed size. Segmented fovea images at different times form a standard *spatiotemporal fovea sequence*, in which both temporal and spatial dimensions are normalized. The global motion information of the object of interest is placed in a global motion vector, which records the size and position information of the segmented object in the original image. This vector is necessary because once the object is segmented and mapped to a fovea sequence with a standard spatiotemporal size, the global motion information is lost.

Let a fovea image f of m rows and n columns be an (mn) -dimensional vector. For example, the set of image pixels $\{f(i, j) \mid 0 \leq i < m, 0 \leq j < n\}$ can be written as a vector $\mathbf{V} = (v_1, v_2, \dots, v_d)$ where $v_{mi+j} = f(i, j)$ and $d = mn$. Note that although pixels in an image are lined up to form a 1-D vector \mathbf{V} this way, 2-D neighborhood information between pixels will be characterized by the scatter matrix of \mathbf{V} to be

discussed later. Let p be the standard temporal length and f_i be the hand fovea image corresponding to the frame i . Then we create a new vector \mathbf{X} , called the *fovea vector*, which is a concatenation of the hand foveas and global motion vector G ,

$$\mathbf{X} = (f_1, f_2, \dots, f_p, G). \quad (5.1)$$

The third stage is to recognize the spatiotemporal event from the fovea vector. In Chapter 8, we present a new framework to recognize hand signs from fovea vectors.

Chapter 6

Hand Segmentation from Attention Images Based on Eigen-subspace Learning

Given an image, the object of interest may appear anywhere in the image with certain size and orientation. Therefore, in order to segment the object of interest from the input image, the first step is to determine where to look, or in other words, to select visual attention.

Human's visual attention is accomplished by the rotation of eye [170]. Specifically, it may rotate in a rapid jump-like manner ("saccade") so as to bring the retinal image of an unattended, but eccentric, stationary object to fall at the fovea center. Different visual cues such as motion [24], the generalized symmetry [157], and the casual semantics [16] are used in computer vision to select visual attention. In the

next chapter, we present our own learning-based method to select visual attention and segment hands from images.

In this chapter, we present an eigen-subspace learning method to segment hands from images. We assume the visual attention is accomplished and the object of interest is centered in the fixed size attention window. However, we do allow the background in the attention window. Our goal is to segment the hand from the attention image.

Ever since it was first proposed by Kass *et al.* [98], the snake model has drawn a lot of attention. Various modifications have been proposed to segment objects from intensity images (e.g. [81, 117]). However, in general, these models need good initial position to converge. They also need relatively clean background since the external forces are defined by the gradient. In Malladi *et al.*'s front propagation approach [123], the initial curve can be arbitrary as long as it is either inside or outside the object. This method also suffers when the background (or object) is textured if starting from outside (or inside).

In our work, we also use a dynamic deformation model, called *spring network* [35]. The difference is that we first reconstruct the input fovea image. This reconstruction is based on the learning and has the advantage to preserve the object of interest while blurring the background. Like all the other deformable models, the spring network model has quite a few parameters that need to be tuned. Tuning these parameters is not easy work and these parameters usually depend on the input. Our solution to this problem again resorts to the learning. We first try to get the best results from the spring network, then we go back to the learned examples to find the best match.

Finally, we apply the spring network model again using the best match as the starting curve to get the segmentation result.

6.1 Learning

Given a set of training attention images of hands with black background, we first derive an eigen-subspace using Karhunen-Loeve projection. Next, we transform the training attention images into the *simulated fovea images*. The definition of simulated fovea image will be given later. We store these simulated fovea images in a database. During the testing session, the database is queried to give an initial contour for deformation model.

6.1.1 Karhunen-Loeve projection

Let a training attention image f of m rows and n columns be an (mn) -dimensional vector. For example, the set of image pixels $\{f(i, j) \mid 0 \leq i < m, 0 \leq j < n\}$ can be written as a vector $\mathbf{V} = (v_1, v_2, \dots, v_d)$ where $v_{mi+j} = f(i, j)$ and $d = mn$. Note that although pixels in an image are lined up to form a 1-D vector \mathbf{V} this way, 2-D neighborhood information between pixels will be characterized by the scatter matrix of \mathbf{V} to be discussed later.

Typically an image space is very large. For a moderate 128×128 -pixel image, the dimension is $d = rc = 16,384$. The Karhunen-Loeve projection [121] is a very efficient way to represent a small subspace in a high-dimensional space. It reduces the dimension of representation from d in S to a much lower dimension for S' yet still

keeps most information in the data.

A d -dimensional random vector \mathbf{X} can be expanded exactly by d orthonormal vectors, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$, so that

$$\mathbf{X} = \sum_{i=1}^d y_i \mathbf{v}_i = V\mathbf{Y} \quad (6.1)$$

where V is an orthogonal $d \times d$ square matrix consisting of orthonormal column vectors \mathbf{v}_i . Without loss of generality, we can assume that the mean of the random vector \mathbf{X} is a zero vector, since we can always redefine $\mathbf{X} - E\mathbf{X}$ as the vector to consider. If the set of samples of a class typically occupy a very small portion of the entire d -dimensional space, we can expect that a relatively small number of vectors (or called features) \mathbf{v}_i is sufficient to expand the space of the class. Suppose we use m vectors (features), each corresponds to a component in \mathbf{Y} . The approximate representation is $\hat{\mathbf{X}}(m) = \sum_{i=1}^m y_i \mathbf{v}_i$. It has been proved [121] that the best unit vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ that minimize

$$\epsilon^2(m) = E\|\delta\mathbf{X}(m)\|^2 = E\|\mathbf{X} - \hat{\mathbf{X}}(m)\|^2 \quad (6.2)$$

are the m unit eigenvectors of the covariance matrix Σ_X of \mathbf{X} , associated with the m largest eigenvalues. Let V consists of these m vectors $\{\mathbf{v}_i\}$ as column vectors. Then $\mathbf{Y} = V^\top(\mathbf{X} - \mathbf{M}_X)$, where \mathbf{M}_X is the mean vector of \mathbf{X} , is a projection from a d -dimensional space to a lower m -dimensional space. It is called the Karhunen-Loeve projection. We can choose m so that the ratio $r = \sum_{i=m+1}^n \lambda_i / \sum_{i=1}^n \lambda_i$ is smaller than a given percentage (e.g., 5%). We call these m vectors $\{\mathbf{v}_i\}$ the *most expressive*

features (MEF) in that they best describe the sample population in the sense of linear transform. Fig. 6.1 gives a 2D illustration of Karhunen-Loeve projection to select the MEFs.

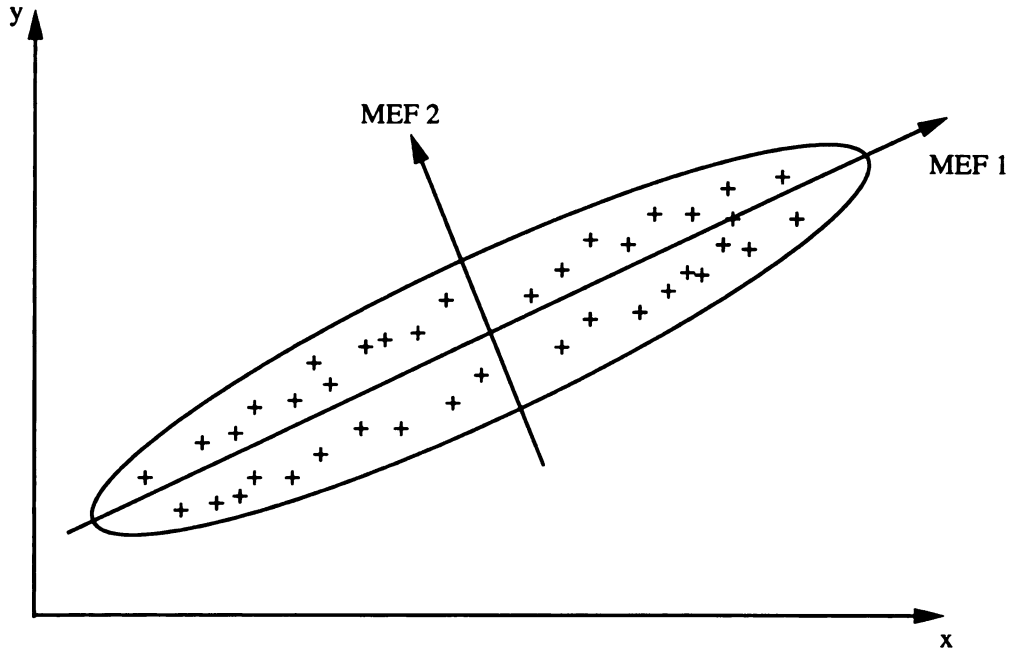


Figure 6.1: A 2D illustration of Karhunen-Loeve projection.

If we are given k discrete training samples, $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k$, Σ_X is approximated by the corresponding scatter matrix:

$$S = \sum_{i=1}^k (\mathbf{X}_i - \mathbf{M})(\mathbf{X}_i - \mathbf{M})^t = UU^t \quad (6.3)$$

where $U = [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_k]$, and $\mathbf{U}_i = \mathbf{X}_i - \mathbf{M}$, $\mathbf{M} = (1/k) \sum_{i=1}^k \mathbf{X}_i$.

In our case, typically $k < n$ and thus, S is degenerate. We can find the eigenvectors and eigenvalues of $k \times k$ matrix $U^t U$, which has the same non-zero eigenvalue as $S = UU^t$. If \mathbf{w}_i is an eigenvector of $U^t U$ associated with the eigenvalue λ_i , then

$\mathbf{v}_i = U\mathbf{w}_i$ is the eigenvector of $S = UU^t$ with the same eigenvalue. Therefore, we just need to compute the eigenvectors and eigenvalues of a much smaller $k \times k$ matrix U^tU . For some earlier works on using MEF for recognition-related problems, the reader is referred to Turk & Pentland 1991 [180] and Murase & Nayar 1994 [136].

6.1.2 Simulated fovea image

It is well known that the resolution acuity of the human fovea varies with eccentricity [115]. The center has the highest resolution. In our case, we know that the hand is centered in the attention image, so it is reasonable to put more weight on the central pixels than the peripheral ones. Here, we simulate human's acuity by transforming the attention image as follows. Let the size of the attention window be $2n \times 2n$ and the center of the attention window be at the row r_c and the column c_c . After the transformation, the intensity value ($I'_{i,j}$) of the pixel at the row i and the column j is

$$I'_{i,j} = \begin{cases} (1 - d/n)I_{i,j} & \text{if } d \leq n \\ 0 & \text{otherwise} \end{cases} \quad (6.4)$$

where $d = \sqrt{(i - r_c)^2 + (j - c_c)^2}$ and $I_{i,j}$ is the original intensity. The new image is called *simulated fovea image* (SFI). The simulated fovea images from the training set are stored in a database for initial contour query during the test.

6.2 Segmentation

During the stage of segmentation, an object is presented in an attention image with complex background and the object is centered in the attention image. Our segmentation scheme includes the following steps:

1. Reconstruct the input attention image based on learned feature values in the eigenspace.
2. Generate the mask from the reconstructed image using the dynamic spring network model.
3. Apply the mask to the input attention image for segmentation.
4. Transform the result of previous step into SFI using equation (6.4) and project the SFI to the eigenspace.
5. Find the nearest neighbor in the training samples as a recognized model.
6. Apply the spring network model again using the contour of the nearest neighbor as the starting curve.

In the above steps, step 3, 4 and 5 are straightforward. Step 6 is simply applying the model. Next, we focus on step 1 and 2.

6.2.1 Reconstruction

Given an image \mathbf{f} with background and a set of eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$, we first obtain the projection coefficients $e_i = \mathbf{v}_i \cdot \mathbf{f}$, where \cdot is the dot product of the two

vectors. Then we reconstruct the image using

$$\mathbf{f}' = \sum_{i=1}^n e_i \mathbf{v}_i \quad (6.5)$$

The image \mathbf{f} can be decomposed into two parts, one is the object of interest with zero background \mathbf{f}_o and the other is the background occluded by the object \mathbf{f}_b . Thus, we have $\mathbf{f} = \mathbf{f}_o + \mathbf{f}_b$. Since our back-projection is linear, we have

$$\mathbf{f}' = \mathbf{f}'_o + \mathbf{f}'_b \quad (6.6)$$

If we choose the m so that after the projection the mean-square error is less than 5%, then statistically the \mathbf{f}'_o has less than 5% loss compared with \mathbf{f}_o if the \mathbf{f}_o happens to be the training sample. The loss for \mathbf{f}_b is expected more severe since in the training stage the background information has never been used. If the \mathbf{f}_o is not the training sample but the training includes some similar objects which can represent this class of objects, the reconstruction is still expected to be effective. Fig. 6.2 shows the reconstruction.

Of course, the degree of background reduction depends on the training samples. If we train the system to segment and recognize the objects which have similar appearance, the reduction will be more obvious.

This kind of background reduction is essential to the success of our dynamic deformation model. If the background has texture, we can pretty much remove it due to information loss in the reconstruction.

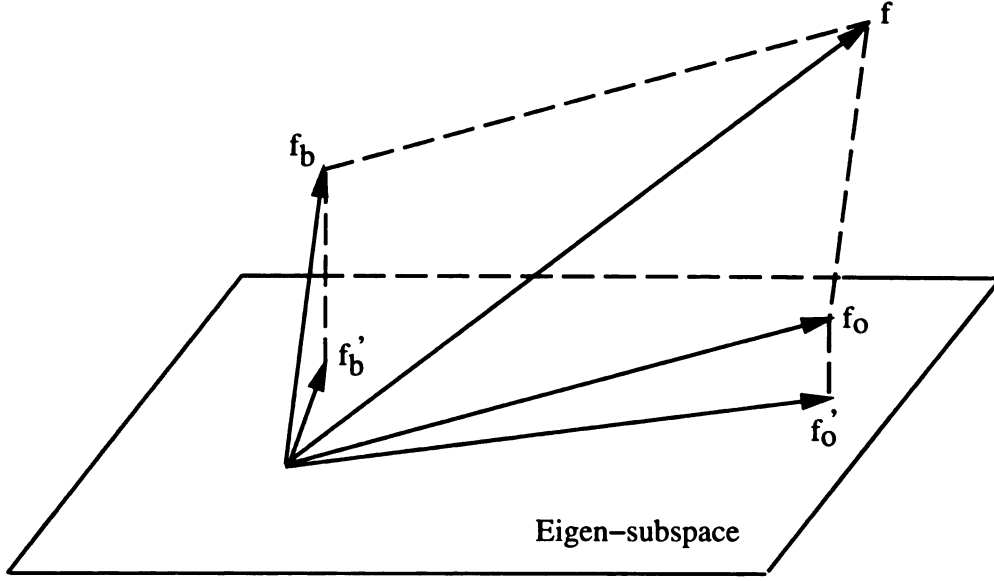


Figure 6.2: An illustration of the reconstruction.

6.2.2 Dynamic Deformation

In this stage, our input is \mathbf{f}' . We want to single out the object of interest from the reconstructed image \mathbf{f}' . First, we apply the Canny edge operator to find the edge map of the \mathbf{f}' . The edge map includes the boundary edges of the object and the number of edges outside the object boundary is minimum since the \mathbf{f}' preserves most of the information of the object and at the same time reduces the background if this type of the object has been presented in the training session.

Second, we apply our dynamic deformation model to obtain the mask. The pixels on edges act as attractors to pull a deformable spring, which is closed in shape and is presumably made of an elastic material. This type of model has been widely used (e.g. [143, 174]). In our case, we use a two-stage deformation mechanism for suppressing undesirable effects appearing in the course of model construction [35]. These effects are the oversmoothing and the local concentration of the deformable

model.

In the first stage of global deformation, globally convex portions of the edge map are recovered. More specifically, the spring network is restricted to contract down to the convex hull of the given pixels on edges. In the second stage, we deform the convex hull to fit the data.

Assume that the vertices of the polygon are interconnected with imaginary springs. The dynamic behavior of this spring network can be described using the equilibrium of forces. Let the spring polygon be $SP = (V, E)$, where $V = \{x_i | i = 1, \dots, n\}$ is the set of vertices, and $E = \{e_i | i = 1, \dots, m\}$ is the set of links. Assume that all nodes have the same mass m , and all springs have the same stiffness k . The motion equation for a single vertex is

$$m \frac{d^2 x_i}{dt^2} + k \frac{dx_i}{dt} + g_i = f_i. \quad (6.7)$$

The g_i is the internal spring force. Define the adjacent set of a node x_i as the set of nodes connected to x_i . The spring force applied to a node is a net force summarizing the individual forces of springs connecting the node and its adjacent nodes. Let c be a constant coefficient, then

$$g_i = \sum_j \frac{c(|x_j - x_i| - l)}{|x_j - x_i|} \times (x_j - x_i). \quad (6.8)$$

The f_i is the external force. In our case, the external force is defined as the gravitational forces between spring nodes and sampled points. Let y be the edge

pixel and S is the collection of all the edge pixels, then

$$f_i = \sum_{y \in S} G \times \frac{m^2}{|y - x_i|^2} \times \frac{y - x_i}{|y - x_i|}, \quad (6.9)$$

where G is the gravitational coefficient.

The explicit Euler time-integration procedure can be used to solve the equation (6.7) [151]. Specifically

$$\begin{aligned} x_i^{t+\Delta t} &= x_i^t + \Delta t v_i^{t+\Delta t} \\ v_i^{t+\Delta t} &= v_i^t + \Delta t a_i^{t+\Delta t} \\ a_i^{t+\Delta t} &= f_i^{t+\Delta t} / m \\ f_i^{t+\Delta t} &= f_i^t - c v_i^t - g_i^t, \end{aligned} \quad (6.10)$$

where Δt is the time step between each iteration, $a_i = \frac{d^2 x_i}{dt^2}$ is the acceleration and $v_i = \frac{dx_i}{dt}$ is the velocity. At each iteration, forces, accelerations and velocity are evaluated. The model becomes stable when $a_i \approx 0$.

The final polygon is used as a mask to mark off the image \mathbf{f} . The marking is simple, we simply assign the zero intensity to pixels outside the polygon. The result of masking is transformed into SFI and then is projected to the eigenspace. Finally, we find a nearest neighbor in the space decomposition tree. This nearest neighbor is our segmentation result.

6.3 Experiments

In our experiments, we have applied the above approach to the task of segmentation and recognition of hands from fovea images.

6.3.1 Training

The system has been trained with 25 classes of different hand shapes. These hand shapes are illustrated in Fig. 6.3. For each hand shape, five or six training samples were used. In the training session, these samples were manually segmented.

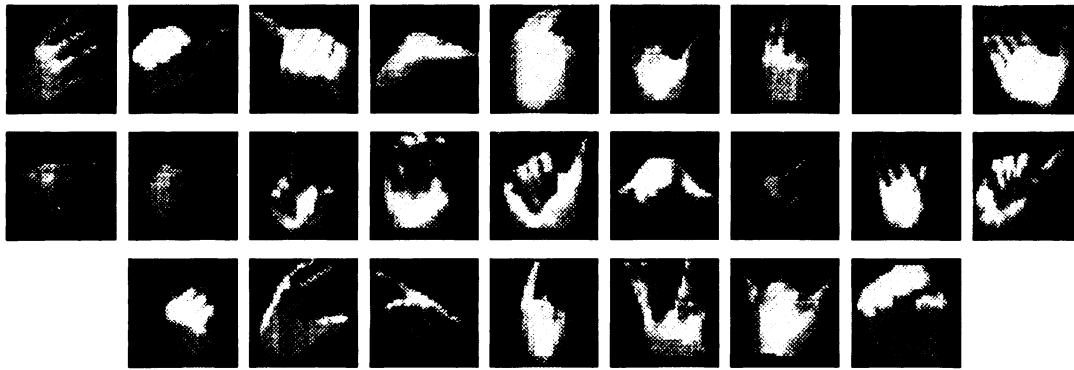


Figure 6.3: Twenty five different hand shapes used in the experiments.

There are two steps in the training. First, we derived the eigenvectors of the training samples. In the current implementation, we keep m eigenvectors so that after the projection the mean-square error is less than 5%. Second, we obtained the SFIs based on equation (6.4). Fig. 6.4 shows a few SFIs of the training samples.



Figure 6.4: The SFIs of the training samples.

6.3.2 Testing

We have conducted two types of testing. In the first type of test, we used the training images. However, in the testing, we do not do any manual segmentation. So, this type of testing images combines the hands presented in the training and the backgrounds. In the second type of test, a new set of fovea images was used. The segmentation routine for test image \mathbf{f} is outlined as follows.

Outline of segmentation algorithm

begin

- 1) Reconstruct \mathbf{f} using m eigenvectors.
- 2) Build mask using dynamic model.
- 3) Apply mask to \mathbf{f} to get \mathbf{f}' .
- 4) Get SFI (\mathbf{f}'_s) of \mathbf{f}' .
- 5) Reconstruct \mathbf{f}'_s using m' eigenvectors.
- 6) Search the nearest neighbor.
- 7) Adjust the curve using contour of the nearest neighbor as initial guess.

end

In the dynamic model, there is a tradeoff between the shape fidelity and the time complexity when choosing the number of nodes of a spring network. It is natural to think that an object with a complicated shape and large size should require a network having a large number of nodes. Currently, we use an adaptive-size model. If two nodes become too close or too far away, we start to delete the old vertices or add new vertices. We also decrease the time step monotonically with the number of iterations l . Empirically, we define $\Delta t(l) = \alpha l^{-\beta}$, where α and β are positive constants.

For the first type of test, if the nearest neighbor is right, we have the perfect segmentation. The number of images used in the first type of test is 135, the test results

show that we have the correct segmentation for 131 of them. The misclassification is due to background which was not presented in the training. The correct rate is 97%.

For the test using new set of fovea images, we classify the result of segmentation to be correct if the hand shape of the nearest neighbor is similar to the one in the input fovea image. Out of 115 testing images, we have correctly classified 107 of them. The correct rate is 93%. The testing results are summarized in Table 6.1. We show the results of 9 testing images from the second type of test in Fig. 6.5.

Table 6.1: Summary of the segmentation results

Test No.	Number of images	Correct rate
1	135	97%
1	135	93%

6.4 Conclusions

A learning-based segmentation scheme is presented in this chapter. During the training, we use the Karhunen-Loeve projection for the training set to obtain a set of eigenvectors. The eigenvectors are used to reconstruct the test attention image. A dynamic spring network model is used to generate the proper mask. The system is tested to segment hands from fovea images. The experimental results show 97% correct rate for the hands presented in the training and 93% correct rate for the hands that have not been used in the training phase.

The scheme also has its drawbacks. The reconstruction is not going to work well when the attention image is dominated by the background. Imagining we have

an elongated object in the square attention image, in this case the fovea image is dominated by the background. If we try to reconstruct this fovea image, the object can be lost in the reconstructed image. The SFI is also going to fail since it is symmetrical in all directions. We believe in this kind of situation, single fixation can not solve the problem. Like human vision, multiple fixations are needed. We need to examine different parts of the object first, then assemble each of the individual results together to get the global picture of the object.

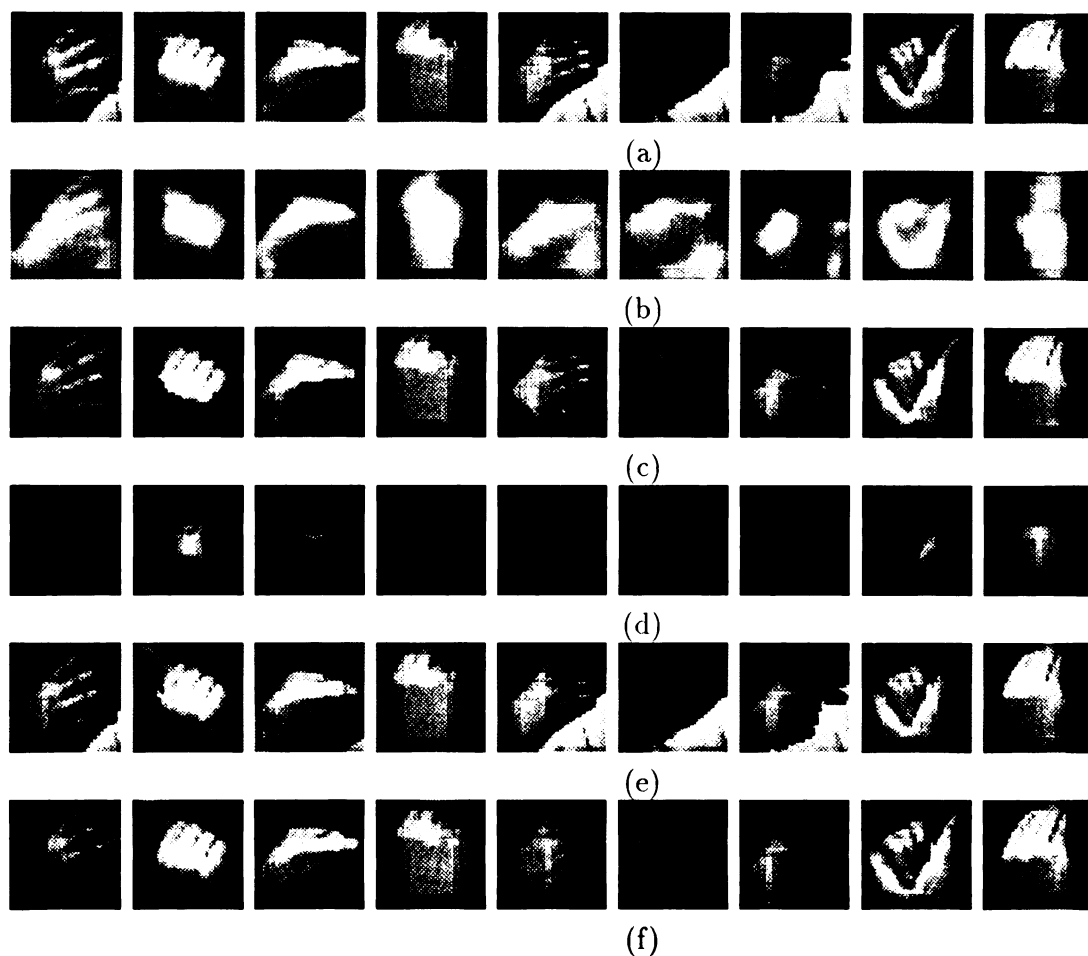


Figure 6.5: Segmentation results of the fovea images which are not used in the training. (a) Input fovea images. (b) The results of the reconstruction. We also blur the results of the reconstruction. (c) The results of applying masks to the original input fovea images. The masks were derived using the dynamic spring network model to the reconstructed images. (d) The SFIs of the images in (c). (e) The contours of the nearest neighbors are superimposed onto the input images. (f) Segmentation results.

Chapter 7

Hand Segmentation Using a Prediction-and-Verification Scheme

In Chapter 6, we presented an eigen-subspace learning method to segment hands from attention images. The approach was motivated by the fixation of human vision. The fixation is the time when the human selects and examines objects from the fovea [170]. In that approach, the object was assumed to position in a rectangular attention image together with the background. The attention image first went through a reconstruction based on features. Then, the reconstructed image was used to predict the segmentation result. Such a reconstruction is based on learning and can reduce the background interference to a certain degree. This type of reconstruction is motivated from studies in psychology. It is well known in psychology that in the retrieval stage, humans may make some kind of reconstructive changes based on past knowledge [161]. However, the reconstruction is not able to fully get rid of the background interference.

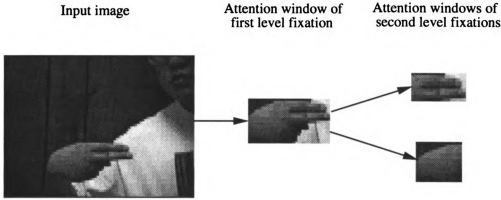


Figure 7.1: An illustration of two level fixations of an input hand image.

One attention window from a single fixation can not solve the segmentation problem completely. Similar to human vision, multiple fixations are needed. This kind of multiple fixations has a hierarchal structure. As shown in Fig. 7.1, the first level of the fixation concentrates on the entire hand, while the next level of the fixation takes care of different parts of the hand. The attention window of the first level fixation usually contains a part of the background. But as we continue zooming in the object from different fixations, the attention windows become focusing on different parts of the object. One important feature of these attention windows is that they typically contain much less background than the attention window of the first level fixation. These attention images from multiple fixations can be used as important visual cues to segment the object of interest from the input image.

In this chapter, we present a new approach which efficiently utilizes the attention images obtained from the multiple fixations through a prediction-and-verification scheme to perform the task of hand segmentation. A general object segmentation system accepts an input image \mathbf{I} and an intention signal \mathbf{P} which specifies the type of object that it is looking for and outputs the segmentation result $\mathbf{C} = S(\mathbf{I}, \mathbf{P})$. To

check the validation of the segmentation result, we need a verifier f . In order to build such a segmentation system, we need to answer two questions: 1) how to find \mathbf{C} ; 2) how to construct f .

We present a prediction-and-verification scheme to answer above two questions in this chapter. First, we introduce the concept of *valid segmentation* and provide a criteria to evaluate whether a segmentation is valid or not.

Secondly, we develop a systematic approach to predict the valid segmentation using attention images of multiple fixations as visual cues. The prediction is based on the nearest neighbor decision rule. It has been shown that the probability of error of the nearest neighbor rule is bounded above by twice the Bayes probability of error [43]. Unlike the Bayesian approach, the nearest neighbor decision rule is independent of the underlying joint distribution on the sample points. In practice, the joint distribution is unknown and in many cases, a normal distribution is assumed. The assumption may be invalid and could lead to poor performance.

A hierarchical quasi-Voronoi tessellation is presented to organize the training samples and propose an efficient algorithm to query the nearest neighbor in high dimensional space. Thus, unlike the exhaustive search method or other stochastic search approaches such as simulated annealing, our search for valid solution is guided by prediction using the past knowledge and is significantly more efficient.

The outline of the chapter is as follows. We dedicate the next sections to the major components of this chapter, namely, verification and prediction. In Section 7.3, we present the experimental results.

7.1 Valid Segmentation

In this section, we define the verifier f mentioned in the previous section. Let the segmentation result of an intensity image \mathbf{I} with r pixel rows and c pixel columns be represented by a vector \mathbf{C} in (rc) -dimensional space, where $\mathbf{C}[i \times c + j] = 1$ if pixel (i, j) in \mathbf{I} belongs to the object, otherwise $\mathbf{C}[i \times c + j] = 0$. Let $P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ be the set of pixels in \mathbf{I} such that $\mathbf{C}[x_i \times c + y_i] = 1$. We denote $x_{min} = \min_{i=1}^n x_i$, $x_{max} = \max_{i=1}^n x_i$, $y_{min} = \min_{i=1}^n y_i$, and $y_{max} = \max_{i=1}^n y_i$.

Definition 1 *An extractor \mathcal{E} extracts a subimage \mathbf{I}' with s rows and t columns from an image $\mathbf{I}_{r,c}$ based on the segmentation result \mathbf{C} , such that $\mathbf{I}'[i, j] = \mathbf{I}[i + x_{min}, j + y_{min}]$ if $\mathbf{C}[(i + x_{min}) * c + j + y_{min}] = 1$, otherwise $\mathbf{I}'[i, j] = 0$ for all $0 \leq i < s$ and $0 \leq j < t$, where $s = x_{max} - x_{min} + 1$ and $t = y_{max} - y_{min} + 1$.*

Intuitively speaking, an extractor extracts a subimage from an image \mathbf{I} according to the segmentation result \mathbf{C} and \mathbf{C} acts like a mask which marks off background.

Definition 2 *A scaler \mathcal{M} maps an image \mathbf{I}' with s rows and t columns to an attention image \mathbf{F} with m rows and n columns, such that $\mathbf{F}[i, j] = g(\mathbf{I}', \frac{i * m}{s}, \frac{j * n}{t})$ for all $0 \leq i < m$ and $0 \leq j < n$, where g is an appropriate antialiasing function which reduces the sampling effect from a digital image to another.*

Using an extractor and a scaler, we can construct an attention image from the result of the segmentation. This entire process is illustrated in Fig. 7.2.

Definition 3 *A verifier f is defined such that, $f(\mathbf{F}, \mathbf{P}) = 1$ if the segmentation result is correct, otherwise $f(\mathbf{F}, \mathbf{P}) = 0$. Here \mathbf{F} is an attention image based on a*

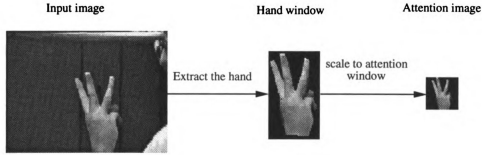


Figure 7.2: The illustration of constructing attention images.

segmentation result \mathbf{C} such that $\mathbf{F} = \mathcal{M}(\mathcal{E}(\mathbf{I}, \mathbf{C}, \mathbf{P}))$, where \mathbf{I} is the input image and \mathbf{P} is the intention signal.

The verifier defined in Definition 3 makes a decision based on the information presented in the attention window. The biggest advantage of using the attention window is that we can achieve size and position invariance with a fixed size attention window.

The function f is extremely complex, because of the high-dimensionality of the attention image \mathbf{F} . A challenging task is to approximate f . One common way to address this problem is to extract a particular type of feature and then design some rules to make decisions. One major difficulty of this common approach is in dealing with different appearances of the same object. It is intractable to manually define the features that can characterize the variety of appearances of objects in the real world.

In this chapter, we use a learning-based approach to approximate the function f . The method assumes no restriction on the type of the object the system can handle. Therefore, it can be applied to a wide variety of objects. The approximation takes three steps:

1. Manually extract the object of interest from each training image. The extracted objects are mapped to attention images using the extractor and scaler.
2. Extract the most expressive features (the principle components) from the training set.
3. Build an interpolation function to approximate f .

In the following subsections, we discuss step 2 and 3 in detail.

7.1.1 Karhunen-Loeve projection

Definition 4 *A vectorizer operator \mathcal{T} transforms an attention image \mathbf{F} of r rows and c columns to a d -dimensional vector \mathbf{V} , such that $\mathbf{V}[i \times c + j] = \mathbf{F}[i, j]$ for any $0 \leq i < r$ and $0 \leq j < c$, where $d = r \times c$.*

Typically d is very large. The Karhunen-Loeve projection (see Chapter 6.1.1 for details) is a very efficient way to reduce a high-dimensional space to a low-dimensional subspace.

Definition 5 *Let V be a $d \times m$ matrix consisting of these m vectors $\{\mathbf{v}_i\}$ as column vectors. Then a Karhunen-Loeve (KL) projection operator \mathcal{P} projects an input vector \mathbf{Y} to \mathbf{Z} , such that $\mathbf{Z} = V^\top(\mathbf{Y} - \mathbf{M}_\mathbf{X})$, where $\mathbf{M}_\mathbf{X}$ is the mean vector of \mathbf{X} .*

We can choose m so that the ratio $\sum_{i=m+1}^n \lambda_i / \sum_{i=1}^n \lambda_i$ is smaller than a given percentage (e.g., 5%).

7.1.2 Approximation as function interpolation

Now we are ready to give the approximation of the verifier $f(\mathbf{F}, \mathbf{P})$. Assuming the intention signal \mathbf{P} is gesture k . Let $L_k = \{l_{k,1}, l_{k,2}, \dots, l_{k,n}\}$ be a training set, where each $l_{k,i}$ is an attention image of a training sample for gesture k . We first use the vectorizer \mathcal{T} to transform each attention image to a vector. Then, we learn the Karhunen-Loeve projection matrix V and the mean vector $\mathbf{M}_\mathbf{X}$. Finally, we use the KL projection operator \mathcal{P} to project each attention image $l_{k,i}$ to a vector $\mathbf{X}_{k,i}$ in the MEF space, such that $\mathbf{X}_{k,i} = \mathcal{P}(\mathcal{T}(l_{k,i}))$ for $i = 1, 2, \dots, n$.

Definition 6 *Given a training vector $\mathbf{X}_{k,i}$ in the MEF space, a Gaussian basis function s_i is*

$$s_i(\mathbf{X}) = e^{-\frac{\|\mathbf{X} - \mathbf{X}_{k,i}\|^2}{\sigma}}, \quad (7.1)$$

where σ is a positive damping factor, and $\|\cdot\|$ denotes the Euclidean distance.

A very small σ tends to reduce the contribution of neighboring training samples. The Gaussian is one of the widely used basis functions [147]. There are other interpolation schemes that may be used here, such as the generalized multiquadratics [169, 193].

Definition 7 *Given a set of n training samples $L_k = \{l_{k,1}, l_{k,2}, \dots, l_{k,n}\}$ of gesture k , the confidence level that the input \mathbf{X} belongs to class k is defined as:*

$$g_k(\mathbf{X}) = \sum_{i=1}^n c_i s_i(\mathbf{X}), \quad (7.2)$$

where the s_i is a Gaussian basis function and the coefficients c_i 's are to be determined by the training samples.

Given n training samples, we have n equations

$$g_k(\mathbf{X}_{k,i}) = \sum_{i=1}^n c_i s_i(\mathbf{X}_{k,i}), \quad (7.3)$$

which are linear with respect to the coefficients c_i 's. If we set $g_k(\mathbf{X}_{k,i})$ equal to 1, we can solve the above equations for c_i using the Gauss-Jordan elimination method [151]. Fig. 7.3 shows how the interpolation function would look in the case when two training samples $(0, 0)$ and $(\sqrt{2}, \sqrt{2})$ are used and $\sigma = 1$.

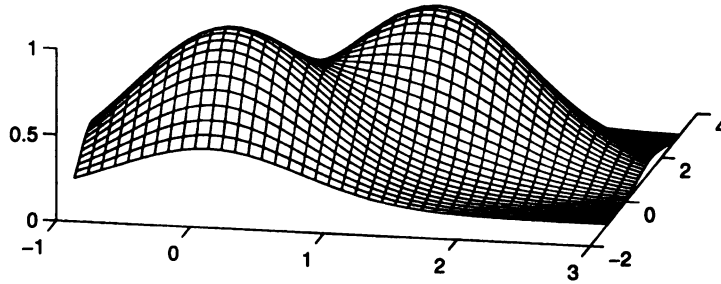


Figure 7.3: Interpolation function with two training samples, $(0,0)$ and $(\sqrt{2}, \sqrt{2})$.

Definition 8 Given a set of training samples $L_k = \{l_{k,1}, l_{k,2}, \dots, l_{k,n}\}$ of gesture k , the corresponding interpolation function g_k , and a confidence level l , a function interpolation scheme approximates the verifier f as follows:

$$\begin{aligned}
f(\mathbf{X}, \mathbf{P}) &= 1 && \text{if } g_k(\mathcal{P}(\mathcal{T}(\mathbf{X}))) > l \\
&= 0 && \text{otherwise}
\end{aligned}$$

7.1.3 Valid segmentation

Based on the definition for the verifier f , we define the concept of *valid segmentation*.

Definition 9 *A segmentation result \mathbf{C} defined on an input image \mathbf{I} and an intention signal \mathbf{P} is valid if $f(\mathcal{P}(\mathcal{T}(\mathcal{M}(\mathcal{E}(\mathbf{I}, \mathbf{C}, \mathbf{P})))), \mathbf{P}) = 1$, where f is the verifier.*

Intuitively, a segmentation result \mathbf{C} is valid if there is a training sample that is sufficiently close to it.

7.2 Predication for Valid Segmentation

This section investigate the first major problem presented in the introduction section, that is, how to find a valid segmentation. Our solution to this problem again resorts to learning.

7.2.1 Overview

Definition 10 *An attention image \mathbf{F} from a fixation of image \mathbf{I}' of m rows and n columns with scale $r \leq 1$ and center position (s, t) , where $0 \leq s < m$ and $0 \leq t < n$, is defined as $\mathbf{F} = \mathcal{M}(\mathbf{B})$, where \mathcal{M} is a scaler and \mathbf{B} is an image with $m' = r \times m$ rows and $n' = r \times n$ columns and*

$$\begin{aligned}
\mathbf{B}[i, j] &= \mathbf{I}'[b_1 + i, b_2 + j] \quad \text{if } 0 \leq b_1 + i < m \text{ and } 0 \leq b_2 + j < n \\
&= 0 \quad \text{otherwise}
\end{aligned}$$

where $b_1 = s - \frac{m'}{2}$, $b_2 = t - \frac{n'}{2}$, $i = 0, 1, \dots, m' - 1$, and $j = 0, 1, \dots, n' - 1$.

The above definition states how to obtain an attention image \mathbf{F} given an image \mathbf{I}' . Each fixation is a partial view of the object in the image \mathbf{I}' . The fixation position is determined by the center position (s, t) and the r is a zooming factor.

Given a training set $L = \{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n\}$, where \mathbf{I}_i is a training image, we first manually get a segmentation mask \mathbf{C} for each \mathbf{I}_i . Then, we apply the extractor to get a subimage \mathbf{I}'_i for each \mathbf{I}_i . Next, we obtain a set of attention images from multiple fixations for each \mathbf{I}'_i . We denote $\mathbf{F}_{i,j}$ be an attention image from j th fixation of sample i in L . The attention images from the training set $L_{\mathbf{F}}$ is

$$L_{\mathbf{F}} = \{\mathbf{F}_{1,1}, \dots, \mathbf{F}_{1,m_1}, \mathbf{F}_{2,1}, \dots, \mathbf{F}_{1,m_2}, \dots, \mathbf{F}_{n,1}, \dots, \mathbf{F}_{n,m_n}\},$$

where m_i is the number of the attention images generated from the training image \mathbf{I}_i . Each attention image from a fixation is associated with the segmentation mask \mathbf{C} , the scale r and the position of the fixation (s, t) . These information is necessary to recover the segmentation for the entire object.

During the segmentation stage, we first use the motion information to select visual attention. Then, we try different fixations on the input image. An attention image from a fixation of an input image is used to query the training set $L_{\mathbf{F}}$. The segmentation mask associated with the query result $\mathbf{F}_{i,j}$ is the predication result. The

predicted segmentation mask is then applied to the input image. Note in order to put the mask back in the right position with the right size, we need to use the scale and the center position of the fixation associated with the input attention image as well as the scale and the center position of the fixation associated with $F_{i,j}$. Finally, we verify the segmentation result to see if the extracted subimage corresponds to a hand gesture that has been learned. If the answer is yes, we find the solution. This solution can further go through a refinement process. Fig. 7.4 gives the outline of the scheme. In the following subsections, we discuss the organization of attention images from the training set.

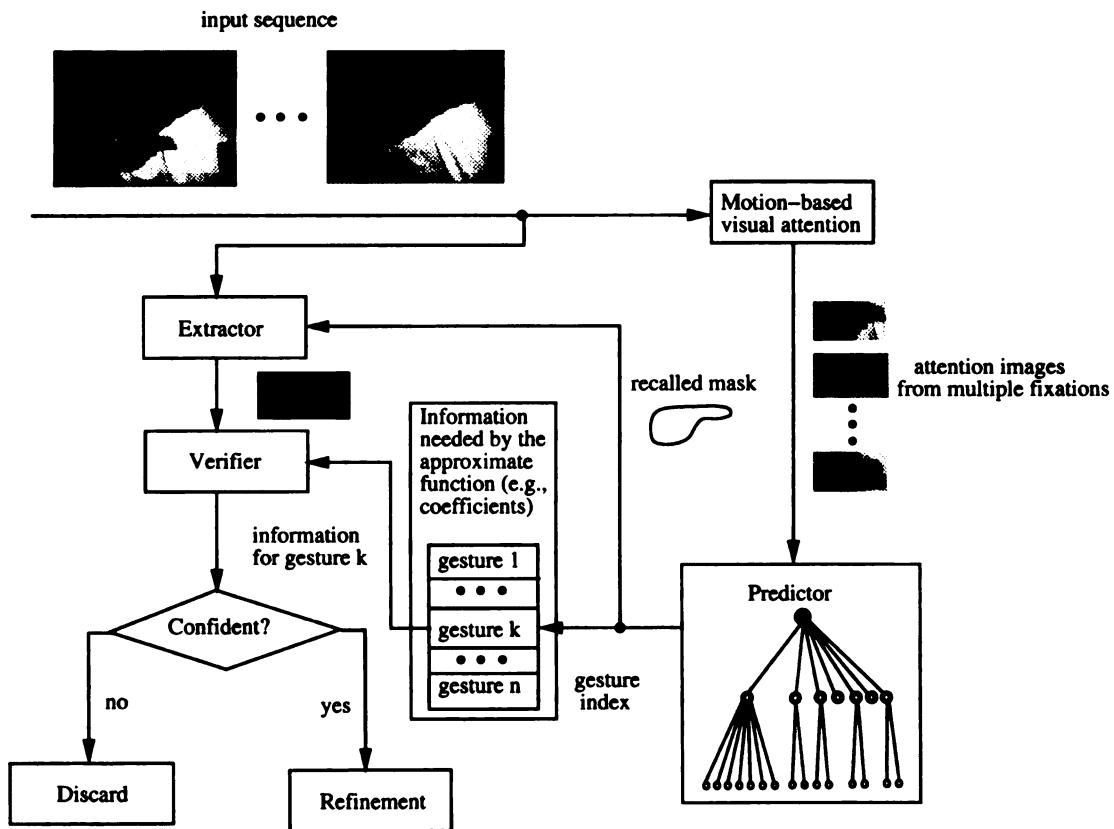


Figure 7.4: Overview of the segmentation scheme.

7.2.2 Organization of attention images from fixations

Our objective is to achieve a retrieval with time complexity $O(\log n)$ for a learning set of size n . With this goal in mind, we build a hierarchical structure to organize the data.

Similar to the training in approximating the verifier, we first use the vectorizer operator \mathcal{T} to transform each attention image in the training set L to a vector. Next, we obtain the Karhunen-Loeve projection matrix V and the mean vector \mathbf{M}_X . Finally, we apply the KL projection operator \mathcal{P} to project the training sample to the MEF space. These vectors in the MEF space are organized using a hierarchical quasi-Voronoi diagram structure as we explain below.

Definition 11 *Given a set of points $V = \{V_1, V_2, \dots, V_n\}$ in the space S , the Voronoi diagram partitions S into $R = \{R_1, R_2, \dots, R_n\}$ regions, where $R_i \cap R_j = \emptyset$ when $i \neq j$, $\bigcup_{i=1}^n R_i = S$, and for any $x \in S$, $x \in R_i$ if and only if $\|x - V_i\| < \|x - V_j\|$ for any $j \neq i$. We denote V_i be the center of the region R_i .*

Definition 12 *A hierarchical quasi-Voronoi diagram P of S is a set of partitions $P = \{P_1, P_2, \dots, P_m\}$, where every $P_i = \{P_{i,1}, \dots, P_{i,n_i}\}$, $i = 1, 2, \dots, m$ is a partition of S . $P_{i+1} = \{P_{i+1,1}, \dots, P_{i+1,n_{i+1}}\}$ is a finer Voronoi diagram partition of P_i in the sense that corresponding to every element $P_{i,k} \in P_i$, P_{i+1} contains a Voronoi partition $\{P_{i+1,s}, \dots, P_{i+1,t}\}$ of $P_{i,k}$.*

The graphic description in Fig. 7.5 gives an simplified but intuitive explanation of the hierarchical quasi-Voronoi diagram. The structure is a tree. The root corresponds

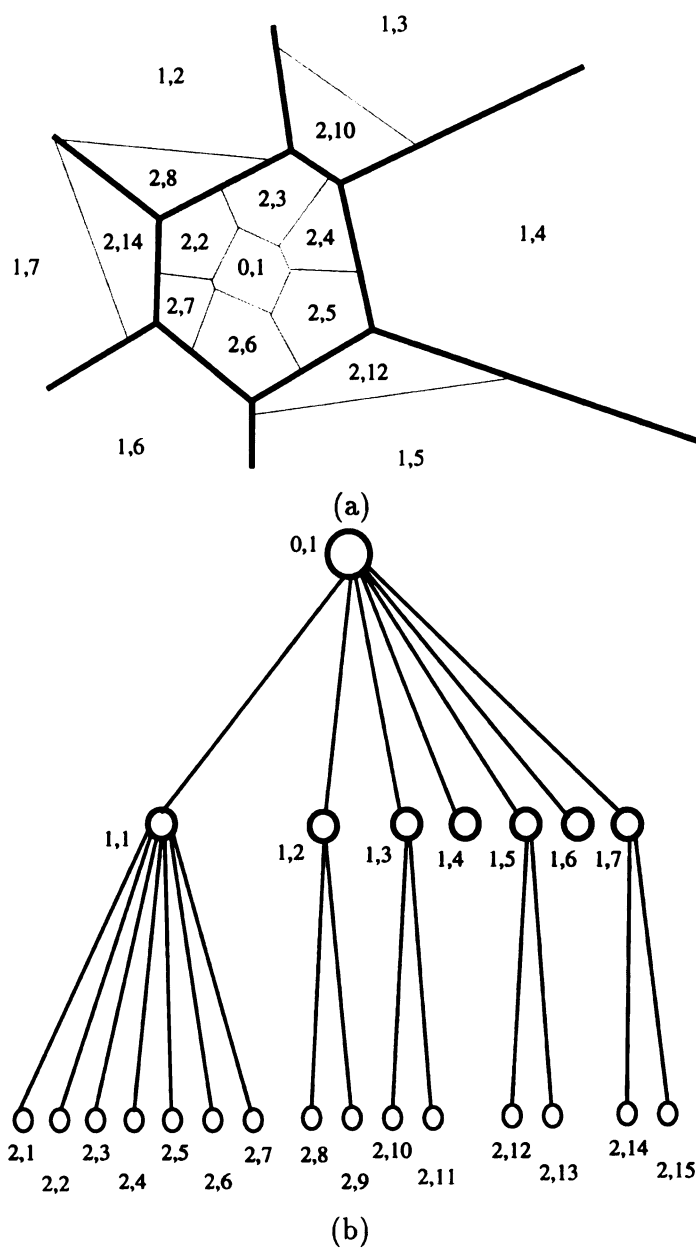


Figure 7.5: A 2-D illustration of a hierarchical quasi-Voronoi diagram and the corresponding recursive partition tree. (a) In partition, the label indicates the center of a cell. The label of the child to which its parent's center belongs is not shown due to the lack of space. (b) The corresponding recursive partition tree.

to the entire space of all the possible inputs. The children of the root partition the space into large cells, as shown by thick lines in Fig. 7.5. The children of a parent subdivide the parent's cell further into smaller cells, and so on.

7.2.3 Prediction as querying the training set

Given a training set L , a hierarchical quasi-Voronoi diagram $P = \{P_1, P_2, \dots, P_n\}$ corresponding to L and a query sample \mathbf{X} , the prediction problem is to find a training sample $\mathbf{X}' \in L$, such that $\|\mathbf{X} - \mathbf{X}'\| \leq \|\mathbf{X} - \mathbf{X}''\|$ for any $\mathbf{X}'' \in L$ with $\mathbf{X}'' \neq \mathbf{X}'$.

The type of query mentioned above is a nearest neighbor problem, also known as *post-office* problem [104]. The nearest neighbor problem has been studied extensively in the past. There are efficient query algorithms $O(\log n)$ for two- or three-dimensional cases [37, 55]. However, there still lacks of efficient solutions for the case with dimension higher than three. k -d tree based nearest neighbor algorithms have been widely used in computer vision [15, 205]. k -d trees are extremely versatile and efficient to use in low dimensions. However, the performance degrades exponentially in high dimensions. R-tree and its variants [82, 162, 14] have similar performance of nearest neighbor searches in high dimensions. In this section, we will present an efficient algorithm when the training set is d -supportive as defined below.

Definition 13 *Let S be a set which contains all possible samples. A training set $L = \{L_1, L_2, \dots, L_n\}$ is a d -supportive training set if for any test sample $\mathbf{X} \in S$, there exist i such that $\|\mathbf{X} - L_i\| < d$, where $\|\cdot\|$ is the Euclidean distance.*

Next, we will show that a training set can become a d -supportive set as the size of

the training set increases if any point $\mathbf{X}_0 \in S$ positively supported as defined below.

Definition 14 *Let S be a set which contains all possible samples. We consider \mathbf{X} in a learning set L as a random sample from S . A point $\mathbf{X}_0 \in S$ is positively supported if for any $\delta > 0$ we have $P\{\|\mathbf{X} - \mathbf{X}_0\| \leq \delta\} > 0$, where $P\{e\}$ denotes the probability of the event e .*

If S consists of a finite number of discrete points, a point \mathbf{X} in P is positively supported means that the probability of selecting \mathbf{X} as a sample is not a zero-probability event. If S consists of infinitely many points, a point \mathbf{X} in P is positively supported means that in any small neighborhood centered at \mathbf{X} , the probability of selecting any point in the neighborhood is not a zero-probability event.

Theorem 4 *Suppose \mathbf{X}_0 is a positively supported point in S . Given any small $\epsilon > 0$, there is a positive number $k_0 > 0$, such that as long as we independently draw $k > k_0$ learning set $L = \{L_1, L_2, \dots, L_k\}$, the probability that \mathbf{X}_0 is d -supported by L has the following property*

$$P\{\|\mathbf{X}_0 - L_i\| < d \mid L_i \in L\} > 1 - \epsilon.$$

Proof of Theorem 4. \mathbf{X}_0 is positively supported, we have $P\{\|\mathbf{X}_0 - \mathbf{X}\| < d\} > 0 = p$. If we independently draw k learning samples L , the probability $P\{\|\mathbf{X}_0 - \mathbf{X}_l\| \geq d \mid \forall \mathbf{X}_l \in L\} = (1 - p)^k$. Thus, $\lim_{k \rightarrow \infty} P = 0$. \square

The theorem says that as training size increases, the training set *pointwisely* converges to a d -supportive set. Next two theorems show the fact that if the training set is d -supportive, we have an efficient query algorithm.

Theorem 5 We have a set of d -supportive training set $L = \{L_1, L_2, \dots, L_n\}$, a hierarchical quasi-Voronoi diagram $P = \{P_1, P_2, \dots, P_n\}$ corresponding to L and a query sample $\mathbf{X} \in S$. Let the i th partition be $P_i = \{P_{i,1}, P_{i,2}, \dots, P_{i,n_i}\}$ and $C = \{C_1, C_2, \dots, C_n\}$ be the corresponding centers of regions in P_i . Assume C_1 be the center to \mathbf{X} such that $\|C_1 - \mathbf{X}\| \leq \|C_i - \mathbf{X}\|$ for any $i \neq 1$. Let C_2 be any other center and P_1 be a boundary hyperplane between regions represented by C_1 and C_2 as illustrated in Fig. 7.6. Then the region of C_2 does not contain the nearest training sample to \mathbf{X} if the distance between \mathbf{X} and the hyperplane P_1 is greater than d .

Proof of Theorem 5. If the distance from \mathbf{X} to P_1 is greater than d , then according to the definition of the d , there is going to a training sample \mathbf{X}_1 in C_1 such that $\|\mathbf{X} - \mathbf{X}_1\| < d$. Since for any samples \mathbf{Y} in region C_2 , $\|\mathbf{X} - \mathbf{Y}\| \geq d$, so the closest training sample will not be in the region of C_2 . \square

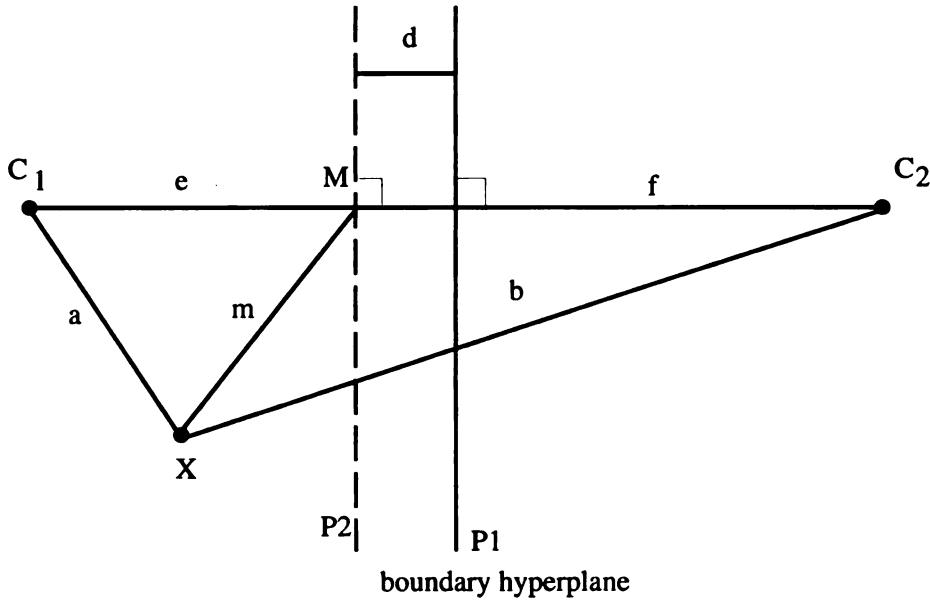


Figure 7.6: A 2D illustration of nearest neighbor query theorems.

In order to avoid to calculate the point to hyperplane distance in a high dimen-

sional space, we can use following equivalent theorem.

Theorem 6 *Let $\|C_1 - C_2\| = r$, $f = \frac{r}{2}$, $e = \frac{r}{2} - d$, $\|C_1 - \mathbf{X}\| = a$ and $\|C_2 - \mathbf{X}\| = b$ as shown in Fig. 7.6. The region of C_2 does not contain the nearest training sample to \mathbf{X} if $a^2 - e^2 < b^2 - f^2$.*

Proof of Theorem 6. Let P_2 be the hyperplane such that the distance between P_1 and P_2 is d as shown in Fig. 7.6. Notice according to the definition of Voronoi diagram, f is the distance from C_2 to hyperplane P_1 and e is the distance from C_1 to hyperplane P_2 . Let M be the intersection point between line C_1C_2 and P_2 , $\|\mathbf{X} - M\| = m$, and the angle $\angle C_1MS = \alpha$. Then, we have

$$a^2 - e^2 = m^2 - 2em \cos \alpha$$

$$b^2 - f^2 = m^2 - 2fm \cos(180^\circ - \alpha).$$

As we have shown in the Theorem 5, the C_2 region does not contain the nearest neighbor if \mathbf{X} is not in the region between hyperplane P_1 and P_2 . This means $\alpha < 90^\circ$. Thus, we have $a^2 - e^2 < m^2$ due to $\cos \alpha > 0$ and $b^2 - f^2 > m^2$ due to $\cos(180^\circ - \alpha) < 0$. Therefore, we have $a^2 - e^2 < b^2 - f^2$. \square

Now we are ready to present our query algorithm. Theorem 7 shows the time complexity of the query algorithm.

```

Given query sample  $\mathbf{X}$ , a hierarchical quasi-Voronoi diagram  $P$  corresponding to
a  $d$ -supportive learning set with  $C_{i,j}$  denotes to the center of the partition region  $P_{i,j}$ .
begin
   $nodes\_list = root$ ;
  while  $nodes\_list \neq \text{nil}$  do
    Pop the first node  $nd$  from  $nodes\_list$ ;
    if  $nd$  is leaf then
      add center  $C_{nd}$  to the  $center\_list$ ;
    else
      for regions under the node  $nd$  do
        Add region which is closest  $\mathbf{X}$  to  $nodes\_list$ ;
        Add regions which satisfy Theorem 3 to  $nodes\_list$ ;
      end for
    end if
  end while
  Output the center in the  $center\_list$  that minimizes  $\|\mathbf{X} - C\|$ .
end begin

```

The next theorem proves that the average time complexity of the query algorithm is $O(\log n)$ where n is the size of the training set.

Theorem 7 *Assume that n training samples in L are independently drawn and the training set L is d -supportive. The hierarchical quasi-Voronoi diagram P is constructed based on the training such that each $P_{i,j}$ has no more than p branches and the volume of region $P_{i,j}$ reduces by a constant factor as the i increases, $V_{i,j} = fV_{i+1,j}$ and $f > 1$. Assume $P_{i,j}$ be the region C_1 as shown in Fig. 7.6. We select d such that for each region $P_{i,j}$ in P , $V'_{i,j} \subset V_{i,j}$ where $V_{i,j}$ is the volume of the region $P_{i,j}$ and $V'_{i,j}$ is the volume of the region between the hyperplane P_1 and the hyperplane P_2 in Fig. 7.6. Then, the above algorithm has the average time complexity $O(\log n)$ in terms of*

finding the nearest neighbor.

Proof of Theorem 7. Assume at the stage i , the query example \mathbf{X} is in the region $P_{i,j}$. Let $P_{i,j}$ be the C_1 region as shown in Fig. 7.6. Let $V_{i,j}$ be the volume of the region $P_{i,j}$ and $V'_{i,j} \subset V_{i,j}$ be the volume of the region (between P_1 and P_2 in Fig. 7.6) where the multiple search is needed. Thus, the average time complexity

$$T(i) \leq \left(p \frac{V'_{i,j}}{V_S} + \frac{V_S - V'_{i,j}}{V_S}\right) T(i+1) + 1, \quad (7.4)$$

where V_S is the volume of the entire space. Equation (7.4) means that if $\mathbf{X} \in V'_{i,j}$, then may need to search maximum p branches, otherwise one branch is needed. We expand the equation (7.4)

$$T(i) \leq T(i+m) \prod_{k=1}^m \left(1 + \frac{(p-1)V'_{i+k,j}}{V_S}\right) + \sum_{k=1}^m \left(1 + \frac{(p-1)V'_{i+k,j}}{V_S}\right) + 1$$

Since $V_{i,j} = fV_{i+1,j}$, we have $V'_{i,j} < V_{i,j} = f^k V_{i+k,j}$. Then

$$\frac{(p-1)V'_{i+k,j}}{V_S} < \frac{(p-1)V_{i,j}}{f^k V_S}$$

For $f > 1$, it is not difficult to show that

$$\prod_{k=1}^m \left(1 + \frac{(p-1)V_{i,j}}{f^k V_S}\right) = O(1)$$

and

$$\sum_{k=1}^m \frac{(p-1)V_{i,j}}{f^k V_S} = O(1)$$

Thus, we have

$$T(i) = O(m)$$

Let $i = 0$ be the root and m be the height of the tree, we have $m = O(\log n)$.

Therefore, $T(0) = O(\log n)$. \square

The query algorithm finds the nearest training sample from L using a tree structure. If the training set is d -supportive, the nearest neighbor is guaranteed to be found. However, in order to achieve fast retrieval, as shown in Theorem 7, d should be selected such that for each region $P_{i,j}$ in P , $V'_{i,j} \subset V_{i,j}$ where $V_{i,j}$ is the volume of the region $P_{i,j}$ and $V'_{i,j}$ is the volume of the region between the hyperplane P_1 and the hyperplane P_2 in Fig. 7.6. This means that d can not be too large. If d is small, the assumption that the training set is d -supportive may be too strong. Another problem is that there is no way to verify whether the training set is d -supportive or not because the true distribution of samples is unknown. The next theorem shows what kind of the solution the query will obtain if the training set is not d -supportive.

Definition 15 *Let $P_{i,j}$ be a region in the hierarchical quasi-Voronoi diagram P and C be the center of the region. A minimum ball $B(C, r)$ of the region $P_{i,j}$ is defined*

$$r = \inf_{r \in R}, \tag{7.5}$$

where C is the center of the ball, r is the radius of the ball, and R is a set such that

for any $r' \in R$, we have $P_{i,j} \subset B(C, r')$.

The minimum ball $B(C, r)$ is centered at C and has the smallest radius for all balls which contain the region $P_{i,j}$.

Theorem 8 *Let $P = \{P_1, P_2, \dots, P_m\}$ be the hierarchical quasi-Voronoi diagram based on the training set L . Let the $\mathbf{X}_c \in P_{m,k}$ be the solution found by the query algorithm and \mathbf{X}_N be the nearest neighbor of the query \mathbf{X} in L . Then, we have*

$$\|\mathbf{X} - \mathbf{X}_c\| \leq \|\mathbf{X} - \mathbf{X}_N\| \times \frac{r}{d}, \quad (7.6)$$

where r is the radius of the minimum ball of the region $P_{m,k}$ and d is our assumption of the training set.

Proof of Theorem 8. First, since P_m is the finest partition and according to the definition of the hierarchical quasi-Voronoi diagram, there exists a $P_{m,k}$ such that $\mathbf{X}_c \in P_{m,k}$. Next, since \mathbf{X}_c is the solution, we have \mathbf{X} in the minimum ball of the region $P_{m,k}$. So, we have $\|\mathbf{X} - \mathbf{X}_c\| \leq r$. Finally, we have $\|\mathbf{X} - \mathbf{X}_N\| > d$ because otherwise \mathbf{X}_c will not be the only choice according to Theorem 5. Thus, we have

$$\|\mathbf{X} - \mathbf{X}_c\| \leq \|\mathbf{X} - \mathbf{X}_N\| \times \frac{r}{d}. \quad (7.7)$$

□

Theorem 8 gives us some insights to select d . There is a trade off between speed and accuracy. In practice, given a training set $L = \{l_1, l_2, \dots, l_n\}$, we choose the d as

follows:

$$d = \frac{0.2 \sum_{i=1}^n \|l_i - l'_i\|}{n - 1}, \quad (7.8)$$

where l'_i is the nearest neighbor of l_i in L . Since the region $P_{m,k}$ in Theorem 8 only has one training sample, \mathbf{X}_c , we can expect that r is comparable to d . This means that a reasonable neighbor is found although it is not guaranteed to be the nearest one.

7.3 Experiments

We have applied our segmentation scheme to the task of hand segmentation in the experiments.

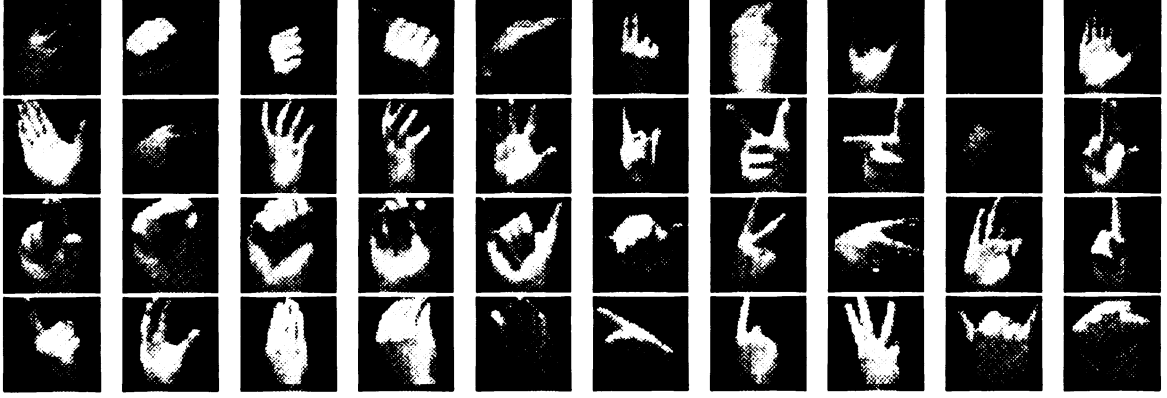


Figure 7.7: A representative subset of hand shapes used in the experiment.

7.3.1 Training

Two types of training were conducted in the experiments. The first type of training is to get the approximation for verifier f which would be used later to check the validation of the segmentation. For each gesture, a number between (27 and 36)

of training samples were used to obtain the approximation of the verifier f for that gesture. Given a set of training samples $L = \{l_1, l_2, \dots, l_n\}$ for gesture k , we empirically determined the damping factor σ in the interpolation function as follows:

$$\sigma = \frac{0.2 \sum_{i=1}^{n-1} \|\mathbf{X}_i - \mathbf{X}_{i+1}\|}{n-1}, \quad (7.9)$$

where $\mathbf{X}_i = \mathcal{P}(\mathcal{T}(l_i))$, and \mathcal{P} and \mathcal{T} are the vectorizer operator and the projection operator respectively.

The second type of training was to generate the attention images from multiple fixations of training samples. In the current implementation, the selection of the fixations is mechanical. Totally 19 fixations were used for each training sample. The scales s and positions (s, t) of these 19 fixations for an image with m rows and n columns are listed in Table 7.1. Fig. 7.8 shows the attention images of the 19 fixations from one training sample. The attention images with more than 30% background pixels presented in the attention window would be discarded. The total number of remaining attention images used in the experiment is 1742.

Table 7.1: The list of fixation scale and position

Scale	P1	P2	P3	P4	P5	P6	P7	P8	P9
1.0	$(\frac{m}{2}, \frac{n}{2})$								
0.75	$(\frac{m}{2}, \frac{n}{2})$								
0.5	$(\frac{m}{2}, \frac{n}{2})$	$(\frac{m}{3}, \frac{n}{3})$	$(\frac{m}{3}, \frac{n}{2})$	$(\frac{m}{3}, \frac{2n}{3})$	$(\frac{m}{2}, \frac{n}{3})$	$(\frac{m}{2}, \frac{2n}{3})$	$(\frac{2m}{3}, \frac{n}{3})$	$(\frac{2m}{3}, \frac{n}{2})$	$(\frac{2m}{3}, \frac{2n}{3})$
0.33		$(\frac{m}{3}, \frac{n}{3})$	$(\frac{m}{3}, \frac{n}{2})$	$(\frac{m}{3}, \frac{2n}{3})$	$(\frac{m}{2}, \frac{n}{3})$	$(\frac{m}{2}, \frac{2n}{3})$	$(\frac{2m}{3}, \frac{n}{3})$	$(\frac{2m}{3}, \frac{n}{2})$	$(\frac{2m}{3}, \frac{2n}{3})$

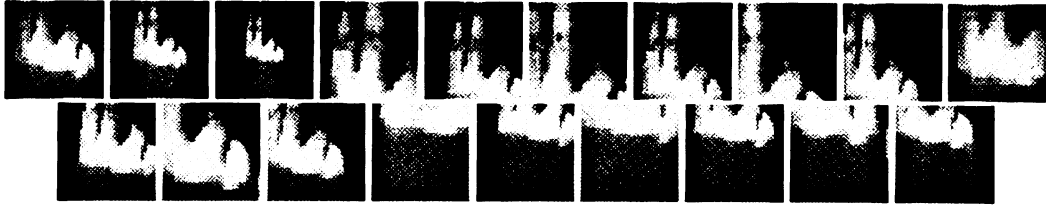


Figure 7.8: The attention images from 19 mechanical fixations of a training sample.

7.3.2 Hand segmentation

The trained system was tested to perform the segmentation task from a temporal sequence of intensity images. Each sequence represents a complete hand sign. Fig. 7.9 shows eight sample sequences.

Motion-based visual attention

In order to speed up the process of the segmentation, we utilize motion information to find a motion attention window. The algorithm to find the motion attention window is outlined as follows. Fig. 7.10 shows results of motion-based visual attention.

Given an image \mathbf{I} and a neighboring image \mathbf{I}' in the sequence.

begin

1. Get the difference image \mathbf{D} such that

$$\mathbf{D}[i, j] = \|\mathbf{I}[i, j] - \mathbf{I}'[i, j]\|$$

2. Thresholding \mathbf{D} .

3. Find the smallest rectangular window containing the largest connected component in \mathbf{D} .

end begin

Segmentation

The task of segmentation would be a lot easier if the motion attention window generated by the attention algorithm was centered at the right position and included



Figure 7.9: Eight sample sequences. From top to bottom, they represent the signs “happy”, “hot”, “nothing”, “parent”, “pepper”, “smart”, “welcome”, and “yes”.



Figure 7.10: Results of motion-based attention are shown using dark rectangular windows.

the entire hand. Unfortunately this is not always true. The attention algorithm can detect the rough position of a moving object, but the accuracy is not guaranteed. In Fig. 7.10, we show some of the results of motion-based attention. The dark rectangular is the smallest rectangular which contains the largest connected component of the image difference. The positions of these rectangles only give us the rough positions of the hand and they can deviate from the desired positions. We solve this problem by doing some limited search based on the motion attention window. In the current implementation, given a motion attention window with m rows and n columns, we try the candidates with size from $(0.5m, 0.5n)$ to $(2m, 2n)$ using step size $(0.5m, 0.5n)$. The search stops if a valid segmentation is found. There is a trade off between training and testing. The more fixations we use in the training, the less search we need in the testing.

We tested the system with 161 sequences (each has 5 images) which were not used in the training. A result was rejected if the system could not find a valid segmentation with a confidence level l . The segmentation was considered as a correct one if the correct gesture segmentation C was retrieved and placed in the right position of the

test image. For the case of $l = 0.2$, we have achieved 95% correct segmentation rate with 3% false rejection rate. Fig. 7.11 shows some segmentation results. The average computational cost for each image was 58.3 seconds on a SGI INDIGO 2 workstation. The experimental results are summarized in Table 7.2.

Table 7.2: Summary of the experimental data

<i>Training</i>			
samples per verifier		samples for predictor	
27~36		1742	
<i>Testing with confidence level=0.2</i>			
number of images	false rejection	correct segmentation	CPU time per image
805	3%	95%	58.3 sec.

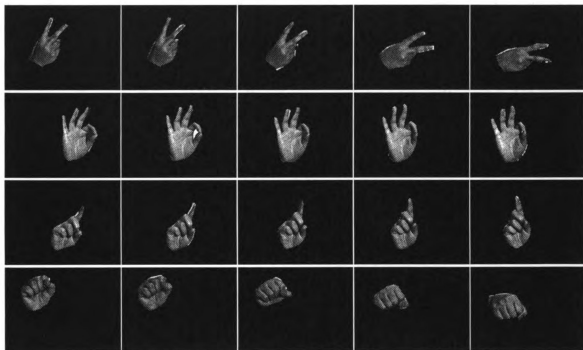


Figure 7.11: The results of the segmentation are shown after masking off the background.

7.4 Conclusions and Future Work

A segmentation scheme using attention images from multiple fixations is presented in this chapter. The major advantage of this scheme is that it can handle a large number of different deformable objects presented in various complex backgrounds. The scheme is also relatively efficient since the search of the segmentation is guided by the past knowledge through a predication-and-verification scheme.

In the current implementation, the fixations are generated mechanically. The number of fixations and the positions of fixations are the same regardless of the types of gestures. This is not very efficient. Some gestures may be very simple so that a few fixations are enough to recognize them. In order to achieve the optimal performance, different gestures require different positions of fixations. In the future, we plan to investigate the generation of the fixations based on learning. The previous fixations are used to guide the next action. The next action could be (a) termination of the process of generating fixation if the gesture has already been recognized; or (b) finding the appropriate position for next fixation.

Chapter 8

View-Based Hand Sign

Recognition from Intensity Image

Sequences

After segmentation, the hand in each image of a sequence is mapped to a fovea image of a standard fixed size. Segmented fovea images at different times form a standard *spatiotemporal fovea sequence*, in which both temporal and spatial dimensions are normalized. The global motion information of the object of interest is placed in a global motion vector, which records the size and position information of the segmented object in the original image. This vector is necessary because once the object is segmented and mapped to a fovea sequence with a standard spatiotemporal size, the global motion information is lost.

Let a fovea image f of m rows and n columns be an (mn) -dimensional vector. For

example, the set of image pixels $\{f(i, j) \mid 0 \leq i < m, 0 \leq j < n\}$ can be written as a vector $\mathbf{V} = (v_1, v_2, \dots, v_d)$ where $v_{mi+j} = f(i, j)$ and $d = mn$. Note that although pixels in an image are lined up to form a 1-D vector \mathbf{V} this way, 2-D neighborhood information between pixels will be characterized by the scatter matrix of \mathbf{V} to be discussed later. Let p be the standard temporal length and f_i be the hand fovea image corresponding to the frame i . Then we create a new vector \mathbf{X} , called the *fovea vector*, which is a concatenation of the hand foveas and global motion vector G ,

$$\mathbf{X} = (f_1, f_2, \dots, f_p, G). \quad (8.1)$$

In this chapter, we present a learning-based method to recognize hand signs from the fovea vector.

An automatic hand gesture recognition system accepts an input fovea vector \mathbf{X} and outputs the recognition result \mathbf{C} which classifies the \mathbf{X} into one of the gestures. Thus, a recognition system can be denoted by a function f that maps elements in the space of \mathbf{X} to elements in the space of \mathbf{C} . Our objective of constructing a recognition system is equivalent to approximating function $f : S \mapsto C$ by another function $\hat{f} : S \mapsto C$. The error of a approximation can be indicated by certain measure of the error $\hat{f} - f$. One such measure is the mean square error:

$$E(\hat{f} - f) = \int_{\mathbf{X} \in S} (\hat{f}(\mathbf{X}) - f(\mathbf{X}))^2 dF(\mathbf{X})$$

where $F(\mathbf{X})$ is the probability distribution function \mathbf{X} in S . In other words, \hat{f} can

defer a lot from f in parts where \mathbf{X} never occurs, without affecting the error measure. Another measure is the pointwise absolute error $\|\hat{f}(\mathbf{X}) - f(\mathbf{X})\|$ for any point \mathbf{X} in S' , where $S' \subset S$ is a subset of S that is of interest to a certain problem.

Of course, f is typically high-dimensional and highly complex. A powerful method of constructing \hat{f} is using learning. Specifically, a series of cases is acquired as the learning data set:

$$L = \{(\mathbf{X}_i, f(\mathbf{X}_i)) | i = 1, 2, \dots, n\}.$$

Then, construct \hat{f} based on L . For notational convenience, the sample points in L is denoted by $X(L)$:

$$X(L) = \{\mathbf{X}_i | i = 1, 2, \dots, n\}. \quad (8.2)$$

$X(L)$ should be drawn from the real situation so that the underlying distribution of $X(L)$ is as close to the real distribution as possible.

In this chapter, we compare two different approximators of the function f . The first approximator uses the nearest neighbor decision rule in the MEF space. In the second approach, we use a recursive partition tree to approximate the function f in the MDF space. The definitions of MEF and MDF will be given at the following sections.

8.1 Nearest Neighbor Approximator in the MEF Space

Typically an image space is very large. The Karhunen-Loeve projection (see Chapter 6.1.1 for details) is a very efficient way to reduce a high-dimensional space a much lower dimensional space. The base vector of this low dimensional space is called the *most expressive features* (MEF) in that they best describe the sample population in the sense of linear transform.

Given a training set of fovea vectors $L = \{F_1, F_2, \dots, F_n\}$, we first obtain the Karhunen-Loeve projection matrix V and the mean vector \mathbf{M}_F . Then, we project each training sample F_i to a vector \mathbf{X}_i in the MEF space, where $\mathbf{X}_i = V^T(F_i - \mathbf{M}_F)$. Similarly, we also project any query sample onto the above MEF subspace.

Definition 16 *Given a learning set $L = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$ in its MEF space and its corresponding labels $M = \{l_1, l_2, \dots, l_n\}$, a nearest-neighbor (NN) approximator \hat{f} of f associated with L is defined as follows. For any query sample \mathbf{X} , $\hat{f}(\mathbf{X}) = l_i$, where \mathbf{X}_i is the nearest neighbor of \mathbf{X} in L .*

An NN approximator is a piecewise constant function, constant in every P_i , where P_i is a region of the Voronoi diagram based on the training set L . In Chapter 7.2.3, we presented an efficient nearest neighbor query algorithm which uses the hierarchical quasi-Voronoi diagram.

8.2 Approximation Using Recursive Partition Tree in the MDF Space

The MEF's are, in general, not the best ones for classification, because the features that describe some major variations in the class are typically irrelevant to how the subclasses are divided as illustrated in Fig. 8.1.

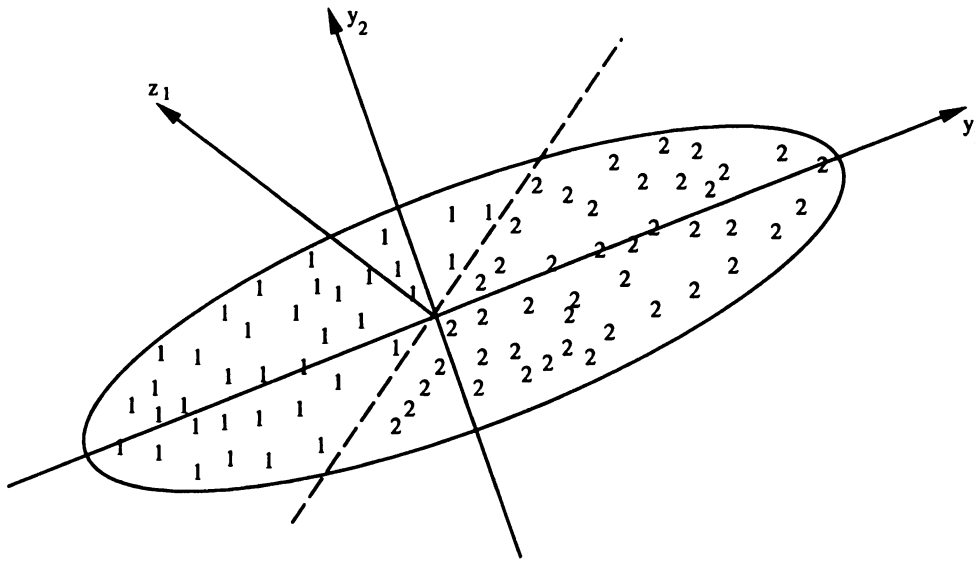


Figure 8.1: A 2D illustration of the most discriminating features (MDF). The MDF is projection along z_1 . The MEF along y_1 can not separate the two subclasses.

8.2.1 The Most Discriminating Features (MDF)

In this chapter, multiclass, multivariate discriminant analysis [197] is used to select the MDF's. It is a generalization of Fisher's linear discriminant [57]. Suppose samples of \mathbf{Y} are m -dimensional random vectors from c classes. The i th class has a probability p_i , a mean vector \mathbf{m}_i and a scatter matrix Σ_i . The *within-class scatter matrix* is

defined by

$$S_w = \sum_{i=1}^c p_i E\{(\mathbf{Y} - \mathbf{m}_i)(\mathbf{Y} - \mathbf{m}_i)^t | \omega_i\} = \sum_{i=1}^c p_i \Sigma_i. \quad (8.3)$$

The *between-class scatter matrix* is

$$S_b = \sum_{i=1}^c p_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t, \quad (8.4)$$

where the grand mean \mathbf{m} is defined as $\mathbf{m} = E\mathbf{Y} = \sum_{i=1}^c p_i \mathbf{m}_i$. The *mixture scatter matrix* is the covariance matrix of all the samples regardless of their class assignments:

$$S_m = E\{(\mathbf{Y} - \mathbf{m})(\mathbf{Y} - \mathbf{m})^t\} = S_w + S_b. \quad (8.5)$$

Suppose we use k -dimensional linear features $\mathbf{Z} = W^t \mathbf{Y}$ where W is an $m \times k$ rectangular matrix whose column vectors are linearly independent. The above mapping represents a linear projection from m -dimensional space to k -dimensional space. The samples $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_n$ project to a corresponding set of samples $\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n$ whose within-class scatter, and between-class scatter matrices are S_{Z_w} and S_{Z_b} , respectively. It is straightforward matter to show that

$$S_{Z_w} = W^t S_w W \quad (8.6)$$

$$S_{Z_b} = W^t S_b W. \quad (8.7)$$

These equations show how the within-class and between-class scatter matrices are transformed by the projection to the lower dimensional space. What we seek is a

transformation matrix W that maximizes in some sense the ratio of the between-class scatter to the within-class scatter. A simple scalar measure of scatter is the determinant of the scatter matrix. The determinant is the product of the eigenvalues, and hence is the product of the “variances” in the principal directions, thereby measuring the square of the hyperellipsoidal scattering volume. Using this measure, we obtain the criterion function to maximize

$$J(W) = \frac{|W^t S_b W|}{|W^t S_w W|}. \quad (8.8)$$

It can be proved [197] that the optimal W that maximizes the above function are the generalized eigenvectors that correspond to the largest eigenvalues in

$$S_b \mathbf{w}_i = \lambda_i S_w \mathbf{w}_i. \quad (8.9)$$

In order to avoid compute the inverse of S_w , we can find the eigenvalues as the roots of the characteristic polynomial

$$|S_b - \lambda_i S_w| = 0 \quad (8.10)$$

and then solve

$$(S_b - \lambda_i S_w) \mathbf{w}_i = 0 \quad (8.11)$$

directly for the eigenvectors \mathbf{w}_i . Since the rank of S_b is at most $c - 1$, we know that only at most $c - 1$ features $\{\mathbf{w}_i\}$ are needed and we call these features the most

discriminating features (MDFs).

8.2.2 Curse of dimensionality and the DKL projection

The discriminant analysis procedure breaks down when the within-class scatter matrix S_w becomes degenerate, which is our case due to a high dimension of the input image and a much smaller number of training samples. Weng [193] proposed *DKL projection* (short for Discriminant Karhunen-Loeve projection). In the DKL projection, the discriminant analysis is based on the space of Karhunen-Loeve projection (MEF space), where the degeneracy typically does not occur.

8.2.3 Recursive partition tree

For large number of classes, the overall feature set may not be best for specific pairs of classes. An alternative classification scheme is the hierarchical classifier, where the most obvious discriminations are done first, postponing the more subtle distinctions to a later stage [135, 163]. In this chapter, we present a recursive partition tree approximator in the MDF space.

Definition 17 *Given a training set of fovea vectors $L = \{F_1, F_2, \dots, F_n\}$, a recursive partition tree participates the space S as follows. Given a partition $P_i = \{P_{i,1}, P_{i,2}, \dots, P_{i,n_i}\}$ of S at level i , the partition at level $i + 1$ $P_{i+1} = \{P_{i+1,1}, P_{i+1,2}, \dots, P_{i+1,n_{i+1}}\}$ is a finer partition of P_i such that for each $P_{i,j} \in P_i$, $P_{i,j}$ contains either $P_{i,j}$ itself or $P_{i+1,k}, P_{i+1,k+1}, \dots, P_{i+1,k+b}$ so that $P_{i,j} = \bigcup_{m=1}^b P_{i+1,k+m}$.*

A hierarchical Voronoi diagram is hierarchical partition that satisfies an additional condition: Every cell is further partitioned at a deeper level only by a Voronoi diagram. The Voronoi diagram in each cell is determined by a few training samples within the cell. In the current implementation, we use an adaptive radius r to split the cell. Once r is determined, we have k ($k > 1$) training samples and the distance between each pair is greater than r . These k samples are the centers to generate a Voronoi diagram and then we move on to the next level. The graphic description in Fig. 8.2 gives an simplified but intuitive explanation of the hierarchical Voronoi diagram. In Fig. 8.2, the partition under the root is created by three samples indicated by circles. Two additional samples indicated by rectangulars are used to get the partition of the next level. The partition of a region ends when the samples under the region are from the same class.

Due to the complexity of the problem, the overall MDF feature set may not be best for specific pairs of classes. In our implementation, the MDF's are computed locally. For each subregion $P_{i,j}$, we obtain DKL projection matrices $V_{i,j}$ and $W_{i,j}$ and mean vector $\mathbf{M}_{i,j}$ based on the training samples within $P_{i,j}$, where $V_{i,j}$ is the projection matrix to the MEF space and $W_{i,j}$ is the projection matrix to the MDF space as defined previously. The leaves of the partition tree correspond to the regions which contain the training samples from a single class. The approximator uses the following decision rule to classify the query fovea vector \mathbf{X} to the class of a leaf cell.

Definition 18 *Given a training set of fovea vectors $L = \{F_1, F_2, \dots, F_n\}$ and corresponding recursive partition tree, for any query fovea vector \mathbf{X} , if the current level*

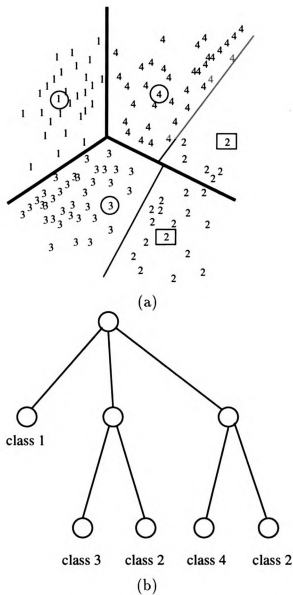


Figure 8.2: A 2-D illustration of a hierarchical Voronoi diagram and the corresponding recursive partition tree. (a) The partition, where the circles and rectangles indicate the samples used to create the Voronoi diagram. (b) The corresponding recursive partition tree.

is not a leaf, the recursive partition tree approximator (RPTA) selects the cell with center C_i if for any other cell with center C_j , we have $R_d(\mathbf{X}, C_i) < R_d(\mathbf{X}, C_j)$. If the current level is a leaf node, RPTA designates the label of the leaf to the query \mathbf{X} .

Since each local cell has its own DKL projection, in order to logically compare between two different cells, we use a measurement called Mixture Distance (R_d).

Definition 19 Let C be the center of the region P , V be the projection matrix to the MEF space and W be the projection matrix to the MDF space. The Mixture Distance (MD) from a query \mathbf{X} fovea vector of the center C is defined as follows.

$$R_d(\mathbf{X}, C) = \sqrt{\|\mathbf{X} - VV^t\mathbf{X}\|^2 + \|VWW^tV^tC - VWW^tV^t\mathbf{X}\|^2}$$

Intuitively, what is being measured can be seen in Fig. 8.3. In Fig. 8.3, the original image space is a 3D space, the MEF space is a 2D subspace, and the MDF space is 1D subspace since two classes are well separated along the first MDF vector. The first term under the radical indicates the distance of the original vector from the population which indicates how well the MEF subspace represents the query vector \mathbf{X} . This term is necessary since it is entirely possible that a query vector that is miles away from a particular subregion's MEF subspace would project very near to the region's center. The second term indicates the distance between the MDF components of the query vector and the MDF components of the center vector in the original image space.

8.3 Convergence of the Approximators

Due to a high complexity and undetermined nature of the way in which a learning set L is drawn from the real world, it is effective to consider that $X(L)$, the set of samples in S , is generated randomly. We know that a fixed L is a special case of random L in that the probability distribution is concentrated at the single location. Thus, we consider \mathbf{X} in $X(L)$ as a random sample from S . The learning set L is

generated by acquiring samples from S with a d -dimensional probability distribution function $F(\mathbf{X})$.

Definition 20 *A point $\mathbf{X}_0 \in S$ is positively supported if for any $\delta > 0$ we have $P\{\|\mathbf{X} - \mathbf{X}_0\| \leq \delta\} > 0$, where $P\{e\}$ denotes the probability of the event e .*

If S consists of a finite number of discrete points, a point \mathbf{X} in P is positively supported means that the probability of selecting \mathbf{X} as a sample is not a zero-probability event. If S consists of infinitely many points, a point \mathbf{X} in P is positively supported means that in any small neighborhood centered at \mathbf{X} , the probability of selecting any point in the neighborhood is not a zero-probability event. In practice, we are not interested in cases that almost never appears in a real-world application. An approximate function \hat{f} can assume any value in subregions of S that will never be used in the application, without hurting the real performance of the system. Thus, we just need to investigate how well the approximation can do at points \mathbf{X} 's that are positively supported.

Definition 21 *A point $\mathbf{X}_0 \in S$ is an interior point of a region $f(\mathbf{X}) = D$ if there is a $\delta > 0$ such that for any \mathbf{X} we have $f(\mathbf{X}_0) = f(\mathbf{X}) = D$, where $\|\mathbf{X}_0 - \mathbf{X}\| < \delta$.*

For a classification problem, we have a set of discrete number of categories to be assigned to the input. Then, $f(\mathbf{X})$ is continuous only at the interior points.

Theorem 9 *Suppose the query vector \mathbf{X} is an interior and positively supported point in a bounded S . Let n be the size of a training set L . Given any small number*

$\epsilon > 0$, there is a number N , so that as long as we independently draw $n > N$ learning samples, the recursive partition tree approximator \hat{f} has the following property

$$P\{\hat{f}(\mathbf{X}) \neq f(\mathbf{X})\} < \epsilon,$$

where \hat{f}' the approximator based on the recursive partition tree.

Proof of Theorem 9. Let \mathbf{X}' be the nearest neighbor of the query \mathbf{X} in the training set $\cup_{i=1}^n X(L_i)$. We show that as long as we independently draw the training samples, the probability that \mathbf{X} traverses the recursive partition tree exactly same as \mathbf{X}' approaches 1 as $n \rightarrow \infty$. Assume at the subregion $P_{i,j}$, we have the projection matrices V and W . The Mixture distance from \mathbf{X} to a center C under region $P_{i,j}$ is

$$R_d^2(\mathbf{X}, C) = \|\mathbf{X} - VV^t\mathbf{X}\|^2 + \|VWW^tV^t\mathbf{X} - VWW^tV^tC\|^2$$

Using the *triangle inequality* property of the Mixture distance, we have

$$\begin{aligned} R_d^2(\mathbf{X}, C) &\leq \|\mathbf{X} - \mathbf{X}'\|^2 + \|\mathbf{X}' - VV^t\mathbf{X}'\|^2 + \|VV^t(\mathbf{X} - \mathbf{X}')\|^2 \\ &\quad + \|VWW^tV^t\mathbf{X}' - VWW^tV^tC\|^2 + \|VWW^tV^t(\mathbf{X} - \mathbf{X}')\|^2 \\ &= R_d^2(\mathbf{X}', C) + \|\mathbf{X} - \mathbf{X}'\|^2 + \|VV^t(\mathbf{X} - \mathbf{X}')\|^2 + \|VWW^tV^t(\mathbf{X} - \mathbf{X}')\|^2 \end{aligned}$$

If the sum of the last three terms (S_3) of the above equation $\rightarrow 0$, we would have $R_d(\mathbf{X}, C) = R_d(\mathbf{X}', C)$ which means that the approximator would make the same decision for \mathbf{X} and \mathbf{X}' . Let D_n be the event that the nearest neighbor of \mathbf{X} in the

training set $\bigcup_{i=1}^n X(L_i)$ has a distance larger than η while E_i denotes the event that the nearest neighbor of \mathbf{X} in the single $X(L_i)$ has a distance larger than η . Since each L_i is independently and identically drawn, we have

$$P\{D_n, \eta\} = \prod_{i=1}^n P\{E_i, \eta\}.$$

On the other hand, \mathbf{X} is a positively supported, which means that $1 - P\{E_i, \eta\} = \beta > 0$. Thus,

$$P\{D_n, \eta\} = (1 - \beta)^n$$

which approaches zero when $n \rightarrow \infty$. So, given any value $S_3 = \eta$ and $\epsilon > 0$, we can have a positive N , so that for any $n > N$, we have

$$P\{D_n, \eta\} < \epsilon.$$

Now, we have shown that the approximator makes the same decision for \mathbf{X} and \mathbf{X}' as $n \rightarrow \infty$ ¹. Finally, the fact that \mathbf{X} is an interior point guarantees that $f(\mathbf{X})=f(\mathbf{X}')$ as $n \rightarrow \infty$.

Theorem 9 means that the RPTA approaches f *pointwisely* in probability: $P\{\hat{f}(\mathbf{X}) \neq f(\mathbf{X})\} \rightarrow 0$, as the size of training set L increases without bound.

¹This may not be true if \mathbf{X}' lies on the decision boundary. However, the event that \mathbf{X}' lies on the decision boundary is a zero probability event.

8.4 k Nearest Neighbors

One drawback of using decision tree to do recognition is that it is unable to reject the undesirable input. In this section, we present a method to measure the confidence of the recognition result based on k nearest neighbors. The confidence can be used as a criteria to do rejection.

Definition 22 *Given a learning set L and k nearest neighbors $(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k)$ of a testing sample \mathbf{X} , the confidence level of \mathbf{X} belonging to class c is defined as*

$$l = \sum_{i=1}^k m(\mathbf{X}_i, c) \alpha^{1 - \frac{\|\mathbf{X} - \mathbf{X}_i\|}{(\epsilon + \|\mathbf{X} - \mathbf{X}_1\|)}}.$$

In the above definition, $m(\mathbf{X}_i, c)$ is a membership function which takes value 1 if \mathbf{X}_i is a member of class c , otherwise it takes -1 . ϵ is a small positive number to avoid the denominator to become zero. Intuitively speaking, this confidence level is the sum of weight of each neighbor. The weight is inversely proportional to the distance between the query and the neighbor. The distance here is the Mixture Distance. The value of α determines how fast the weight will decrease for other runner up. A points \mathbf{X}_i at twice the distance compared to that of the nearest neighbor \mathbf{X}_1 will have its weight decreased by a factor of $1/\alpha$.

8.5 Experimental Results

The framework has been applied to recognize the twenty eight different signs as illustrated in the Fig. 1.2. The image sequences are obtained while subjects perform

hand signs in front of a video camera. The variation of hand size in images is limited.

Two different lighting conditions are used. In the current implementation, each hand sign was represented by five images sampled from the video. Figure 7.9 shows several examples of these sequences.

We first applied our segmentation scheme as discussed in Chapter 7 to segment hands from input images. Then we construct the fovea vectors as shown in Chapter 5.2. These fovea vectors were used as the input for sign recognition. The problem now is how to deal with the sequences which has some images that have been rejected by the segmentation routine. In this case, we still output those sequences because there are still good chances that they can be recognized if only one or two images in the sequences are rejected while the rest of them are fine. The number of images used in the training is 3300 (660 sequences). The number of testing images is 805 (161 sequences).

8.5.1 Results of the nearest neighbor approximator in the MEF space

We show some experimental results to indicate the performance of the nearest neighbor approximator in the MEF space. We computed MEF's using 660 training sequences. Fig. 8.4 shows top 10 MEF's.

The number of MEF's was selected based on the variation ratio $r = \sum_{i=1}^m \lambda_i / \sum_{i=1}^n \lambda_i$, where m out of n MEF vectors were used, as defined in Section 6.1.1. Table 8.1 shows the number of MEF's corresponding to the variation ratio.

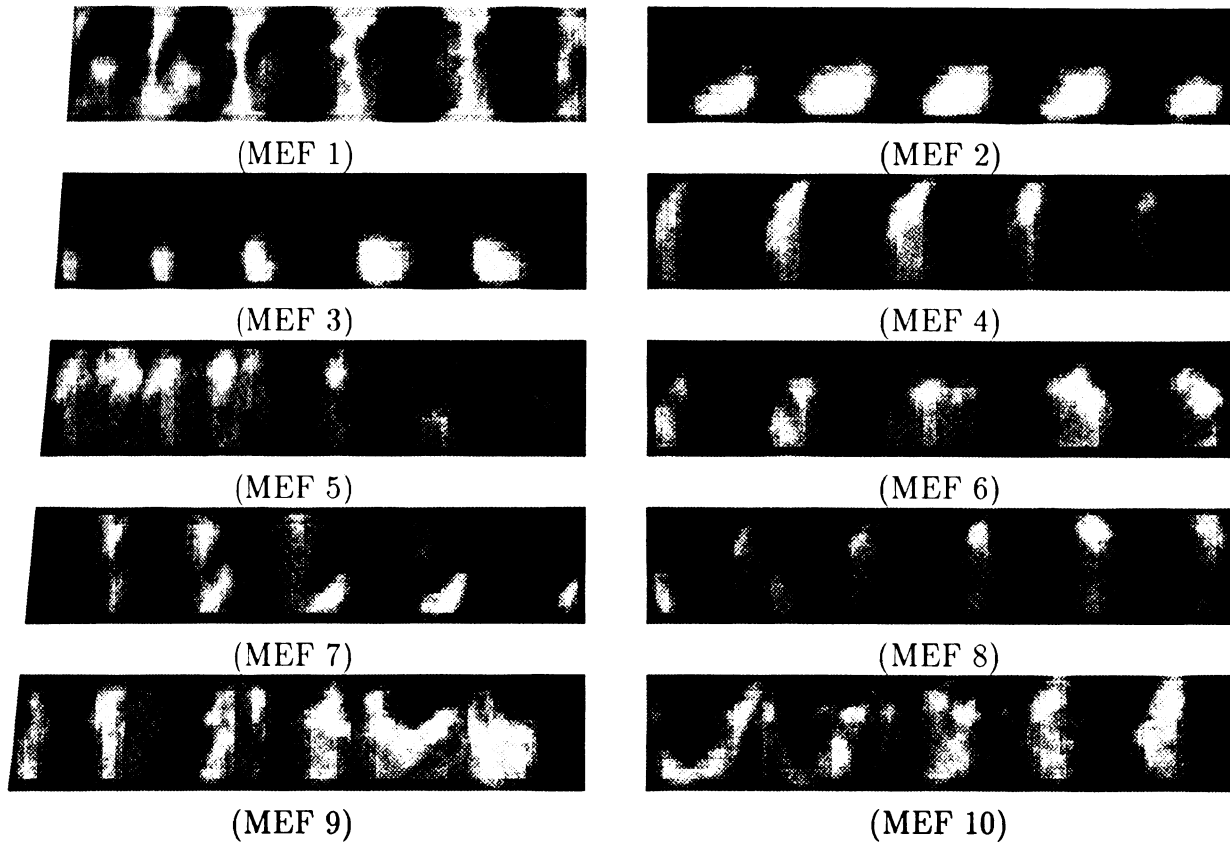


Figure 8.4: Top ten MEF's

Fig. 8.5 shows the performance of the nearest neighbor approximator under the different variation ratio. The performance first improves when the ratio r increases. Then, at the point $r = 0.4$, the performance saturates at the recognition rate 87.0%. Fig. 8.6 shows average computation time for each sequence on a SGI INDIGO 2. The time was obtained based on the two different nearest neighbor query approaches,

Table 8.1: The number of MEF's vs. the variation ratio

The variation ratio	The number of MEF's
10%	1
20%	2
40%	6
80%	48
95%	125

namely, the linear search and the hierarchical quasi-Voronoi diagram in 7.2.3. The use of the hierarchical quasi-Voronoi diagram approach dramatically improves the query time.

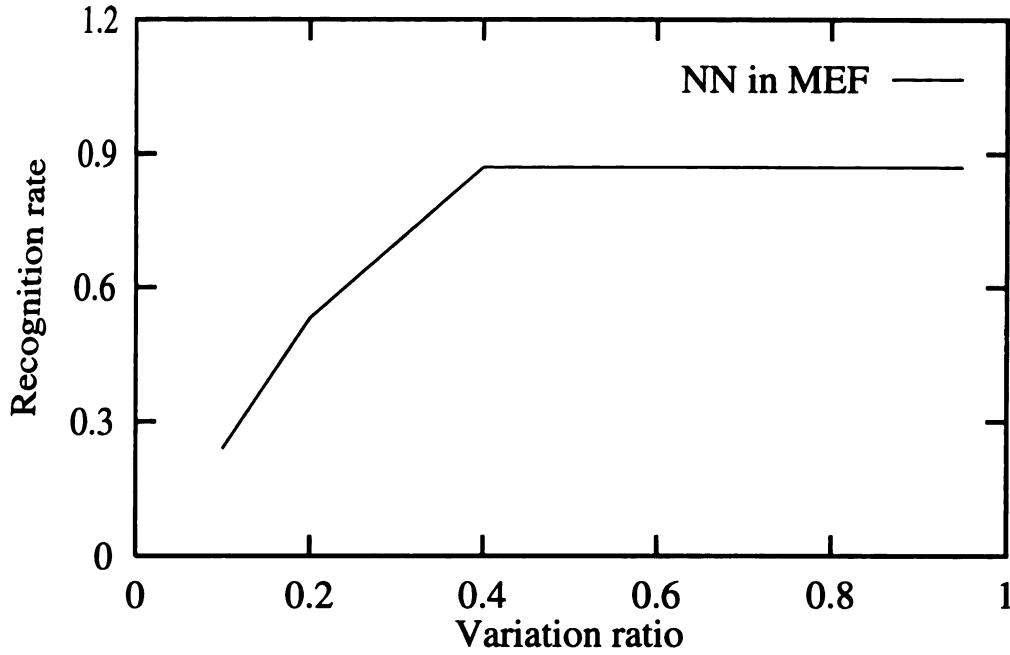


Figure 8.5: Performance of the nearest neighbor approximator in the MEF space. The performance is given as a function of the number of MEF's used.

8.5.2 Results of the recursive partition tree approximator in the MDF space

In this experiment, the same 660 training sequences were used to build a recursive *partition tree*. For each nonterminal region, we selected an adaptive radius r as *defined in Section 8.2.3* to split the region into subregions. Given r for a nonterminal region, we have $k > 1$ training samples and the distance between each pair of these k samples is greater than r . These k samples were the centers to generate a Voronoi diagram. The distance here is the Euclidean distance in the MDF space corresponding

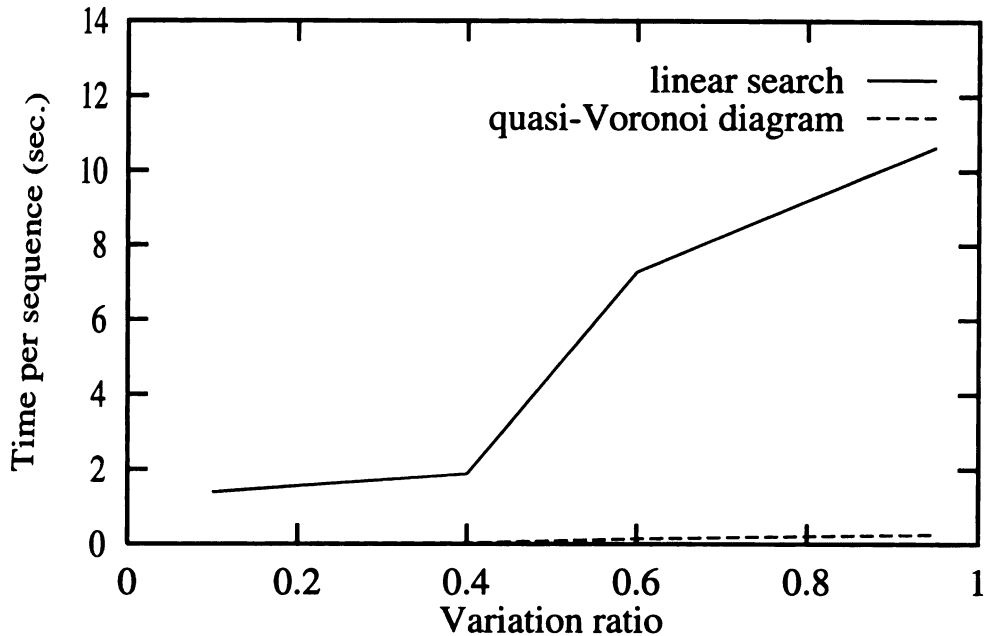


Figure 8.6: Performance of the two different nearest neighbor query approaches: linear vs. quasi-Voronoi diagram.

to the region. Fig. 8.7 shows the top 10 MDF's at the root level.

Once we have created the recursive partition tree, we used it to recognize the sign. As we did in the experiments for the nearest neighbor approximator in the MEF space, the segmentation result was used as the input for sign recognition. The results are summarized in Table 8.2. The correct recognition rate of 161 testing sequences is 93.2% which is better than the recognition rate (87.0%) of the nearest neighbor approximator in the MEF space. The average recognition time per sequence is 0.63 second on a SGI INDIGO 2. The time is longer than the time (0.27 seconds) of the nearest neighbor approximator when the quasi-Voronoi diagram is used in the query. This is because each nonterminal node in the recursive partition tree has its own DKL projection matrices and each time the query vector traverses the node, it has to go through the local projection, whereas in the case of the nearest neighbor

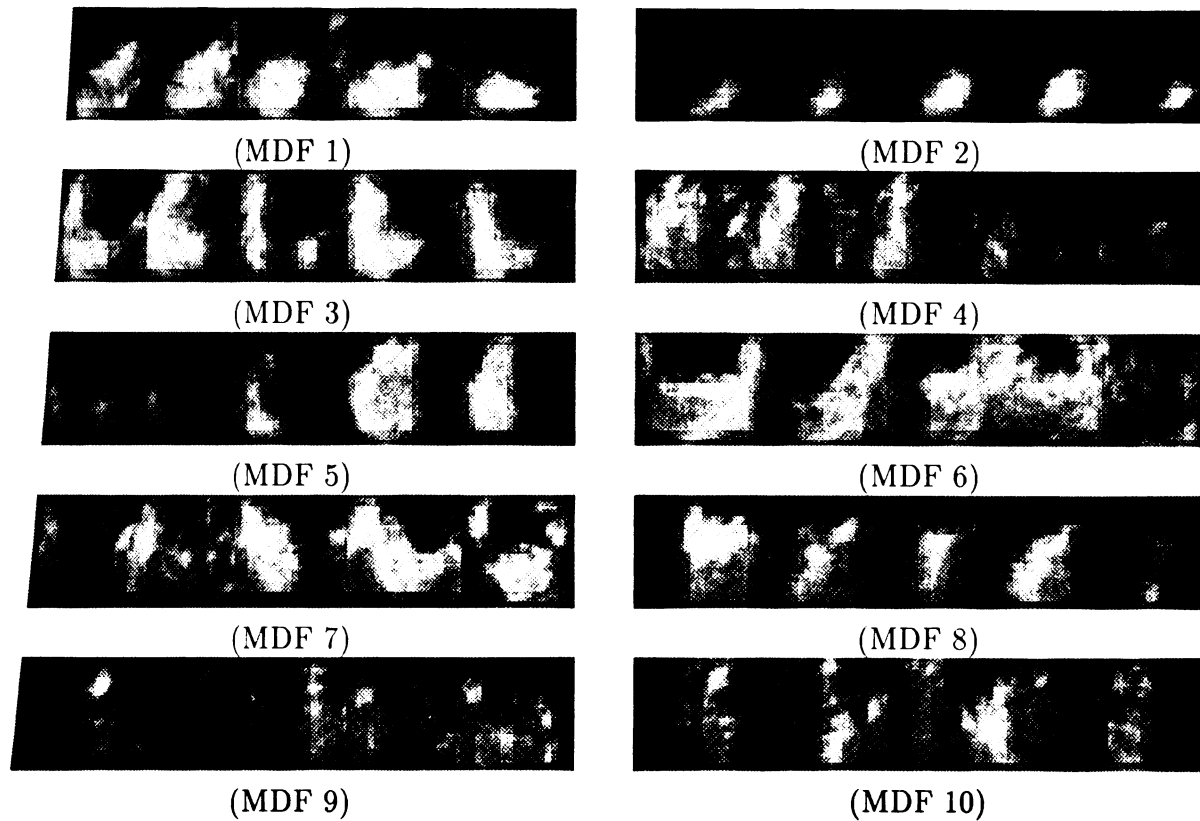


Figure 8.7: Top ten MDF's

approximator, only one projection is necessary.

Table 8.2: Summary of the experimental data for RPTA

<i>Training</i>		<i>Testing</i>	
Number of training samples	660 (3300 images)	Number of testing sequences	161 (805 images)
Height of the tree	7	Recognition rate	93.1% (87% for MEF)
Number of nodes	90	Time per sequence (sec.)	0.63

8.5.3 k nearest neighbors

There are 16 sequences in the above 161 testing sequences and in each of these 16 sequences, at least two out of five images have bad segmentation results. What we want to do is to reject these sequences. We used the method of k nearest neighbors (see Chapter 8.4) to evaluate the confidence level of each recognition result. Then, we set up a threshold and reject any result which has a confidence level below the threshold. In the experiments, we used the top 5 nearest neighbors and selected 1.1 as the confidence threshold. Here, each of the top 5 nearest neighbors can vote for the result. The weight of voting power of the result R_i equals to $\frac{\|\mathbf{X}-R_n\|}{\|\mathbf{X}-R_i\|}$ where \mathbf{X} is the test sample and R_n is the nearest neighbor of the test sample. The sign of the weight is positive if it agrees with the vote of the nearest neighbor, otherwise, it is negative. The weight for the nearest neighbor is 1.0. The selection of the threshold is heuristic. The threshold 1.1 requires slightly more support than a single nearest neighbor. The system accepts 89% of the input sequences. Within these 89% of the input sequences, the correct recognition rate is 97%. Among those 11% (18) rejected sequences, 56% (10) sequences have segmentation error. The experimental results are shown in Table 8.3.

8.5.4 Experiments related to MDF

We have shown that the approximator in the MDF space has better performance than the one in the MEF space. This is because the pixel-to-pixel distance, whether in the original image space or the MEF space, can not well characterize the difference

Table 8.3: Summary of the experimental data for k NN

Accept rate	89%
Reject rate	11%
Recognition rate among the accepted	97%
Misclassification rate among the accepted	3%
False reject due to segmentation error among the rejected	56%
False reject due to recognition error among the rejected	44%

between two signs due to the effects such as lighting, viewing angle, and hand variation between different subjects. On the other hand, the MDF's are the features that best characterize different categories. In this section, we show some experimental results to indicate quantitatively how the MEF and the MDF may perform very differently in classifying signs.

Clustering effects

We computed MEF's and MDF's, respectively, using 50 sequences (10 for each signs). These signs are obtained from different subjects and the viewing positions are slightly different. Fig. 8.8 (a) shows the samples in the subspace spanned by the first two MEFs and Fig. 8.8 (b) shows them in the subspace spanned by the first two MDFs. As clearly shown, in the MEF subspace, samples from a single class spread out widely and samples of different classes are not far apart. In fact, some samples from different classes mingle together. However, in the MDF subspace, samples of each class are clustered more tightly and samples from different classes are farther apart. This shows that the MDFs are better in terms of classification of signs.

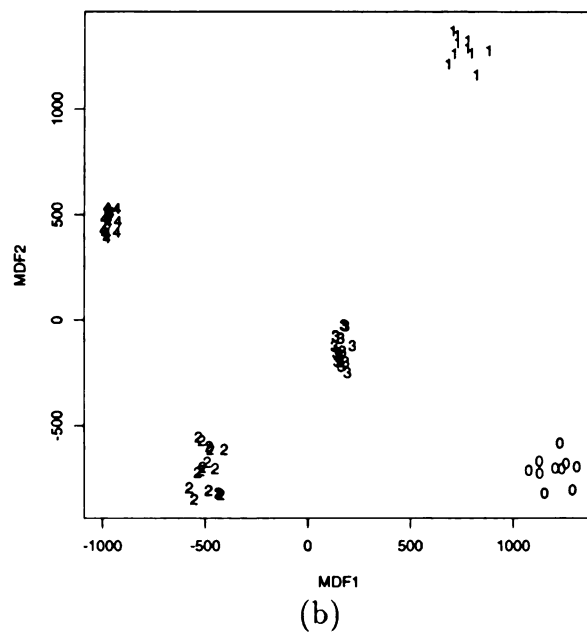
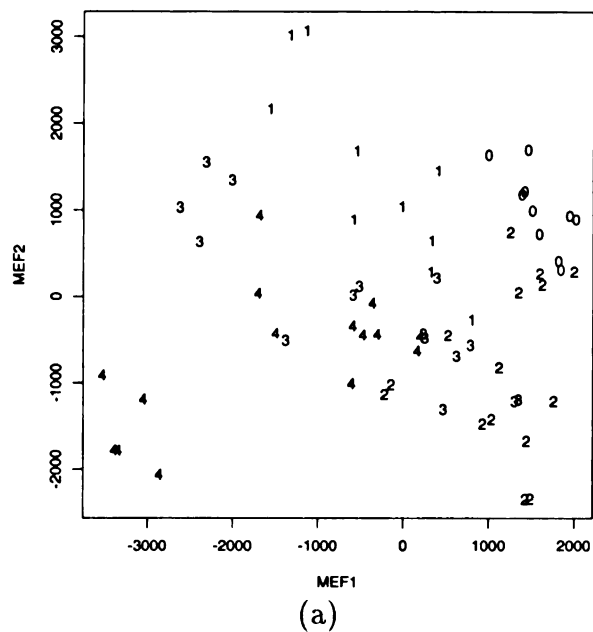


Figure 8.8: The difference between MEF and MDF in representing samples. (a) Samples represented in the subspace spanned by the first two MEFs. (b) Samples represented in the subspace spanned by the first two MDFs. The numbers in the plot are the class labels of the samples.

Geometric meaning of the MDF

In the MDFs, factors that are not related to classification are discarded or weighted down, which is accomplished by minimizing the within-class scatter; factors that are crucial to classification are emphasized, which is achieved by maximizing the between-class scatter. In this experiment, we show an example that the MDFs can capture the important geometric features.

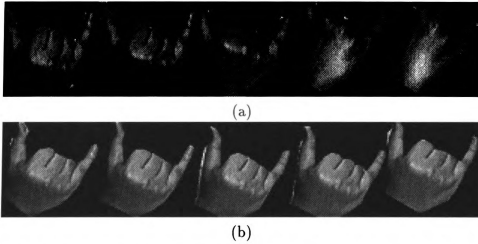


Figure 8.9: Two sample sequences of signs “of course” (a) and “wrong” (b).

In our gesture vocabulary, the image sequences of two signs: “of course” and “wrong” are visually very similar. Fig. 8.9 illustrates two sample sequences of the above signs. The nearest neighbor approximator generally has difficulty to distinguish them, but not the recursive partition tree approximator in the MDF space. Fig. 8.10 shows the difference between the MEF and the MDF. The left sequence in Fig. 8.10 is a reconstruction of the sequence “of course” based on the first MDF and the right sequence is a reconstruction of the same sequence using 95% of MEFs. We can see that the MEFs are good in terms of preserve the information but not much help

for classification. On the other hand, the first MDF captures the feature locations (edges) because it accounts for the major between-sign variation.

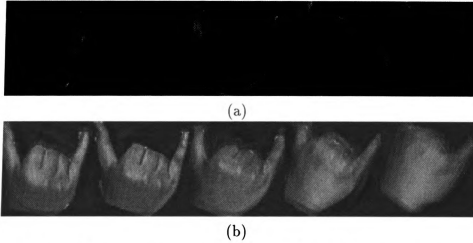


Figure 8.10: The difference between the MEF and MDF. (a) Reconstruction based on the first MDF. (b) Reconstruction based on 95% MEFs.

8.6 Conclusions

In this chapter, we have presented a new approach to recognize hand signs. In our approach, motion understanding (the hand movement) is tightly coupled with spatial recognition (hand shape). To achieve a high applicability and adaptability to various conditions, we do not impose prior features that the system must use, but rather the system automatically selects features from images during learning. The system uses multiclass, multidimensional discriminant analysis to automatically select the most linear discriminating features for gesture classification. The recursive partition tree approximator is proposed to do classification. This approach combined with our previous work on the hand segmentation forms a new framework which addresses three key aspects of the hand sign interpretation, that is, the hand shape, the location, and

the movement. The framework has been tested to recognize 28 different hand signs. The experimental results have shown that the system achieved a 93% recognition rate without a rejection option. The recognition rate was 97% with a 11% reject rate.

Chapter 9

Summary and Future Work

This chapter summarizes the results of the research described in this thesis. Several directions for future research are also outlined.

9.1 Summary

Two pieces of work are reported in this thesis. The first one is about motion and structure estimation using image sequences. The second one is about recognition of hand signs using intensity image sequences.

9.1.1 Integration of transitory image sequences

We have developed a system to estimate motion and structure from transitory image sequences. A transitory image sequence is one in which no scene element is visible through the *entire* sequence. When a camera system scans a scene which cannot be covered by a single view, the image sequence is transitory. We have shown that

from a transitory sequence it is inherently not possible to get better estimates with a longer sequence. The later a scene part comes into the sequence, generally the worse its global accuracy compared to that in the first view. We establish asymptotic error rates with respect to the number of frames, which indicate how the error in the estimates evolves with time and how to minimize the pace of error accumulation. Some concise expressions have been derived in terms of asymptotic error rate for different representations, processing methods, and image sequence types. The asymptotic error rates are in fact the lowest possible error rates based on the Cramér-Rao error bound. We have proposed two different techniques for two different usages of the results: global and local (e.g., visual map generation and global pose determination for the former and obstacle avoidance and object manipulation belong for the latter).

We conducted experiments with synthetic and real world images in order to experimentally examine the error rates and compare the two representations (WC and CC). The performance of the algorithm can be demonstrated through simulations, where ground truth and the amount of noise can be well controlled and the errors in the estimates can be accurately measured. In the simulation, 3-D feature points were generated randomly for each trial, between depth 2000mm and depth 3000mm, with a uniform distribution. The entire scene is covered by 31 frames and the distance between the consecutive frames is roughly 200mm. A small rotation was added between each pair of two consecutive frames. Average errors were obtained through 100 random trials each with a different set of 3D points. The results demonstrated that different representations have very different stabilities. In general, the world-centered (WC) coordinate system is better for a global usage and the camera-centered (CC)

coordinate system is superior for a local usage.

In order to provide actual accuracy with a real system setup, careful experiments have been conducted with a fully calibrated camera system. The setup used for our image acquisition was a Denning MRV-3 mobile robot and a pair of stereo cameras. The stereo camera system was calibrated with distortion compensation using an algorithm from Weng *et al* [190]. An image sequence of 151 frames was acquired from the moving mobile robot. A temporally subsampled (one sample every 5 frames) subsequence of 31 frames was used for motion and structure estimation with a consideration that this subsequence is dense enough for estimation and yet enables cross-frame motions to cover more original frames with a relatively small batch size.

The algorithm includes feature selection, stereo matching, temporal matching and tracking, 3D structure integration, and motion and pose estimation. The results have been compared with the ground truth of the test points on the scene and the pose of the camera system. As we predicted, the error increases with the time. But the estimates appear good. After traveling about 3000mm, the estimated camera global position error is less than 60mm in depth Z (less than 2.3%), about 43mm horizontally and under 25mm vertically with the WC representation. This seems to indicate that reasonable results can be obtained, even with a transitory image sequence.

9.1.2 Hand Sign Recognition

In this thesis, we have presented a new general framework to learn and recognize hand signs from intensity image sequences. The framework has two major components,

namely, segmentation and recognition.

A prediction-and-verification scheme using attention images from multiple fixations is presented to segment hands from complex backgrounds. During the prediction stage, we first apply the motion-based attention to find the rough position of the hands from the input sequences. Then, the attention images are generated around the initial position. These attention images are used to query the database of the training attention images to predict the mask. One important feature of small attention images which focus on the portion of the object is that they typically contain little background. These attention images can be used as important visual cues to segment the object of interest from the input image.

We present a hierarchical quasi-Voronoi tessellation to organize the training attention images. Based on the hierarchical quasi-Voronoi tessellation, we propose a new efficient query algorithm for the nearest neighbor in the high dimensional space. The result of prediction is verified using a learning-based function approximation scheme.

A major advantage of this scheme is that it can handle a large number of different deformable objects presented in complex backgrounds. The scheme is also relatively efficient since the segmentation is guided by the past knowledge through a prediction-and-verification scheme.

In our recognition scheme, motion understanding (hand movement) is tightly coupled with spatial recognition (hand shape). To achieve a high applicability and adaptability to various conditions, we do not impose prior features that the system must use, but rather the system automatically selects features from images during learning. The system uses multiclass, multidimensional discriminant analysis to automatically

select the most discriminating features for gesture classification. A recursive partition tree approximator is proposed to do classification.

The framework has been tested to recognize 28 different hand signs. The experimental results show that the system can achieve a 93% recognition rate without rejection. The recognition rate is 97% with 11% reject rate.

9.2 Future Work

In this section, we discuss a number of research issues which could be addressed in the future.

9.2.1 Integration of transitory image sequences

In current experiments for the real setup, we have only tested one sequence and we realize that in order to fully test the algorithm, we need to collect more data and run the program.

The current design is suitable for a passive navigation system where the navigation of the sensor is guided by the human operator. During a passive navigation, the sensor grabs the image sequences. Later, these images are processed in an off-line fashion to obtain the global structure of the unknown scene. The next question is whether it is possible to build an automatic navigation system. In order to build an automatic navigation system, we must have

- *Real time processing.* Currently, we are unable to do this is because feature extraction and tracking are very time consuming.

- *Scene understanding.* For a goal oriented navigation, the 3D structure information of the scene alone is not enough. The navigator has to understand the scene so that it can follow the road and find the way.

9.2.2 Hand sign recognition

We have presented a three stage framework in Chapter 5. This thesis addresses the stage 2 and 3, which are segmentation and recognition. However, the stage 1 which is the image acquisition is not a trivial problem if the speed of the hand motion is not uniform, for example, person *A* first performs a sign from a slow pace to a fast pace and then performs the same sign changing the pace from fast to slow. Then, we have the problem of selecting the frames to represent the sign since using the same time interval between the consecutive frame could end up with two very different sequences. In that case, time warping is necessary.

In the current implementation of the segmentation, the fixations and the positions of fixations are the same regardless of the types of gestures. This is not very efficient. Some gestures may be very simple so that a few fixations are enough to recognize them. In order to achieve the optimal performance, different gestures require different positions of fixations. In the future, we need to investigate the generation of the fixations based on learning. The previous fixations are used to guide the next action. The next action could be (a) termination of the process of generating fixation if the gesture has already been recognized; or (b) finding the appropriate position for next fixation.

Training in the current implementation is done in a batch fashion. A complete new training session is needed each time we update the training set. This can be very inefficient when the training set becomes large. In the future, we need to investigate the problem of incremental learning.

Finally, we would like to give some personal views on the problem of interpretation of the sign language such as American Sign Language (ASL). Can we extend our current work to interpret ASL for the disabled? The extension faces the following obstacles:

- Currently, we are working at the word level (sign). If we want to understand ASL, we have to understand the sentence made up by a sequence of signs.
- Our experiments only include 28 different signs. Although 28 is a good number in the current state of the art, it is just a small fraction if we compare it with the number of the common words in ASL, which is around 5000.
- There are signs in ASL that use both hands and signs whose meaning of depends on contextual information. Currently our system can not handle these types of signs.

Based on the above discussion, we think there are still a few large gaps between our system and a future system which can interpret ASL.

Bibliography

- [1] J. R. Anderson, *Cognitive Psychology and Its Implications*, 3rd ed., Freeman, New York, 1990.
- [2] G. Adiv, Determining three-dimensional motion and structure from optimal flow generated by several moving objects, in *IEEE Trans. PAMI*, vol. 7, pp. 384-401, July 1985.
- [3] J. Aggarwal and N. Nandhakumar, On the computation of motion from sequences of images-a review, in *Proc. of IEEE*, vol. 76, no. 8, pp. 917-935, 1988.
- [4] K. Akita, Image sequence analysis of real world human motion, in *Pattern Recognition*, vol. 17, pp. 73-83, 1984.
- [5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, Active vision, *Int'l Journal of Computer Vision*, vol. 1, pp. 333-356, 1988.
- [6] J. Aloimonos, Purposive and qualitative active vision, in *IEEE CVPR.*, pp.346-360, 1990.

- [7] P. Anandan and R. Weiss, Introducing a smoothness constraint in a matching approach for the computation of displacement fields, in *Proc. ARPA IUS Workshop*, SAIC, Dec. 1985, pp.186-197.
- [8] K. S. Arun, T. S. Huang and S. D. Blostein, Least-squares fitting of two 3-D point sets, in *IEEE Trans. PAMI*, vol. 9, no. 5, 1987.
- [9] N. Ayache and Lustman, Fast and reliable trinocular stereovision, in *Proc. 1st Intern. conf. Comput. Vis.*, pp. 422-427, London, 1987.
- [10] N. Ayache and O.D. Faugeras, Building, registrating, and fusing noisy visual maps, in *Proc. 1st ICCV*, pp. 73-82, 1987.
- [11] H. Baker and T.O. Binford, Depth from edge and intensity based stereo, *Proc. 7th Joint Conf. Artif. Intell.*, pp.631-636, Vancouver, August, 1981.
- [12] J. L. Barren, A. D. Jepson, and J. K. Tsotsos, The feasibility of motion and structure from noisy time varying image velocity information. in *International Journal of Computer Vision*, 5:239-269, 1990.
- [13] T. Baudel and M. Beaudouin-Lafon, Charade: remote control of objects using free-hand gestures, in *CACM*, pp. 28-35, 1993.
- [14] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, The r^* -tree: an efficient and robust access method for points and rectangles, in *Proceedings of the 1990 ACM SIGMOD*, pp. 322-331, 1990.

- [15] P. J. Besl and N. D. McKay, A method for registration of 3-d shapes, in *IEEE Trans. on PAMI*, vol. 14, pp. 239-256, 1992.
- [16] L. Birnbaum, M. Brand and P. Cooper, Looking for trouble: using causal semantics to direct focus of attention, in *Proc. 4th Int'l Conf. Computer Vision*, pp. 49-56, 1993.
- [17] J. F. Blinn and M. E. Newell, Texture and reflection in computer generated images, in *CACM*, vol. 19, pp. 542-547, 1976.
- [18] A. Bobick, A hybrid approach to structure-from-motion, in *Proc. ACM Interdisc. Workshop Motion*, pp. 91-109, Toronto, Canada.
- [19] A. Bobick and A. Wilson, A state-based technique for the summarization and recognition of gesture, in *Proc. 5th Int'l Conf. Computer Vision*, pp. 382-388, Boston, 1995.
- [20] S. D. Blostein and T. S. Huang, Algorithms for motion estimation based on 3-D correspondences, in *Motion Understanding*, W. Martin and J. K. Aggrawal Eds. Norewell, MA: Kluwer, 1988.
- [21] J. Boissonnat, Geometric structures for three-dimensional shape representation, in *ACM Trans. on Graphics*, vol. 3, no. 4, pp. 266-286, 1984.
- [22] J. Bonvillian, K. Nelson, and V. Charrow, Language and language related skills in deaf and hearing children, in *Sign Language Studies*, vol. 12, pp. 211-250, 1976.

- [23] H. Bornstein and K. Saulnier, *The Signed English Starter*, CLERC BOOKS, Gallaudet University Press, Washington, D.C., 1989.
- [24] P. Bouthemy and E. Francois, Motion segmentation and qualitative dynamic scene analysis from an image sequence, in *International Journal of Computer Vision*, vol. 10, pp. 157-182, 1993.
- [25] O. Braddick, The masking of apparent motion in random dot patterns, in *Vision Res.*, vol. 13, pp. 355-369, 1973.
- [26] O. Braddick, A short-range process in apparent motion, in *Vision Res.*, vol. 14, pp. 519-527, 1974.
- [27] T. Brandt, J. Dichgans and E. Koenig, Differential effects of central versus peripheral vision on egocentric and eccentric motion perception, in *Expl Brain Res.*, vol. 16, pp. 476-491, 1973.
- [28] T. J. Broida and R. Chellappa, Estimation of object motion parameters from noisy images, in *IEEE Trans. PAMI*, vol. 8, no. 1, pp. 90-99, Jan. 1986.
- [29] J. S. Brown, *A Vocabulary of Mute Signs*, Baton Rouge, Louisiana, 1856.
- [30] V. Bruce, *Recognizing faces*, Hillsdale: Lawrence Erlbaum, 1988.
- [31] R. Brunelli and T. Poggio, Face recognition: features versus templates, in *IEEE Trans. PAMI*, vol. 15, no. 10, pp. 1042-1052, 1993.
- [32] A. Bruss and B.K.P. Horn, Passive navigation, in *Massachusetts Inst. Technol.*, Cambridge, A.I. Memo 662, 1981.

- [33] S. Carey, *Conceptual Change in Childhood*, The MIT Press, Cambridge, MA, 1985.
- [34] C. Cédras and M. Shah, A survey of motion analysis from moving light displays, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, WA, pp. 214-221, June 1994.
- [35] S. W. Chen, G. Stockman, C. Dai and C.P. Chuang, Two-stage dynamic deformation for construction of 3D models, working manuscript under review.
- [36] Z. Chen and H. Lee, Knowledge-guided visual perception of 3-D human gait from a single image sequence, in *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, pp. 336-342, 1992.
- [37] B. Chazelle, How to search in history, in *Inf. Control*, vol. 64, pp. 77-99, 1985.
- [38] R. Cipolla, Y. Okamoto and Y. Kuno, Robust structure from motion using motion parallax, in *IEEE Conf. CVPR.*, pp. 374-382, 1993.
- [39] W. F. Clocksin, Perception of surface slant and edge labels from optical flow: a computational approach, in *Perception*, 1980.
- [40] L. D. Cohen, On active contour models and balloons, in *Image Understanding*, vol. 53, pp. 211-218, 1991.
- [41] E. Costello, *Signing: how to speak with your hands*, Bantam Books, New York, 1983.

- [42] G. W. Cottrell, P. W. Munro and D. Zipser, Image compression by back propagation: A demonstration of extensional programming, in N.E. Sharkey (Ed.), *Advances in cognitive science*, vol. 2, Chichester, England: ellis Horwood, 1987.
- [43] T. M. Cover and P. E. Hart, Nearest neighbor pattern classification, in *IEEE Tran. on Information Theory*, vol. 13, pp. 21-27, 1967.
- [44] N. Cui, J. Weng, and P. Cohen, Extended structure and motion analysis from monocular image sequences, in *Proc. 3rd Int'l Conf. on Computer Vision*, Osaka, Japan, pp. 222-229, 1990.
- [45] N. Cui, Weng and Cohen, Motion and structure from long stereo image sequences, in *IEEE Conf. CVPR*, pp. 75-80, 1991.
- [46] Y. Cui, D. Swets and J. Weng, Learning-Based Hand Sign Recognition Using SHOSLIF-M, in *Proc. Int'l Conf. Computer Vision*, pp. 631-636, Boston, June, 1995.
- [47] Y. Cui and J. Weng, 2D object segmentation from fovea images based on eigensubspace learning, in *Proc. International Symposium on Computer Vision*, pp. 305-310, Coral Gables, FL, Nov., 1995.
- [48] Y. Cui and J. Weng, Learning-based object segmentation for fovea images, in the *2nd Asian Conf. on Computer Vision*, Vol II, pp.71-75, Singapore, Dec. 1995.
- [49] Y. Cui and J. Weng, Learning-Based Hand Sign Recognition, in *Proc. International Workshop on Automatic Face- and Gesture-Recognition*, pp. 201-206, Zurich, Switzerland, June 1995.

- [50] Y. Cui and J. Weng, Hand segmentation using learning-based prediction and verification for hand-sign recognition, to appear in *IEEE Conf. Computer Vision and Pattern Recog.*, 1996.
- [51] Y. Cui and J. Weng, "View-Based Hand Segmentation and Hand-Sequence Recognition with Complex Backgrounds", accepted for presentation in *13th International Conference on Pattern Recognition*, 1996.
- [52] K. Daniilidis and H. Nagel, The coupling of rotation and translation in motion estimation of planar surfaces, in *IEEE Conf. CVPR*, pp. 188-193, 1993.
- [53] T. Darrell and A. Pentland, Space-time gestures, in *IEEE CVPR*, pp.335-340, 1993.
- [54] J. Davis and M. Shah, Visual gesture recognition, in *IEE Proc. Vis. Image Signal Process*, vol. 141, No. 2, pp. 101-106, April 1994.
- [55] D. P. Dobkin and R. J. Lipton, Multidimensional searching problems, in *SIAM J. Comput.*, vol. 5, pp. 181-186, 1976.
- [56] G. W. Donohoe, D. R. Hush and N. Ahmed, Change detection for target detection and classification in video sequences, in *Proc. Int'l Conf. Acoust., Speech, Signal Processing*, pp. 1084-1087, 1988.
- [57] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
- [58] P. Ekman, *Unmasking the Human Face*, New York: Prentice-Hall, 1971.

- [59] T. F. Elbert, *Estimation and Control of Systems.*, New York, Van Nostrand Reinhold, 1984.
- [60] J. L. Elman and D. Zipser, Discovering the hidden structure of speech, in *Journal of the Acoustical Society of America*, vol. 83, pp. 1615-1626, 1988.
- [61] J. L. Elman, Finding structure in time, in *Cognitive Science*, vol. 14, pp. 179-211, 1990.
- [62] I. Essa and A. Pentland, A vision system for observing and extracting facial action parameters, in *IEEE Conf. CVPR*, pp. 76-83, 1994.
- [63] J. Q. Fang and T.S. Huang, Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames, in *IEEE Trans. PAMI*, vol. 6, pp. 547-554, 1984.
- [64] O. Faugeras, E. Bras-Mehlman and J. Boissonnat, Representing stereo data with the Delaunay triangulation, in *Artificial Intelligence*, vol. 44, pp.41-87, 1990.
- [65] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, Cambridge, MA, 1993.
- [66] S. S. Fels, *Building Adaptive Interfaces with Neural Networks: The Glove-Talk Pilot Study*, Tech. Report No. CRG-TR-90-1, Dept. of Computer Science, University of Toronto, Canada, 1990.
- [67] K.Finn and A. Montgomery, Automatic optically-based recognition of speech, in *Pattern Recognition Letters*, vol. 8, pp. 159-164, 1988.

- [68] W. Frieman, *About time: inventing the fourth dimension*, MIT Press, Cambridge, MA, 1990.
- [69] S. Furui, *Digital Speech Processing, Synthesis, and Recognition*, pp. 244-249, Marcel Dekker Inc., New York and Basel, 1989.
- [70] S. Ganapathy, Decomposition of transformation matrices for robot vision, in *Pattern Recog. Letter*, vol. 2, pp. 401-412, 1989.
- [71] A. Gelb, *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.
- [72] J.J. Gibson, Visually controlled locomotion and visual orientation in animals, in *Br. J. Psychol.*, 1954.
- [73] J.J. Gibson, *The Ecological Approach to Visual Perception*, Houghton-Mifflin, Boston, MA, 1979.
- [74] C. M. Ginberg and D. Maxwell, Graphical marionette, in *Proc. ACM Siggraph/Sigart Workshop on Motion*, pp. 172-179, ACM Press, New York, 1983.
- [75] A. Goshtasby, Recovering scene structures from scattered surface points, in *Pattern Recognition*, vol. 26, no. 10, pp. 1543-1547, 1993.
- [76] R. L. Gregory, *Eye and Brain*, World University Library.
- [77] U. Grenander, Y. Chow, and D. M. Keenan, *Hands. A Pattern Theoretic Study of Biological Shapes*, Springer, 1991.
- [78] W. E. L. Grimson, A computer implementation of a theory of human stereo vision, in *Phil. Trans. Roy. Soc. London*, vol. B292, pp. 217-253, 1981.

- [79] W. E. L. Grimson, *From Images to Surfaces*, MIT Press: Cambridge, MA, 1981.
- [80] W. E. L. Grimson, Computational experiments with a feature based stereo algorithm, in *IEEE Trans. PAMI*, vol. 7 no. 1, pp. 17-34, 1985.
- [81] S. Gwydir, H. Buettner and S. Dunn, Non-rigid motion analysis and feature labeling of the growth cone, in *Proc. IEEE Workshop on Biomedical Image Analysis*, pp. 80-87, 1994.
- [82] A. Guttman, R-trees: a dynamic index structure for spatial searching, in *ACM SIGMOD*, pp. 905-910, 1984.
- [83] J. A. Hall, *The Human Interface in Three-Dimensional Computer Art Space*, MSVS thesis, Media Lab, MIT, Cambridge, MA, 1985.
- [84] T. Hastie and W. Stuetzle, Principal curves, in *Journal of the American Statistical Association*, vol. 84, pp. 502-516, 1989.
- [85] D. J. Heeger and A. D. Jepson, Subspace methods for recovering rigid motion I: Algorithm and implementation, in *International Journal of Computer Vision*, 7:95-117, 1992.
- [86] D. Hogg, Model based vision: a program to see a walking person, in *Image and Vision Comp.*, vol. 1, pp. 5-20, 1983.
- [87] B. K. P. Horn, Closed-form solution of absolute orientation using unit quaternions, in *J. Opt. Soc. Amer.*, vol. 4, pp. 629-642, 1987.

- [88] T. S. Huang and Y. S. Shim, Linear algorithm for motion estimation: how to handle degenerate cases, in *Proc. British Pattern Recog. Association Conf.*, Cambridge, England, 1987.
- [89] T. S. Huang (ed.), *Advances in Computer Vision and Image Processing, Vol. 3: Time-Varying Imagery Analysis*, JAI Press, Greenwich, Connecticut 1988.
- [90] T. S. Huang and O. D. Faugeras, Some properties of the e-matrix in two-view motion estimation, in *IEEE PAMI*, vol. 11, no. 12, pp.1310-1312, 1989.
- [91] D. H. Hubel, *Eye, Brain, and Vision*, Scientific American Library, Vol. 22, 1988.
- [92] R. Ivry and A. Cohen, Dissociation of short- and long-range apparent motion in visual search, in *J. of Exp. Psy.: Human Perception and Performance*, vol. 16, No. 2, pp. 317-331, 1990.
- [93] A. K. Jain, Y. Zhong, and S. Lakshmanan, Object matching using deformable templates, in *IEEE Trans. on PAMI*, vol. 18, pp. 267-278, 1996.
- [94] M. I. Jordan, Serial order: A parallel distributed processing approach (Tech. Rep. No. 8604), San Diego: Univ. of California, Institute for Cognitive Science, 1986.
- [95] G. Johansson, Spatial-temporal differentiation and integration in visual motion perception, in *Psychol. Research*, vol. 38, pp. 379-393, 1976.
- [96] G. Johansson, Visual motion perception, in *Scientific American*, vol. 232, pp. 76-88, June, 1976.

- [97] P. N. Johnson-Laird, *The Computer and the Mind: An introduction to cognitive science*, Harvard Univ. Press, Cambridge, MA., 1988.
- [98] M. Kass, A. Witkin and D. Terzopoulos, Snakes: active contour models, in *Proc. 1st ICCV*, pp. 259-268, 1987.
- [99] Y. Kaya and K. Kobayashi, A basic study on human face recognition, in *Frontiers of Pattern Recognition* (S. Watanabe Ed.), pp. 265, 1972.
- [100] C. Kervrann and F. Heitz, A hierarchical statistical framework for the segmentation of deformable objects in image sequences, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 724-728, 1994.
- [101] S. M. Kiang R. J. Chou and J. K. Aggarwal, Triangulation errors in stereo algorithms, in *Proc. IEEE Workshop Computer Vision* (Miami Beach, FL), pp. 72-78, 1987.
- [102] M. Kirby, F. Weisser, and G. Dangelmayr, A model problem in the representation of digital image sequences, in *Pattern Recognition*, vol. 26, pp. 63-73, 1993.
- [103] V. Klee, On the complexity of d -dimensional Voronoi diagrams, in *Arch. Math.*, vol. 34, pp. 75-80, 1980.
- [104] D. Knuth, *The Art of Computer Programming III: Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.
- [105] T. Kohonen, *Self-Organization and Associative Memory*, 2nd., Springer-Verlag, Berlin, 1988.

- [106] L.T. Kozlowski and J.E. Cutting, Recognizing the sex of a walker from dynamic point-light display, in *Perception and Psychophys.*, vol. 21, no. 6, pp.575-580, 1977.
- [107] J. Kramer and L. Leifer, *The Talking Glove: An Expressive and Receptive "Verbal" Communication Aid for the Deaf, Deaf-Blind, and Nonvocal*, Tech. Report, Stanford University, Dept. of Electrical Engineering, Stanford, Calif., 1989.
- [108] J. D. Krol and W. A. Grind, The double nail illusion: experiments on binocular vision with nails, needles and pins, in *Perception* 9, 651.
- [109] E. Kruppa, Zur Ermittlung eines Objektes auss zwei Perspektiven mit innerer Orientierung, in em Sitz-Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. IIa., 122:1939-1948, 1913.
- [110] J. J. Kuch and T. S. Huang, Vision based hand modeling and tracking, in *Proc. International Conference on Computer Vision*, June, 1995.
- [111] R. Kumar, H. S. Sawhney and A. R. Hanson, 3D model acquisition from monocular image sequences, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Champaign, IL, pp. 209-215, 1992.
- [112] D. Lee and E. Aronson, Visual proprioceptive control of standing in human infants, in *Percept. Psychophys.*, vol. 15, pp. 529-532, 1974.
- [113] D. N. Lee, The optical flow field: the foundation of vision, in *Phil. Trans. Roy. Soc. London*, 1980.

- [114] H. Lee and Z. Chen, Determination of 3d human body postures from a single view, in *CVGIP*, vol. 30, 1985.
- [115] P. Lennie, C. Trevarthen, D. V. Essen and H. Wassele, Parallel processing of visual information, in *Visual Perception The Neurophysiological Foundations* (L. Spillmann and J. Werner Eds.), Academic Press, 1990.
- [116] T. S. Levitt, D. T. Lawton, D. M. Chelberg, and P. C. Nelson, Qualitative navigation, in *Proc. Image Understanding Workshop*, pp. 447-465, 1987.
- [117] F. Leymarie and M. D. Levine, Tracking deformable objects in the plane using an active contour model, in *IEEE Trans. PAMI*, vol. 15, pp. 617-634, 1993.
- [118] H. Li, P. Roivainen and R. Forchheimer, 3-D motion estimation in model-based facial image coding, in *IEEE Trans. PAMI*, vol. 15, pp. 545-555, 1993.
- [119] H. C. Longuet-Higgins and K. Pradny, The interpretation of a moving retinal image, in *Proc. R. Soc. Lond.*, B208:385-397, 1980.
- [120] H. C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, in *Nature*, 1981.
- [121] M. M. Loeve, *Probability Theory*, Princeton, NJ: Van Nostrand, 1955.
- [122] D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.
- [123] R. Malladi, J. Sethian and B. Vemuri, Shape modeling with front propagation: a level set approach, in *IEEE Tran. PAMI*, vol. 17, pp. 158-175, 1995.

- [124] D. Marr and T. Poggio, Cooperative computation of stereo disparity, in *Science*, 194:283-287, 1975.
- [125] D. Marr and H.K. Nishihara, Representation and recognition of the spatial organization of three-dimensional shapes, in *Proc. R. Soc. Lond. B 200*, pp. 269-294, 1978.
- [126] D. Marr and T. Poggio, A computational theory of human stereo vision, in *Proc. Roy. Soc., B-204*:301-328, 1979.
- [127] D. Marr, *Vision*, Freeman, San Francisco, 1982.
- [128] W. N. Martin, J. K. Aggarwal (eds.), *Motion Understanding: Robot and Human Motion*, Kluwer, Boston, MA 1988.
- [129] J. L. Martinez, Jr. and R. P. Kessner (eds.), *Learning & Memory: A Biological View*, 2nd ed., Academic Press, San Diego, 1991.
- [130] K. Mase and A. Pentland, Lip Reading: Automatic visual recognition of spoken words, Technical Report 117, M.I.T. Media Lab Vision Science, 1989.
- [131] L. Matthies and S. Shafer, Error modeling in stereo navigation, in *IEEE J. of Robotics and Automation*, vol. RA-3, no. 3, 1987.
- [132] A. Mitiche and J. K. Aggarwal, A computational analysis of time-varying images, in *Handbook of Pattern Recognition and Image Processing*, T.Y. Young and K.S. Fu Eds, New York: Academic. 1986.

- [133] B. Moghaddam and A. Pentland, Maximum likelihood detection of faces and hands, in Proc. International Workshop on Automatic Face- and Gesture- Recognition", pp. 122-128, June 1995.
- [134] H. P. Moravec, Obstacle avoidance and navigation in the real world by a seeing robot rover, Ph.D dissertation, Stanford Univ., CA, 1980.
- [135] J. K. Mui and K. S. Fu, Automatic classification of cervical cells using a binary tree classifier, in *Pattern Recognition*, vol. 16, pp. 69-80, 1980.
- [136] H. Murase and S. K. Nayar, Illumination planning for object recognition in structured environments, in Proc *IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, WA, pp. 31 - 38, June 1994.
- [137] M. Murray, Gait as a total pattern of movement, in *American Journal of Physiological Medicine*, vol. 46, pp. 290-333, 1967.
- [138] K. Nakayama, Biological image motion processing: a review, in *Vision Res.*, vol. 25, no. 5, pp. 625-660, 1985.
- [139] H. K. Nishihara, PRISM, a practical real-time imaging stereo matcher, Technical Report A.I. Memo 780, MIT, Cambridge, MA.
- [140] Y. Ohta and T. Kanade, Stereo by intra- and inter-scanline search, in *IEEE Trans. PAMI*, vol. 7, no. 2, pp. 139-154, 1985.

- [141] J. O'Rourke and N. I. Badler, Model-based image analysis of human motion using constraint propagation, in *IEEE Trans. PAMI*, vol. 2, no. 6, pp. 523-536, 1980.
- [142] I. Overington, *Computer Vision: a unified, biologically-inspired approach*, Elsevier Science Publishing Company Inc., New York.
- [143] A. Pentland and B. Horowitz, Recovery of nonrigid motion and structure, in *IEEE Trans. PAMI*, vol. 13, pp. 730-742, 1991.
- [144] E. D. Petajan, B. Bischoff, D. Bodoff, and N. M. Brooke, An improved automatic lipreading system to enhance speech recognition, in *Proc. SIGCHI'88: Human Factors in Computing Systems*, pp. 19-25, 1988.
- [145] F. J. Pineda, Generalization of back propagation to recurrent and higher order neural networks, in D.Z. Anderson (Ed.), *Neural information processing system*. New York: American Institute of Physics, 1988.
- [146] G. F. Poggio and T. Poggio, The analysis of stereopsis, in *Ann. Rev. Neurosci.*, vol. 7, pp. 379-412, 1984.
- [147] T. Poggio and F. Girosi, Networks for Approximation and Learning, in *Proceedings of the IEEE*, no. 9, vol. 78, pp. 1481-1497, 1990.
- [148] J. Pola and H. Wyatt, Target position and velocity: the stimuli for pursuit eye movements, in *Vision Res.*, vol. 20, pp. 523-534, 1980.
- [149] R. Polana and R. Nelson, Detecting Activities, in *IEEE CVPR*, pp. 2-7, 1993.

- [150] Pollard, Mayhew and Frisby, PMF: A stereo correspondence algorithm using a disparity gradient constraint, in *Perception*, 14:449-470, 1985.
- [151] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, New York, 1986.
- [152] V. S. Ramachandran, "Perceiving shape from shading," in I. Rock (ed.), *The Perceptual World*, Freeman, San Francisco, CA, 1990, pp. 127-138.
- [153] K. Rangarajan and M. Shah, Establishing motion correspondence, in *CVGIP: image understanding*, vol. 54, pp. 56-73, 1991.
- [154] C. R. Rao, *Linear Statistical Inference and Its Applications*, 2nd Ed., Wiley, New York, 1973.
- [155] C. Rashbass, The relationship between saccadic and smooth tracking eye movements, in *J. Physiol.*, vol. 159, pp. 326-338, 1961.
- [156] R.F. Rashid, Towards a system for the interpretation of moving light displays, in *IEEE Trans. PAMI*, vol. 2, no. 6, pp. 523-536, 1980.
- [157] , D. Reisfeld, H. Woldson and Y. Yeshurun, "Context-free attentional operators: the generalized symmetry transform", in *Int'l J. of Computer Vision*, vol. 14, pp. 119-130, 1995.
- [158] J. W. Roach and J. K. Aggarwal, Determining the movement of objects from a sequence of images, in *IEEE Trans. PAMI*, vol. PAMI-2, no. 6, pp. 554-562, 1980.

- [159] K. Rohr, Incremental recognition of pedestrians from image sequences, in *IEEE. CVPR*, pp. 8-13, 1993.
- [160] O. Sacks, "To see and not see," *The New Yorker*, May, 10, 1993.
- [161] A. Searleman and D. Herrmann, *Memory from a Broader Perspective*, ch. 5, McGraw-Hill, Inc., 1994.
- [162] T. Sellis, N. Roussopoulos and C. Faloutsos, "The r+-tree: a dynamic index for multidimensional objects", in *Proceedings of 13th International Conference on VLDB*, pp. 507-518, 1987.
- [163] I. K. Sethi and G. P. R. Savarayudu, "Hierarchical classifier design using mutual information", in *IEEE Trans. Pattern Recog. and Machine Intell.*, vol. 4, pp. 441-445, 1982.
- [164] H. Shariat, and K. E. Price, Motion estimation with more than two frames, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 5, pp. 417-434, 1990.
- [165] A. Sommerfeld, *Mechanics of Deformable Bodies*, 1950.
- [166] R. Sorabji, *Aristotle on Memory*, Brown Univ. Press, Providence, RI 1972.
- [167] G. Sperling, M. Landy, Y. Cohen, and M. Pavel, Intelligible encoding of ASL image sequences at extremely low information rates, in *CVGIP*, vol. 31, pp. 335-391, 1985.

- [168] T. E. Starner and A. Pentland, Visual recognition of American sign language using hidden markov models, in *Proc. International Workshop on Automatic Face- and Gesture- Recognition*, pp. 189-194, June 1995.
- [169] S. E. Stead, *Smooth multistage multivariate approximation*, Ph.D. Dissertation, Department of Math., Brown University, 1983.
- [170] R. M. Steinman and J. Z. Levinson, The role of eye movement in the detection of contrast and spatial detail, in *Eye Movements and Their Role in Visual and Cognitive Process* (E. Kowler Ed.). pp. 115-212, Elsevier, 1990.
- [171] W. Stokoe, Sign language structure: an outline of the visual communication system of the American deaf, *Studies in Linguistics Occasional Paper No. 8*, 1960.
- [172] N. Sugie and H. Inagaki, A computational aspect of kinetic depth effect, in *Biol. Cybern.*, vol. 50, pp. 431-436, 1984.
- [173] T. Takahashi and F. Kishino, Hand gesture coding based on experiments using a hand gesture interface device, in *Sigchi Bulletin*, vol. 23, pp. 67-74, 1991.
- [174] D. Terzopoulos and D. Metaxas, Dynamic 3D models with local and global deformations: deformable superquadrics, in *IEEE Trans. PAMI*, vol. 13, pp. 703-714, 1991.
- [175] D. Terzopoulos and K. Waters, Analysis and synthesis of facial image sequences using physical and anatomical models, in *IEEE Trans. PAMI*, vol. 15, pp. 569-579, 1993.

- [176] P. Thompson, Margaret Thatcher: a new illusion, *Perception*, vol. 9, pp. 483-484, 1980.
- [177] A. P. Tirumalai, B. G. Schunck, and R. C. Jain, Dynamic stereo with self-calibration, in *Proc. 3rd International Conference on Computer Vision*, Osaka, Japan, pp. 466-470, 1990.
- [178] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: a factorization method, in *Int. J. of comput. Vis.*, 9:2, pp.137-154, 1992.
- [179] R. Y. Tsai and T. S. Huang, Uniqueness and estimation of three-dimensional parameters of rigid objects with curved surface, in *IEEE Trans. PAMI*, vol. 6, pp. 13-26, Jan. 1984.
- [180] M. Turk and A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
- [181] S. Ullman, *The interpretation of visual motion*, Cambridge, MA: MIT Press, 1979.
- [182] S. Ullman, Analysis of visual motion by biological and computer systems, in *IEEE*, August, pp. 57-69, 1981.
- [183] M. Umeda, Recognition of multi-font printed Chinese characters, in *Proc. 6th ICPR*, pp. 793-796, 1982.
- [184] B. Waerden, *Modern Algebra*, vol. 2, Berlin, Germany, Springer, 1940.

- [185] D. F. Watson, Computing the n-dimensional Delaunay triangulation with application to Voronoi polytopes, in *Computer Journal*, vol. 24, 1981.
- [186] J. A. Webb and J. K. Aggarwal, Structure from motion of rigid and jointed bodies, in *Proc. 7th Int. Joint Conf. Artificial Intell.*, Vancouver, Canada, 1981.
- [187] J. Weng, T.S. Huang and N. Ahuja, 3-D motion estimation, understanding and prediction from noisy image sequences, in *IEEE Trans. PAMI*, vol. 6, pp.545-554, Sept. 1984.
- [188] J. Weng, N. Ahuja, and T. S. Huang, Two-view matching, in *Proc. Second International Conference on Computer Vision*, pp. 64-73, 1988.
- [189] J. Weng T.S. Huang and N. Ahuja, Motion and structure from two perspective views: algorithm, error analysis, and error estimation, in *IEEE Trans. PAMI*, vol. 11, no. 5, pp. 451-476, May, 1989.
- [190] J. Weng and P. Cohen and M. Herniou, Stereo camera calibration with nonlinear corrections, in *Proc. Tenth International Conference on Pattern Recognition*, Atlantic City, New Jersey, pp. 246-253, June, 1990.
- [191] Weng, Cohen and Rebibo, Motion and structure estimation from stereo image sequences, in *IEEE Trans. Robotics and Automation*, vol. 8, no. 3, pp. 362-382, June, 1992.
- [192] J. Weng, N. Ahuja, and T. S. Huang, Optimal motion and structure estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, Sept. 1993, pp. 864-884.

- [193] J. Weng, On comprehensive visual learning, in *Proc. NSF/ARPA Workshop on Performance vs. Methodology in Computer Vision*, pp. 152-166, Seattle, WA, June 24-25, 1994.
- [194] J. Weng, SHOSLIF: the hierarchical optimal subspace learning and inference framework, Technical Report CPS 94-15, Michigan State University.
- [195] J. Weng, Y. Cui, N. Ahuja and A. Singh, Integration of Transitory Image Sequences, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 966-969, Seattle, June 1994.
- [196] J. Weng and S. Chen, SHOSLIF convergence properties and MDF version of SHOSLIF-N, Technical Report CPS-95-22, Department of Computer Science, Michigan State University, East Lansing, MI, 1995.
- [197] S.S.Wilks, *Mathematical Statistics*, Wiley, New York, 1963.
- [198] P.R. Wolf, *Elements of Photogrammetry*, New York: McGraw-Hill, 1974.
- [199] Y. Yacoob and L. Davis, Computing spatial-temporal representations of human faces, in *IEEE Conf. CVPR*, pp. 70-75, 1994.
- [200] Y. Yasumoto and G. Medioni, Robust estimation of three-dimensional motion parameters from sequence of image frames using regularization, in *IEEE Trans. PAMI*, vol. 8, no. 4, pp. 464-471, 1986.

- [201] M. Yachida, , 3D data acquisition by multiple views, in O.D. Faugeras and G. Giralt eds. *Robotics Research: the Third International Symposium*, MIT Press, Cambridge, MA. pp. 11-18, 1986.
- [202] J. Yamato J. Ohya and K. Ishii, Recognizing human action in time- sequential images using hidden markov model, in *IEEE CVPR*, pp. 379-385, 1992.
- [203] X. Zhuang, T. S. Huang and R. M. Haralick, Two-view motion analysis: A unified algorithm, in *J. Opt. Soc. Amer. A*. vol. 3, no. 9, pp. 1492-1500, 1986.
- [204] Z. Zhang and O. Faugeras, Three-Dimensional Motion Computation and Object Segmentation in a Long Sequence of Stereo Frames, in *Int'l J. of Computer Vision*, 7:3, pp. 211-241, 1992.
- [205] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces", in *International Journal of Computer Vision*, vol. 13, pp. 119-152, 1994.
- [206] T. G. Zimmerman et al., A hand gesture interface device, in *Proc. Human Factors in Computing Systems and Graphics Interface*, pp. 189-192, ACM Press, New York, 1987.

MICHIGAN STATE UNIV. LIBRARIES



31293015550225